

The Irrationality of $\zeta(5)$

Alex Hyman

October 29, 2026

Abstract

We formalise the proof that $\zeta(5)$ is irrational from A. Fauzan’s 2026 preprint [2], following the Lean 4/Mathlib formalisation by domino14 [1], of which this entry is a port. The argument builds integer polynomials Q_n with $0 < Q_n(\zeta(5)) < e^{-cn^2}$ as Hankel determinants of a rational functional: a positive moment representation of $\zeta(5)$ gives positivity, p -adic estimates through unimodular bases in two prime ranges and a uniform asymptotic for the valuations give the integrality factor, and a logarithmic-energy estimate for an explicit measure gives the archimedean bound, proving irrationality. The main theorem is $\text{zeta } 5 \notin \mathbb{Q}$ for the *zeta* of the entry `Zeta_Function`; the only oracle is *holds*, from the 72 numeric inequalities discharged by *approximation*. The port is not literal: Lean’s dependently typed *Fin* n matrices and Σ -indexed bases become `Jordan_Normal_Form` matrices over explicit index lists, and the p -adic field $\mathbb{Q}_p$ is replaced by truncated rational series with congruences modulo powers of p , so that the whole development lives in `HOL` over $\mathbb{Q}$ without `Types-To-Sets`; Lean’s kernel computation on 1052 dyadic intervals becomes *approximation* with splitting.

Thirteen theories are independent of the proof and reusable: a calculus for lower bounds on p -adic valuations of rationals and polynomials with a mod- p unimodularity criterion (*unimodular_criterion*), the Vandermonde determinant for `Jordan_Normal_Form` matrices (*det_vandermonde_jnf*) and positivity of Gram determinants, Riemann-sum limits (*outer_prime_sum_bound*) of sums over primes from the prime number theorem, the log-sine integral and the circle average of $\ln|z - a|$ (*ln_sin_integral*, *circle_avg_log*), the arcsine law and its potential (*arcsine_potential*), Andréief’s identity for σ -finite measures (*andreief*), and the zero-mass logarithmic energy inequality on $\mathbb{C}$ (*log_energy_zero_mass_nonpos*).

The port was an experiment in AI proof engineering: an orchestration model (Claude Fable 5.1) wrote Isar skeletons with fact hints, proof agents (Claude Opus 5.5) closed about 1,100 leaves interactively in Isabelle/jEdit, and automated scripts harvested *try0* and *sledgehammer* results to reduce context load on the agents. The problem was chosen because its final statement is short and is expressed with

existing AFP definitions, so the result is easy to audit. The formalisation effort spanned 3 days of continuous work with a fair amount of human intervention to keep Fable and the agents on track, give them Isabelle-related hints, insist on legible proofs, and oversee the structure of the project. The author has audited the files and takes full responsibility for their contents.

Relation to the Lean formalisation

The definitions, statements and proof structure follow the Lean project theory by theory (each Isabelle theory names its Lean counterparts). The main departures:

- The p -adic field $\mathbb{Q}_p$ of the paper’s Sections 3.1–3.2 is not constructed. Every p -adically convergent series is replaced by its length- M partial sum (the `__tr` definitions) and the identities become congruences modulo powers of p , so the development stays in HOL over $\mathbb{Q}$ without Types-To-Sets (theories `Zeta5_Truncated_Expansions`, `Zeta5_Pole_Distribution`).
- Lean’s *Fin* n matrices and Σ -indexed bases become `Jordan_Normal_Form` matrices with explicit index lists (`inner_idx`, `outer_idx`); dependent types were never needed, only bookkeeping.
- Lean checks the potential inequality (A.9) by kernel computation on 1052 dyadic intervals; here it is 52 region lemmas by *approximation* with splitting, and the 16 energy enclosures (A.10) are per-term *approximation* runs after rescaling the logarithms into $[1, 2)$.
- Mathlib facts with no HOL/AFP counterpart were proved from scratch and are the general theories listed in the abstract; the moment representation (Proposition 2.2) is derived from the Laplace transform of $\sin$ and the cotangent partial-fraction formula of the AFP rather than from a Hurwitz-zeta integral, the prime number theorem is the AFP’s (*var asymptotics*), the von Staudt–Clausen facts come from the entry `Bernoulli`, and the rational valuations from `Kummer_Congruence`.
- The zero-mass energy inequality (Lemma 6.2) is stated with explicit hypotheses (a finite dominating measure with $\ln|z - w|$ integrable on the product and a null diagonal), which is what the paper’s “finite energy, compact support” has to mean for signed measures.

Contents

1	A calculus for lower bounds on p-adic valuations of rationals and polynomials	8
----------	---	----------

1.1	Definitions	8
1.2	Basic closure properties	8
1.3	The ultrametric inequality for <i>qmultiplicity</i> (not in the AFP)	9
1.4	Valuations of powers and products	14
2	Vandermonde and Gram determinants for JNF matrices	16
3	Taylor coefficients of a rational polynomial at a point	21
4	Determinants of p-integral matrices and a unimodularity criterion	24
5	Legendre’s formula for the p-adic valuation of $n!$	28
6	Riemann-sum limits of sums over primes from the prime number theorem	30
6.1	Chebyshev’s ϑ over cells	30
6.2	Small primes: $\psi - \vartheta$ and the valuation of 24	32
6.3	Step majorants on a uniform partition	35
6.4	Dominated convergence for the majorants	42
6.5	The Riemann-sum bounds	47
7	Interval integrals of piecewise polynomial and rational functions	56
7.1	Generic piece lemmas	56
8	Gamma-type integrals, the Laplace transform of <i>sin</i> and the <i>coth</i> series	60
8.1	Gamma integrals and the interchange of $\int$ and $\sum$	60
8.2	The S -series and the Laplace transform of <i>sin</i>	67
8.3	The pole integral	76
9	Substitutions in interval integrals with finite endpoints	79
10	The log-sine integral and circle averages of $\ln z - a$	84
11	The arcsine distribution and its logarithmic potential	100
11.1	The arcsine measure	100
12	Logarithmic energy of measures on $\mathbb{C}$ and the zero-mass energy inequality	123
13	Andreief’s identity (continuous Cauchy–Binet)	155

14 Definitions of the construction	161
14.1 Generic helpers	161
14.2 Section 2.1 Parameters (2.1)	161
14.3 Section 2.1 The functional μ_X (2.2), (2.3)	162
14.4 Section 2.1 The matrix, determinant and normalisation (2.4), (2.5)	162
14.5 Section 3 The functional τ_X	163
14.6 Section 3.1-3.2 The p -adic extension, in truncated rational form	164
14.7 Section 3.3 Small primes	165
14.8 Section 4.1 The inner range	165
14.9 Section 4.2 The outer range	167
14.10 Section 5 Normalisation (5.1), (5.2), (2.6)	169
14.11 Section 5.1-5.3 Limiting functions (real-valued; Lean names- pace <i>Lim</i>)	170
14.12 Section 6 The real determinant	171
15 The functionals μ_X and τ_X: algebra and partial fractions	173
15.1 Linearity	173
15.2 Well-definedness: common factors cancel	174
15.3 Partial fractions (Lagrange decomposition)	176
15.4 (3.1) The pullback identity	178
15.5 τ on polynomials	179
15.6 (3.1) The pullback identity: proof	184
16 Integer-valued polynomials and the bound (3.9)	195
16.1 Binomial polynomials	195
16.2 The coefficients of $\text{binom}(x, n)$	197
16.3 Newton interpolation	199
16.4 (3.9) and (3.8)	202
17 The far-pole constant and Lemma 3.3	205
17.1 Bernoulli valuations (von Staudt-Clausen, from the AFP) . .	205
17.2 Section 3.1 The far-pole constant C_p	211
17.3 Section 3.3 Lemma 3.3 and the small-prime bound	218
17.4 Counting multiples in an interval	218
17.5 Valuations of integers as counts	220
17.6 Partial fractions with linear poles	223
17.7 Valuations of integers	225
17.8 The product-difference estimate	227
17.9 Lemma 3.3	228

18 The small-prime bound (3.12)	238
18.1 Section 3.3 Integer-valued basis polynomials	238
18.2 The determinant identity (3.11)	243
18.3 The small-prime bound (3.12)	249
19 The truncated p-adic layer and Lemma 3.1	252
19.1 Truncated expansions of $1/(z - s)$ as polynomials	252
19.2 The near residues	259
19.3 The analytic part	263
19.4 Lemma 3.1 with far poles, truncated rational form	274
20 The distribution formula (3.7) and Lemma 3.2	278
20.1 Raabe's multiplication formula	278
20.2 Harmonic increments and the pole sums	281
20.3 The pole values of $\tau^{\widehat{\{ext\}}}$ - Y on the distributed poles	285
20.4 Partial fractions and the substitution $x = a + p z$	287
20.5 Lemma 3.2	291
21 The inner range: Proposition 4.1	294
21.1 Section 4.1 The inner range	294
21.2 Extracting powers of p from $g(c + p z)$	305
21.3 The inner entry bounds (4.2)/(4.3)	317
22 The outer range: Lemma 4.2 and Proposition 4.3	341
22.1 Section 4.2 The outer range	341
22.2 The modified functional $\mu^{\widehat{0-X}}$ and the structure of the outer basis	352
22.3 Arithmetic of the pole classes	356
22.4 Valuations at the poles	360
22.5 The two small poles $a, p - a$	369
22.6 The zero class	377
22.7 The entry bound (4.12)	379
22.8 The closed form (4.14) of $\gamma^{\widehat{p-out}}$	385
22.9 Lemma 4.2	398
22.10 Proposition 4.3	407
22.11 Proposition 4.3 for $p > K$	428
23 The determinant construction and the moment representa- tion of $\zeta(5)$	439
23.1 (2.9): degree and leading coefficient of Δ - K	440
23.2 Proposition 2.2: the positive integral representation	451
23.3 Moments and the pole values	455
24 Exact rational constants	468

25 The uniform estimate (5.7) in the inner range	469
25.1 Floors and $v\text{-}p(S\text{-}K)$ in the inner range	469
25.2 The Riemann-sum identity for the z -integral	475
25.3 Exact class counts	479
25.4 The discrete decomposition of $\gamma\text{-}p\text{-}\widehat{in}$	482
25.5 Discrete sum versus the integral	491
25.6 The comparison functions	495
25.7 Assembly	498
26 The explicit integrals of Section 5	504
26.1 The closed form $R(x) = x F(x) + Q(x)$ (B.1), (5.13)	505
26.2 $\int_{-\{1/3\}}^{\{1/2\}} d(y) dy = 9/640$, the mean of P , and $ C $ < 16	517
26.3 (5.10): $\int_{-\{1/3\}}^{\{2\lambda\}} \text{outerIntegrand} = I\text{-out}$ by 11 affine pieces	523
26.4 (5.18): $\int_{-\{3\}}^{\{20\}} R(x)/x^3 dx$ by 143 affine pieces	535
26.5 Integrability on $[3, \infty)$ and the tails (5.14)-(5.17)	648
26.6 Continuity off a countable set, and boundedness	668
27 Normalisation and the asymptotics of $Q_{K,M}$: Propositions 5.1 and 5.2	676
27.1 The small-prime sum	676
27.2 Standing hypotheses and integrality	679
27.3 Legendre's formula and the uniformity estimates	680
27.4 Proposition 5.1	682
27.5 The outer range of the uniformity estimate (Lean: Legendre5)	685
27.6 Exact integrals and tails	696
27.7 Prime sums via the prime number theorem	696
27.8 Continuity of the limiting functions off a countable set	697
27.9 Proposition 5.2 and (5.21)	697
28 The comparison measure ϱ and its logarithmic potential	712
28.1 Table 1	712
28.2 Elementary enclosures	722
28.3 (A.9): the potential inequality on the half-line	724
28.4 Ingredients of (6.9) that live here	730
29 Numeric enclosures: (A.9), (A.10) and Lemma 6.1	733
29.1 Elementary enclosures (A.10)	734
29.2 (A.9): the region check on $[0, 2]$	738
29.3 (6.9) ingredient	752

30 The archimedean bound: Proposition 6.3	753
30.1 (6.11) and Lemma 6.2	754
30.2 (6.12) and the one-coordinate bound	761
30.3 (6.9) The bound for every configuration	771
30.4 (6.14), (6.15) and Proposition 6.3	805
31 The main theorem: $\zeta(5)$ is irrational	822

```

theory Rational-Padic-Valuations
  imports Kummer-Congruence.Rat-Congruence Jordan-Normal-Form.Determinant
  HOL-Library.Log-Nat
begin

```

1 A calculus for lower bounds on p -adic valuations of rationals and polynomials

The p -adic valuation of a rational number is *qmultiplicity* from the AFP entry `Kummer_Congruence`. This theory provides the predicate *vge* $p\ x\ b$ ($x = 0$ or $v_p(x) \geq b$) and its coefficientwise version *vpGge* $p\ A\ b$ for polynomials (the Gauss valuation), together with their closure under sums, products, powers, polynomial evaluation, division by monic polynomials and the Leibniz terms of determinants, the ultrametric inequality for *qmultiplicity* (absent from the AFP), and the valuations of powers and products. Nothing here depends on the irrationality proof.

1.1 Definitions

The polynomial indeterminate.

abbreviation $pX :: \text{rat poly}$ **where** $pX \equiv \text{Polynomial.monom } 1\ 1$

The p -adic valuation of a rational (Mathlib: *padicValRat*).

definition $vp :: \text{nat} \Rightarrow \text{rat} \Rightarrow \text{int}$ **where**
 $vp\ p\ x = \text{qmultiplicity } (\text{int } p)\ x$

$\lfloor \log_p n \rfloor$ (Mathlib's *Nat.log* $p\ n$; 0 for $n = 0$ or $p \leq 1$).

definition $\text{nat-log} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat}$ **where**
 $\text{nat-log } p\ n = \text{floorlog } p\ n - 1$

$x = 0 \vee b \leq v_p(x)$.

definition $vge :: \text{nat} \Rightarrow \text{rat} \Rightarrow \text{rat} \Rightarrow \text{bool}$ **where**
 $vge\ p\ x\ b \longleftrightarrow (x \neq 0 \longrightarrow b \leq \text{of-int } (vp\ p\ x))$

$v_p \wedge G(A) \geq b$: every nonzero coefficient has valuation at least b .

definition $vpGge :: \text{nat} \Rightarrow \text{rat poly} \Rightarrow \text{rat} \Rightarrow \text{bool}$ **where**
 $vpGge\ p\ A\ b \longleftrightarrow (\forall i. \text{Polynomial.coeff } A\ i \neq 0 \longrightarrow b \leq \text{of-int } (vp\ p\ (\text{Polynomial.coeff } A\ i)))$

1.2 Basic closure properties

lemma *prime-multiplicity-prime*:
fixes $p\ q :: 'a :: \text{factorial-semiring}$
assumes $\text{prime } p\ \text{prime } q$

shows $\text{multiplicity } p \ q = (\text{if } p = q \text{ then } 1 \text{ else } 0)$
using *assms* **by** (*auto simp: prime-multiplicity-other*)
lemma *vge-zero*: $\text{vge } p \ 0 \ b$ **by** (*simp add: vge-def*)
lemma *vge-mono*: $\text{vge } p \ x \ b \implies b' \leq b \implies \text{vge } p \ x \ b'$
by (*auto simp: vge-def*)

1.3 The ultrametric inequality for *qmultiplicity* (not in the AFP)

lemma *qmultiplicity-power-int*:
assumes *prime p*
shows $\text{qmultiplicity } (\text{int } p) \ (\text{of-nat } p \text{ powi } k) = k$
proof –
have *pe*: $\text{prime-elem } (\text{int } p)$ **using** *assms* **by** (*simp add: prime-nat-int-transfer*)
have *pow*: $\bigwedge n. \text{qmultiplicity } (\text{int } p) \ (\text{of-nat } p \ ^n) = \text{int } n$
proof –
fix *n* :: *nat*
have $\text{qmultiplicity } (\text{int } p) \ (\text{of-nat } p \ ^n) = \text{qmultiplicity } (\text{int } p) \ (\text{of-nat } (p \ ^n))$
by *simp*
also **have** $\dots = \text{int } (\text{multiplicity } (\text{int } p) \ (\text{int } (p \ ^n)))$ **by** (*rule qmultiplicity-of-nat*)
also **have** $\dots = \text{int } n$ **using** *pe* **by** (*simp add: multiplicity-prime-power*)
finally **show** $\text{qmultiplicity } (\text{int } p) \ (\text{of-nat } p \ ^n) = \text{int } n$.
qed
show *?thesis*
by (*cases k ≥ 0*) (*simp-all add: power-int-def pow power-inverse*)
qed

lemma *qmultiplicity-nonneg-add*:
assumes *prime p* $0 \leq \text{qmultiplicity } (\text{int } p) \ x$ $0 \leq \text{qmultiplicity } (\text{int } p) \ y$ $x + y \neq 0$
shows $0 \leq \text{qmultiplicity } (\text{int } p) \ (x + y)$
proof –
obtain *a b* **where** *ab*: $\text{quotient-of } x = (a, b)$ **using** *prod.exhaust* **by** *blast*
obtain *c d* **where** *cd*: $\text{quotient-of } y = (c, d)$ **using** *prod.exhaust* **by** *blast*
have *b*: $b > 0$ **and** *d*: $d > 0$ **using** *ab cd quotient-of-denom-pos* **by** *blast+*
have *p1*: $|\text{int } p| \neq 1$ **using** *prime-gt-1-nat[OF assms(1)]* **by** *simp*
have *nb*: $\neg \text{int } p \text{ dvd } b$ **using** *qmultiplicity-nonneg-imp-not-dvd-denom[OF assms(2)]*
p1 *ab*
by *simp*
have *nd*: $\neg \text{int } p \text{ dvd } d$ **using** *qmultiplicity-nonneg-imp-not-dvd-denom[OF assms(3)]*
p1 *cd*
by *simp*
have *x*: $x = \text{of-int } a \ / \ \text{of-int } b$ **and** *y*: $y = \text{of-int } c \ / \ \text{of-int } d$
using *ab cd* **by** (*simp-all add: quotient-of-div*)
have *xy*: $x + y = \text{of-int } (a * d + c * b) \ / \ \text{of-int } (b * d)$
using *b d* **by** (*simp add: x y add-frac-eq*)
have *pe*: $\text{prime-elem } (\text{int } p)$ **using** *assms(1)* **by** (*simp add: prime-nat-int-transfer*)

```

have num:  $a * d + c * b \neq 0$ 
proof
  assume  $a * d + c * b = 0$ 
  hence (of-int ( $a * d + c * b$ ) :: rat) = 0 by simp
  with xy have  $x + y = 0$  by simp
  thus False using assms(4) by simp
qed
have n0: (of-int ( $a * d + c * b$ ) :: rat)  $\neq 0$  using num
  by (simp only: of-int-eq-0-iff num simp-thms)
have qmultiplicity (int p) ( $x + y$ ) =
  qmultiplicity (int p) (of-int ( $a * d + c * b$ )) - qmultiplicity (int p) (of-int
( $b * d$ ))
  unfolding xy by (intro qmultiplicity-divide pe n0) (use b d in simp)
also have ... = int (multiplicity (int p) ( $a * d + c * b$ )) - int (multiplicity (int
p) ( $b * d$ ))
  by (simp only: qmultiplicity-of-int)
finally have qmultiplicity (int p) ( $x + y$ ) =
  int (multiplicity (int p) ( $a * d + c * b$ )) - int (multiplicity (int p) ( $b * d$ )) .
moreover have multiplicity (int p) ( $b * d$ ) = 0
  using nb nd assms(1) by (intro not-dvd-imp-multiplicity-0) (simp add: prime-dvd-mult-iff
prime-nat-int-transfer)
ultimately show ?thesis by simp
qed

```

```

lemma qmultiplicity-add-ge:
  assumes prime p  $x \neq 0$   $y \neq 0$   $x + y \neq 0$ 
  shows min (qmultiplicity (int p) x) (qmultiplicity (int p) y)  $\leq$ 
    qmultiplicity (int p) ( $x + y$ )
proof -
  define k where  $k = \min (qmultiplicity (int p) x) (qmultiplicity (int p) y)$ 
  define u :: rat where  $u = \text{of-nat } p \text{ powi } (-k)$ 
  have pe: prime-elem (int p) using assms(1) by (simp add: prime-nat-int-transfer)
  have u0:  $u \neq 0$  using assms(1) by (simp add: u-def prime-gt-0-nat)
  have qu: qmultiplicity (int p) u = -k using assms(1)
    by (simp add: u-def qmultiplicity-power-int)
  have 0  $\leq$  qmultiplicity (int p) ( $x * u$ ) 0  $\leq$  qmultiplicity (int p) ( $y * u$ )
    using assms u0 pe qu by (simp-all add: qmultiplicity-mult k-def)
  moreover have  $x * u + y * u \neq 0$  using assms u0 by (simp add: distrib-right[symmetric])
  ultimately have 0  $\leq$  qmultiplicity (int p) (( $x + y$ ) * u)
    using qmultiplicity-nonneg-add[OF assms(1)] by (simp add: distrib-right)
  thus ?thesis using assms u0 pe qu by (simp add: qmultiplicity-mult k-def)
qed

```

```

lemma vge-add:
  assumes prime p vge p x b vge p y b
  shows vge p ( $x + y$ ) b

```

```

proof (cases  $x = 0 \vee y = 0 \vee x + y = 0$ )
  case True
    thus ?thesis using assms by (auto simp: vge-def)
next
  case False
    hence  $x \neq 0 \wedge y \neq 0 \wedge x + y \neq 0$  by auto
    have  $b \leq \text{of-int } (\min (\text{qmultiplicity } (\text{int } p) \ x) (\text{qmultiplicity } (\text{int } p) \ y))$ 
      using assms(2,3)  $\langle x \neq 0 \rangle \langle y \neq 0 \rangle$ 
      by (auto simp: vge-def vp-def min-def)
    also have  $\dots \leq \text{of-int } (\text{qmultiplicity } (\text{int } p) \ (x + y))$ 
      using qmultiplicity-add-ge[OF assms(1)  $\langle x \neq 0 \rangle \langle y \neq 0 \rangle \langle x + y \neq 0 \rangle$ ] by (simp only: of-int-le-iff)
    finally show ?thesis by (simp add: vge-def vp-def)
qed
lemma vge-neg:  $\text{vge } p \ x \ b \implies \text{vge } p \ (-x) \ b$  by (simp add: vge-def vp-def)
lemma vge-sum:
  assumes prime  $p$  finite  $I \wedge i. i \in I \implies \text{vge } p \ (f \ i) \ b$ 
  shows  $\text{vge } p \ (\sum i \in I. f \ i) \ b$ 
  using assms(2,3) by (induction  $I$  rule: finite-induct) (auto simp: vge-zero vge-add[OF assms(1)])
lemma vge-mul:
  assumes prime  $p$   $\text{vge } p \ x \ a \ \text{vge } p \ y \ b$ 
  shows  $\text{vge } p \ (x * y) \ (a + b)$ 
proof (cases  $x = 0 \vee y = 0$ )
  case True
    thus ?thesis by (auto simp: vge-def)
next
  case False
    have prime-elem (int  $p$ ) using assms(1) by (simp add: prime-nat-int-transfer)
    with False assms show ?thesis by (simp add: vge-def vp-def qmultiplicity-mult)
qed
lemma vge-one:  $\text{vge } p \ 1 \ 0$  by (simp add: vge-def vp-def)
lemma vge-pow:  $\text{prime } p \implies \text{vge } p \ x \ 0 \implies \text{vge } p \ (x \wedge^n) \ 0$ 
  by (induction  $n$ ) (auto simp: vge-one dest: vge-mul[where  $a = 0$  and  $b = 0$ , simplified])

lemma vge-inv-p:  $\text{prime } p \implies \text{vge } p \ (1 / \text{of-nat } p) \ (-1)$ 
  by (simp add: vge-def vp-def inverse-eq-divide[symmetric] prime-gt-0-nat)
lemma vge-of-int:  $\text{prime } p \implies \text{vge } p \ (\text{of-int } n) \ 0$  by (simp add: vge-def vp-def)

lemma vpGge-iff:  $\text{vpGge } p \ A \ b \iff (\forall i. \text{vge } p \ (\text{Polynomial.coeff } A \ i) \ b)$ 
  by (simp add: vpGge-def vge-def)
lemma vpGge-zero:  $\text{vpGge } p \ 0 \ b$  by (simp add: vpGge-def)
lemma vpGge-mono:  $\text{vpGge } p \ A \ b \implies b' \leq b \implies \text{vpGge } p \ A \ b'$ 
  by (auto simp: vpGge-def)
lemma vpGge-add:  $\text{prime } p \implies \text{vpGge } p \ A \ b \implies \text{vpGge } p \ B \ b \implies \text{vpGge } p \ (A + B) \ b$  by (auto simp: vpGge-iff vge-add)
lemma vpGge-neg:  $\text{vpGge } p \ A \ b \implies \text{vpGge } p \ (-A) \ b$  by (simp add: vpGge-def vp-def)

```

```

lemma vpGge-sum:
  assumes prime p finite I  $\bigwedge i. i \in I \implies \text{vpGge } p (f i) b$ 
  shows  $\text{vpGge } p (\sum i \in I. f i) b$ 
  using assms(2,3) by (induction I rule: finite-induct) (auto simp: vpGge-zero
    vpGge-add[OF assms(1)])
lemma vpGge-C:  $\text{vge } p c b \implies \text{vpGge } p [:c:] b$ 
  by (auto simp: vpGge-def vge-def coeff-pCons split: nat.splits)
lemma vpGge-one:  $\text{vpGge } p 1 0$  by (simp add: vpGge-def vp-def)
lemma vpGge-mul:  $\text{prime } p \implies \text{vpGge } p A a \implies$ 
   $\text{vpGge } p B b \implies \text{vpGge } p (A * B) (a + b)$ 
  by (auto simp: vpGge-iff coeff-mult intro!: vge-sum vge-mul)
lemma vpGge-prod:
  assumes prime p finite I  $\bigwedge i. i \in I \implies \text{vpGge } p (f i) (b i)$ 
  shows  $\text{vpGge } p (\prod i \in I. f i) (\sum i \in I. b i)$ 
  using assms(2,3) by (induction I rule: finite-induct) (auto simp: vpGge-one
    vpGge-mul[OF assms(1)])

```

Determinant bound from entry bounds: if every Leibniz term has valuation at least B , so does the determinant (Lean: *vpGge-det-of-terms*). Proof via *Determinant.det-def'*.

```

lemma vpGge-det-of-terms:
  assumes prime p A  $\in \text{carrier-mat } n n$ 
  and  $\bigwedge \sigma. \sigma \text{ permutes } \{0..<n\} \implies$ 
     $\text{vpGge } p (\prod i = 0..<n. A \$\$ (i, \sigma i)) B$ 
  shows  $\text{vpGge } p (\text{Determinant.det } A) B$ 
proof –
  have  $\text{vpGge } p (\sum \sigma \in \{\sigma. \sigma \text{ permutes } \{0..<n\}\}. \text{signof } \sigma * (\prod i = 0..<n. A \$\$ (i, \sigma i))) B$ 
  proof (intro vpGge-sum)
    fix  $\sigma$  assume  $\sigma \in \{\sigma. \sigma \text{ permutes } \{0..<n\}\}$ 
    hence perm:  $\sigma \text{ permutes } \{0..<n\}$  by simp
    have s:  $\text{vpGge } p (\text{of-int } (\text{Permutations.sign } \sigma)) 0$ 
    by (cases  $\sigma$  rule: Permutations.sign-cases) (simp-all add: vpGge-one vpGge-neg[OF vpGge-one])
    have  $\text{vpGge } p (\text{signof } \sigma * (\prod i = 0..<n. A \$\$ (i, \sigma i))) (0 + B)$ 
    using vpGge-mul[OF assms(1) s assms(3)[OF perm]] .
    thus  $\text{vpGge } p (\text{signof } \sigma * (\prod i = 0..<n. A \$\$ (i, \sigma i))) B$  by simp
  qed (use assms(1) in  $\langle \text{auto simp: finite-permutations} \rangle$ )
  thus ?thesis using assms(2) by (simp add: Determinant.det-def')
qed

```

The intended content of *vpGge-div-monic*, with monicity written as *lead-coeff* $Q = 1$.

```

lemma vpGge-div-lead1:
  assumes p: prime p and Q1: lead-coeff  $Q = 1$  and Qi:  $\text{vpGge } p Q 0$ 
  shows  $\text{vpGge } p P 0 \implies \text{vpGge } p (P \text{ div } Q) 0$ 
proof (induction P rule: measure-induct-rule[where f = degree])

```

```

case (less P)
show ?case
proof (cases degree P < degree Q ∨ P = 0)
  case True
    hence  $P \text{ div } Q = 0$  by (auto simp: div-poly-less)
    thus ?thesis by (simp add: vpGge-zero)
  next
    case False
    hence ge: degree Q ≤ degree P and P0: P ≠ 0 by auto
    have Q0: Q ≠ 0 using Q1 by auto
    define k where  $k = \text{degree } P - \text{degree } Q$ 
    define c where  $c = \text{lead-coeff } P$ 
    define P' where  $P' = P - \text{Polynomial.monom } c \ k * Q$ 
    have c0: c ≠ 0 using P0 by (simp add: c-def)
    have mi: vpGge p (Polynomial.monom c k) 0
      using less.prems unfolding vpGge-def c-def by (auto simp: coeff-monom
split: if-splits)
    have vpGge p (Polynomial.monom c k * Q) 0 using vpGge-mul[OF p mi Qi]
by simp
    hence vpGge p (P + - (Polynomial.monom c k * Q)) 0
      by (intro vpGge-add[OF p less.prem] vpGge-neg)
    hence P'i: vpGge p P' 0 by (simp add: P'-def)
    have d1: degree (Polynomial.monom c k * Q) = degree P
      using c0 Q0 ge by (simp add: degree-mult-eq degree-monom-eq k-def)
    have Polynomial.coeff (Polynomial.monom c k * Q) (degree P) = c
      using d1 Q1 by (metis lead-coeff-mult lead-coeff-monom mult-1-right)
    hence z: Polynomial.coeff P' (degree P) = 0 unfolding P'-def by (simp add: c-def)
    have le: degree P' ≤ degree P unfolding P'-def using d1 by (intro degree-diff-le) auto
    have IH: vpGge p (P' div Q) 0
    proof (cases P' = 0)
      case True
        thus ?thesis by (simp add: vpGge-zero)
      next
        case False
        hence  $\text{degree } P' \neq \text{degree } P$  using z by (metis leading-coeff-0-iff)
        hence  $\text{degree } P' < \text{degree } P$  using le by simp
        thus ?thesis using less.IH P'i by blast
    qed
    have  $P = P' + \text{Polynomial.monom } c \ k * Q$  by (simp add: P'-def)
    hence  $P \text{ div } Q = \text{Polynomial.monom } c \ k + P' \text{ div } Q$ 
      using Q0 by (metis div-mult-self1)
    thus ?thesis using vpGge-add[OF p mi IH] by simp
  qed
qed

```

Division of an integral polynomial by an integral monic polynomial is integral (monicity written as $\text{lead-coeff } Q = 1$; *monic* is not a constant in the base

image).

lemma *vpGge-div-monic*:

prime $p \implies \text{vpGge } p \ P \ 0 \implies \text{lead-coeff } Q = 1 \implies$
 $\text{vpGge } p \ Q \ 0 \implies \text{vpGge } p \ (P \ \text{div } Q) \ 0$
by (rule *vpGge-div-lead1*)

lemma *vge-poly-eval*:

prime $p \implies \text{vpGge } p \ P \ 0 \implies \text{vge } p \ x \ 0 \implies$
 $\text{vge } p \ (\text{poly } P \ x) \ 0$ **unfolding** *poly-altdef*
by (auto *simp*: *vpGge-iff intro!*: *vge-sum vge-mul*[**where** $a = 0$ **and** $b = 0$,
simplified] *vge-pow*)

1.4 Valuations of powers and products

lemma *vp-power-rat*:

assumes *prime* $p \ x \neq 0$
shows $\text{vp } p \ (x \wedge^n) = \text{int } n * \text{vp } p \ x$
proof –
have pe : *prime-elem* (*int* p) **using** *assms*(1) **by** (*simp add*: *prime-nat-int-transfer*)
show *?thesis*
by (*induction* n) (use *assms*(2) pe **in** $\langle \text{simp-all add: vp-def qmultiplicity-mult algebra-simps} \rangle$)
qed

lemma *vp-powi-rat*:

assumes *prime* $p \ x \neq 0$
shows $\text{vp } p \ (x \text{ powi } k) = k * \text{vp } p \ x$
proof (*cases* $0 \leq k$)
case *True*
thus *?thesis* **using** *vp-power-rat*[*OF assms*, *of nat* k] **by** (*simp add*: *power-int-def*)
next
case *False*
have $\text{vp } p \ (x \text{ powi } k) = \text{vp } p \ (\text{inverse } x \wedge^{\text{nat } (-k)})$ **using** *False* **by** (*simp add*:
power-int-def)
also have $\dots = \text{int } (\text{nat } (-k)) * \text{vp } p \ (\text{inverse } x)$
using *assms* **by** (*intro vp-power-rat*) *auto*
finally show *?thesis* **using** *False* **by** (*simp add*: *vp-def qmultiplicity-inverse*)
qed

lemma *vp-prod-rat*:

assumes *prime* p *finite* $I \ \bigwedge i. i \in I \implies f \ i \neq 0$
shows $\text{vp } p \ (\prod i \in I. f \ i) = (\sum i \in I. \text{vp } p \ (f \ i))$
using *assms*(2,3)
proof (*induction* I *rule*: *finite-induct*)
case (*insert* $x \ F$)
have pe : *prime-elem* (*int* p) **using** *assms*(1) **by** (*simp add*: *prime-nat-int-transfer*)
have $(\prod i \in F. f \ i) \neq 0$ **using** *insert* **by** *simp*

thus ?case **using** insert pe **by** (simp add: vp-def qmultiplicity-mult)
qed (simp add: vp-def)

lemma vpGge-C-vp: vpGge p [:c:] (of-int (vp p c))
by (rule vpGge-C) (simp add: vge-def)

Composition with powers and with the identity polynomial pX .

lemma pcompose-power-left: pcompose (p ^ n) q = (pcompose p q) ^ n **for** p q ::
 rat poly
by (induction n) (simp-all add: pcompose-mult pcompose-1)

lemma pcompose-pX: pcompose pX q = q
by (simp add: monom-Suc pcompose-pCons pcompose-1)

$v_p(1/j) \geq -\lfloor \log_p j \rfloor$.

lemma vge-inv-nat: prime p $\implies 1 \leq j \implies$
 vge p (1 / of-nat j) (- of-nat (nat-log p j))

proof -

assume p: prime p **and** j: $1 \leq j$
have p ^ multiplicity p j dvd j **by** (rule multiplicity-dvd)
hence p ^ multiplicity p j $\leq j$ **using** j **by** (intro dvd-imp-le) auto
hence Suc (multiplicity p j) $\leq \text{floorlog } p j$
by (intro floorlog-geI) (use prime-gt-1-nat[OF p] **in** auto)
hence multiplicity p j $\leq \text{nat-log } p j$ **by** (simp add: nat-log-def)
thus ?thesis **by** (simp add: vge-def vp-def inverse-eq-divide[symmetric] multiplicity-int)
qed

lemma nat-log-mono: $m \leq n \implies \text{nat-log } p m \leq \text{nat-log } p n$
by (simp add: nat-log-def floorlog-mono diff-le-mono)

lemma vge-prod0: prime p $\implies \text{finite } I \implies$
 $(\bigwedge i. i \in I \implies \text{vge } p (f i) 0) \implies$
 $\text{vge } p (\prod i \in I. f i) 0$

proof -

assume p: prime p **and** I: finite I **and** hf: $\bigwedge i. i \in I \implies \text{vge } p (f i) 0$
show ?thesis **using** I hf
proof (induction I rule: finite-induct)
case empty
show ?case **using** vge-one **by** simp
next
case (insert x F)
have vge p (f x * ($\prod i \in F. f i$)) (0 + 0) **using** vge-mul[OF p] insert **by** blast
thus ?case **using** insert **by** simp
qed
qed

Strict ultrametric inequality: $v(a) < v(u) \implies v(a + u) = v(a)$.

```

lemma vp-add-eq-of-lt:
  prime p  $\implies a \neq 0 \implies u \neq 0 \implies$ 
     $vp\ p\ a < vp\ p\ u \implies a + u \neq 0 \wedge vp\ p\ (a + u) = vp\ p\ a$ 
proof –
  assume p: prime p and a:  $a \neq 0$  and u:  $u \neq 0$  and lt:  $vp\ p\ a < vp\ p\ u$ 
  have ne:  $a + u \neq 0$ 
  proof
    assume  $a + u = 0$ 
    hence  $-a = u$  by (simp only: neg-eq-iff-add-eq-0)
    hence  $vp\ p\ u = vp\ p\ a$  by (auto simp: vp-def qmultiplicity-minus)
    thus False using lt by simp
  qed
  have  $\min\ (vp\ p\ a)\ (vp\ p\ u) \leq vp\ p\ (a + u)$  using qmultiplicity-add-ge[OF p a u ne]
  by (simp add: vp-def)
  moreover have  $\min\ (vp\ p\ (a + u))\ (vp\ p\ u) \leq vp\ p\ a$ 
  using qmultiplicity-add-ge[of p a + u - u] p a u ne
  by (simp add: vp-def qmultiplicity-minus)
  ultimately show ?thesis using lt ne by (auto simp: min-def split: if-splits)
qed

end

theory Vandermonde-Gram-Determinants
imports Jordan-Normal-Form.Determinant
begin

```

2 Vandermonde and Gram determinants for JNF matrices

The Vandermonde determinant $\det(x_i^j) = \prod_{i < j} (x_j - x_i)$ for the matrices of the `Jordan_Normal_Form` library (HOL-Analysis has it only for its own matrix type, and the AFP not at all), the factorisation of row scalings out of a determinant, and the positivity of the determinant of a positive definite (Gram) matrix, proved by a homotopy to the identity.

Vandermonde determinant for `Jordan_Normal_Form` matrices (absent from HOL and the AFP).

```

lemma det-diag-scale-rows:
  fixes d y :: nat  $\Rightarrow$  'a :: comm-ring-1
  shows Determinant.det (Matrix.mat n n ( $\lambda(i, j). d\ i * y\ i \wedge j$ ))
    = ( $\prod\ i < n. d\ i$ ) * Determinant.det (Matrix.mat n n ( $\lambda(i, j). y\ i \wedge j$ ))
proof –
  define D where  $D = \text{Matrix.mat } n\ n\ (\lambda(i, j). \text{if } i = j \text{ then } d\ i \text{ else } (0 :: 'a))$ 
  define V where  $V = \text{Matrix.mat } n\ n\ (\lambda(i, j). y\ i \wedge j)$ 
  have Dc:  $D \in \text{carrier-mat } n\ n$  and Vc:  $V \in \text{carrier-mat } n\ n$ 
  by (simp-all add: D-def V-def)

```



```

have Matrix.mat n n (λ(i, j). d i * y i ^ j) = D * V
proof (rule eq-matI)
  fix i j assume ij: i < dim-row (D * V) j < dim-col (D * V)
  hence i: i < n and j: j < n by (simp-all add: D-def V-def)
  have (D * V) $$ (i, j) = (∑ k ∈ {0..<n}. (if i = k then d i else 0) * y k ^ j)
    using i j by (simp add: D-def V-def scalar-prod-def)
  also have ... = d i * y i ^ j using i
    by (simp add: if-distrib[where f = λz. z * -] cong: if-cong)
  finally show Matrix.mat n n (λ(i, j). d i * y i ^ j) $$ (i, j) = (D * V) $$ (i,
j)
    using i j by simp
qed (simp-all add: D-def V-def)
hence Determinant.det (Matrix.mat n n (λ(i, j). d i * y i ^ j)) =
  Determinant.det D * Determinant.det V
  using det-mult[OF Dc Vc] by simp
also have Determinant.det D = (∏ i < n. d i)
proof -
  have upper-triangular D by (auto simp: D-def upper-triangular-def)
  hence Determinant.det D = prod-list (diag-mat D) by (rule det-upper-triangular[OF
- Dc])
  also have ... = (∏ i < n. d i)
    unfolding prod-list-diag-prod by (rule prod.cong) (auto simp: D-def)
  finally show ?thesis .
qed
finally show ?thesis by (simp add: V-def)
qed

lemma det-vandermonde-step:
  fixes x :: nat ⇒ 'a :: comm-ring-1
  shows Determinant.det (Matrix.mat (Suc n) (Suc n) (λ(i, j). x i ^ j))
    = (∏ i < n. x (Suc i) - x 0) * Determinant.det (Matrix.mat n n (λ(i,
j). x (Suc i) ^ j))
proof -
  define m where m = Suc n
  define V where V = Matrix.mat m m (λ(i, j). x i ^ j)
  define U where U = Matrix.mat m m (λ(i, j). (if i = j then 1 else 0) + (if
Suc i =
  j then - x 0 else (0 :: 'a)))
  define W where W = V * U
  have Vc: V ∈ carrier-mat m m and Uc: U ∈ carrier-mat m m
    by (simp-all add: V-def U-def)
  have Wc: W ∈ carrier-mat m m using Vc Uc by (simp add: W-def)
  have dU: Determinant.det U = 1
  proof -
    have upper-triangular U by (auto simp: U-def upper-triangular-def)
    hence Determinant.det U = prod-list (diag-mat U) by (rule det-upper-triangular[OF
- Uc])
    also have ... = 1 unfolding prod-list-diag-prod by (rule prod.neutral) (auto
simp: U-def)

```

finally show ?thesis .
 qed
 have $dW: \text{Determinant.det } W = \text{Determinant.det } V$ unfolding $W\text{-def det-mult}[OF$
 $Vc\ Uc]$ dU by simp
 have $Wij: W \ \$\$ (i, j) = (\text{if } j = 0 \text{ then } 1 \text{ else } x\ i \wedge j - x\ 0 * x\ i \wedge (j - 1))$
 if $i < m$ and $j < m$ for $i\ j$
 proof -
 have $W \ \$\$ (i, j) = (\sum k \in \{0..<m\}. x\ i \wedge k * ((\text{if } k = j \text{ then } 1 \text{ else } 0) + (\text{if}$
 $Suc\ k =$
 $j \text{ then } - x\ 0 \text{ else } 0)))$
 using $i\ j$ by (simp add: $W\text{-def } V\text{-def } U\text{-def scalar-prod-def}$)
 also have $\dots = (\sum k \in \{0..<m\}.$
 $\text{if } k = j \text{ then } x\ i \wedge k \text{ else } 0) + (\sum k \in \{0..<m\}.$
 $\text{if } Suc\ k = j \text{ then } - (x\ 0 * x\ i \wedge k) \text{ else } 0)$
 by (simp add: $sum.distrib\ distrib\text{-left}\ \text{if}\text{-distrib}$ [where $f =$
 $\lambda z. - * z]$ $mult.commute\ cong: \text{if}\text{-cong}$)
 also have $\dots = (\text{if } j = 0 \text{ then } 1 \text{ else } x\ i \wedge j - x\ 0 * x\ i \wedge (j - 1))$
 proof (cases j)
 case 0 thus ?thesis using j by simp
 next
 case (Suc j')
 thus ?thesis using j by simp
 qed
 finally show ?thesis .
 qed
 have $\text{Determinant.det } W = (\sum j < m. W \ \$\$ (0, j) * \text{cofactor } W\ 0\ j)$
 by (rule $\text{laplace-expansion-row}[OF\ Wc]$) (simp add: $m\text{-def}$)
 also have $\dots = \text{cofactor } W\ 0\ 0$
 unfolding $m\text{-def sum.lessThan-Suc-shift}$ using $Wij[of\ 0]$ by (simp add: $m\text{-def}$
 power-Suc)
 also have $\dots = \text{Determinant.det } (\text{mat-delete } W\ 0\ 0)$ by (simp add: cofactor-def)
 also have $\text{mat-delete } W\ 0\ 0 = \text{Matrix.mat } n\ n\ (\lambda(i, j). (x\ (Suc\ i) - x\ 0) * x$
 $(Suc\ i) \wedge j)$
 proof (rule eq-matI)
 fix $i\ j$ assume $i < \text{dim-row } (\text{Matrix.mat } n\ n\ (\lambda(i,$
 $j). (x\ (Suc\ i) - x\ 0) * x\ (Suc\ i) \wedge j))$
 $j < \text{dim-col } (\text{Matrix.mat } n\ n\ (\lambda(i, j). (x\ (Suc\ i) - x\ 0) * x\ (Suc\ i) \wedge j))$
 hence $ij: i < n\ j < n$ by simp-all
 have $\text{mat-delete } W\ 0\ 0 \ \$\$ (i, j) = W \ \$\$ (Suc\ i, Suc\ j)$
 using $ij\ Wc$ by (simp add: $\text{mat-delete-def } m\text{-def}$)
 also have $\dots = (x\ (Suc\ i) - x\ 0) * x\ (Suc\ i) \wedge j$ using ij
 by (simp add: $Wij\ m\text{-def algebra-simps}$)
 finally show $\text{mat-delete } W\ 0\ 0 \ \$\$ (i, j) =$
 $\text{Matrix.mat } n\ n\ (\lambda(i, j). (x\ (Suc\ i) - x\ 0) * x\ (Suc\ i) \wedge j) \ \$\$ (i, j)$
 using ij by simp
 qed (use Wc in <simp-all add: $m\text{-def}$ >)
 also have $\text{Determinant.det } \dots = (\prod i <$
 $n. x\ (Suc\ i) - x\ 0) * \text{Determinant.det } (\text{Matrix.mat } n\ n\ (\lambda(i, j). x\ (Suc\ i) \wedge j))$
 by (rule $\text{det-diag-scale-rows}$)

finally show *?thesis* **using** *dW* **by** (*simp add: V-def m-def*)
qed

lemma *det-vandermonde-jnf*:

Determinant.det (Matrix.mat n n (λ(i, j). (x i :: 'a :: comm-ring-1) ^ j))
 $= (\prod i < n. \prod j \in \{j. i < j \wedge j < n\}. x j - x i)$

proof (*induction n arbitrary: x*)

case *0*

thus *?case* **by** (*simp add: det-def'[of - 0]*)

next

case (*Suc n*)

have *s1*: $\{j. 0 < j \wedge j < \text{Suc } n\} = \text{Suc } \{j. j < n\}$

by (*auto simp: image-iff*) (*metis Suc-pred lessThan-iff less-Suc-eq-0-disj not-less-eq*)

have *s2*: $\{j. \text{Suc } i < j \wedge j < \text{Suc } n\} = \text{Suc } \{j. i < j \wedge j < n\}$ **for** *i*

by (*auto simp: image-iff*) (*metis Suc-lessE less-Suc-eq-0-disj not-less-eq*)

have $(\prod i < \text{Suc } n. \prod j \in \{j. i < j \wedge j < \text{Suc } n\}. x j - x i)$

$= (\prod j \in \{j. 0 < j \wedge j < \text{Suc } n\}. x j - x 0) * (\prod i < n. \prod j \in \{j. \text{Suc } i < j \wedge j < \text{Suc } n\}. x j - x (\text{Suc } i))$

by (*simp only: prod.lessThan-Suc-shift*)

also have $\dots = (\prod i < n. x (\text{Suc } i) - x 0) * (\prod i < n. \prod j \in \{j. i < j \wedge j < n\}. x (\text{Suc } j) - x (\text{Suc } i))$

unfolding *s1 s2* **by** (*simp add: prod.reindex*)

finally show *?case* **unfolding** *det-vandermonde-step Suc[of λi. x (Suc i)]* **by**

simp

qed

A Gram matrix of linearly independent functions has positive determinant
(used for $\Delta\text{-}K(\zeta(5)) > 0$ unless the Andreief route is taken).

lemma *posdef-det-nonzero*:

fixes *B* :: *real mat*

assumes *B*: *B* ∈ *carrier-mat n n*

and *pd*: $\bigwedge c :: \text{nat} \Rightarrow \text{real}. (\exists i < n. c \ i \neq 0) \Rightarrow$

$0 < (\sum i < n. \sum j < n. c \ i * B \ \$\$ (i, j) * c \ j)$

shows *Determinant.det B* ≠ 0

proof

assume *Determinant.det B* = 0

then obtain *v* **where** *v*: *v* ∈ *carrier-vec n* *v* ≠ 0_{*v*} *n* *B* *_{*v*} *v* = 0_{*v*} *n*

using *det-0-iff-vec-prod-zero[OF B]* **by** *blast*

define *c* **where** *c* *i* = (*if* *i* < *n* *then* *v* \$ *i* *else* 0) **for** *i*

have $\exists i < n. c \ i \neq 0$

proof (*rule ccontr*)

assume $\neg (\exists i < n. c \ i \neq 0)$

hence *v* = 0_{*v*} *n* **using** *v(1)* **by** (*intro eq-vecI*) (*auto simp: c-def*)

thus *False* **using** *v(2)* **by** *simp*

qed

hence *pos*: $0 < (\sum i < n. \sum j < n. c \ i * B \ \$\$ (i, j) * c \ j)$ **by** (*rule pd*)

have *row*: $(\sum j < n. B \ \$\$ (i, j) * c \ j) = 0$ **if** *i*: *i* < *n* **for** *i*

```

proof –
  have  $(B *_v v) \$ i = 0$  using  $v(3) i$  by simp
  moreover have  $(B *_v v) \$ i = (\sum j < n. B \$\$ (i, j) * c j)$ 
    using  $i B v(1)$  by (simp add: index-mult-mat-vec scalar-prod-def c-def
lessThan-atLeast0)
  ultimately show ?thesis by simp
qed
have  $(\sum i < n. \sum j < n. c i * B \$\$ (i, j) * c j) = (\sum i < n. c i * (\sum j < n. B \$\$ (i, j) * c j))$ 
  by (simp add: sum-distrib-left mult.assoc)
also have  $\dots = 0$  by (simp add: row)
finally show False using pos by simp
qed

```

lemma *posdef-det-pos*:

```

assumes  $A \in \text{carrier-mat } n \ n$ 
  and  $\bigwedge c :: \text{nat} \Rightarrow \text{real}. (\exists i < n. c i \neq 0) \Rightarrow$ 
     $0 < (\sum i < n. \sum j < n. c i * A \$\$ (i, j) * c j)$ 
  shows  $0 < \text{Determinant.det } A$ 
proof (rule ccontr)
  assume neg:  $\neg 0 < \text{Determinant.det } A$ 
  define  $B$  where  $B t = \text{Matrix.mat } n \ n (\lambda(i, j). t * A \$\$ (i, j) + (1 - t) * (\text{if } i$ 
     $=$ 
     $j \text{ then } 1 \text{ else } 0))$  for  $t :: \text{real}$ 
  define  $f$  where  $f t = \text{Determinant.det } (B t)$  for  $t$ 
  have  $Bc$ :  $B t \in \text{carrier-mat } n \ n$  for  $t$  by (simp add: B-def)
  have  $B0$ :  $B 0 = 1_m \ n$  by (rule eq-matI) (auto simp: B-def)
  have  $B1$ :  $B 1 = A$  by (rule eq-matI) (use assms(1) in <auto simp: B-def>)
  have  $f0$ :  $f 0 = 1$  by (simp add: f-def B0)
  have  $f1$ :  $f 1 \leq 0$  using neg by (simp add: f-def B1)
  have  $fne$ :  $f t \neq 0$  if  $t: 0 \leq t \leq 1$  for  $t$ 
    unfolding f-def
  proof (rule posdef-det-nonzero[OF Bc])
    fix  $c :: \text{nat} \Rightarrow \text{real}$  assume  $c: \exists i < n. c i \neq 0$ 
    define  $QA$  where  $QA = (\sum i < n. \sum j < n. c i * A \$\$ (i, j) * c j)$ 
    define  $S$  where  $S = (\sum i < n. c i * c i)$ 
    have  $QA$ :  $0 < QA$  unfolding QA-def by (rule assms(2)[OF c])
    have  $S$ :  $0 < S$ 
    proof –
      obtain  $i$  where  $i: i < n \ c i \neq 0$  using  $c$  by blast
      have  $c i * c i \leq S$  unfolding S-def using  $i(1)$ 
        by (intro member-le-sum) auto
      moreover have  $0 < c i * c i$  using  $i(2)$  not-real-square-gt-zero by blast
      ultimately show ?thesis by linarith
    qed
  have  $(\sum i < n. \sum j < n. c i * B t \$\$ (i, j) * c j)$ 
     $= (\sum i < n. \sum j < n. t * (c i * A \$\$ (i, j) * c j) + (\text{if } i =$ 
     $j \text{ then } (1 - t) * (c i * c j) \text{ else } 0))$ 
    by (intro sum.cong refl) (auto simp: B-def algebra-simps)

```

```

also have ... = t * QA + (1 - t) * S
  by (simp add: sum.distrib sum-distrib-left QA-def S-def sum.delta')
finally have e: (∑ i < n. ∑ j < n. c i * B t $$ (i, j) * c j) =
  t * QA + (1 - t) * S .
have 0 < t * QA + (1 - t) * S
proof (cases t = 0)
  case True thus ?thesis using S by simp
next
  case False
  hence 0 < t * QA using t QA by simp
  moreover have 0 ≤ (1 - t) * S using t S by simp
  ultimately show ?thesis by linarith
qed
thus 0 < (∑ i < n. ∑ j < n. c i * B t $$ (i, j) * c j) unfolding e .
qed
have fdef: f t = (∑ p | p permutes {0..<n}. of-int (Permutations.sign p) *
  (∏ i = 0..<n. t * A $$ (i, p i) + (1 - t) * (if i = p i then 1 else 0)))
  for t
  unfolding f-def det-def'[OF Bc]
proof (intro sum.cong refl arg-cong[where f = λx. - * x] prod.cong)
  fix p i assume p ∈ {p. p permutes {0..<n}} i ∈ {0..<n}
  hence p i < n i < n by (auto simp: permutes-in-image)
  thus B t $$ (i, p i) = t * A $$ (i, p i) + (1 - t) * (if i = p i then 1 else 0)
  by (simp add: B-def)
qed
have cont: continuous-on {0..1} f
  unfolding fdef[abs-def] by (intro continuous-intros)
obtain x where 0 ≤ x x ≤ 1 f x = 0
  using IVT2[of f 1 0 0, OF f1 - - cont] f0 by auto
thus False using fne by blast
qed
end

theory Polynomial-Taylor-Shift
  imports Rational-Padic-Valuations
begin

```

3 Taylor coefficients of a rational polynomial at a point

$\text{tay } P \ r \ i$ is the coefficient of X^i in $P(X + r)$. Linearity, the binomial formula for monomials, the behaviour under multiplication by $(X - r)^n$, and p -integrality of the Taylor coefficients of a p -integral polynomial at a p -integral point.

The i -th Taylor coefficient of P at r : $P(X + r) = \sum -i \text{ tay } P \ r \ i \ X^{\wedge}i$.

definition $\text{tay} :: \text{rat poly} \Rightarrow \text{rat} \Rightarrow \text{nat} \Rightarrow \text{rat}$ **where**
 $\text{tay } P \ r \ i = \text{Polynomial.coeff } (\text{pcompose } P \ (pX + [:r:])) \ i$

lemma tay-add : $\text{tay } (P + Q) \ r \ i = \text{tay } P \ r \ i + \text{tay } Q \ r \ i$
by ($\text{simp add: tay-def pcompose-add}$)

lemma tay-smult : $\text{tay } (\text{Polynomial.smult } c \ P) \ r \ i = c * \text{tay } P \ r \ i$
by ($\text{simp add: tay-def pcompose-smult}$)

lemma tay-shift-lib : $\text{pcompose } ((pX - [:r:]) ^ n * Q) \ (pX + [:r:]) =$
 $\text{Polynomial.monom } 1 \ n * \text{pcompose } Q \ (pX + [:r:])$
proof –
have a : $\text{pcompose } (pX - [:r:]) \ (pX + [:r:]) = pX$ **using** $\text{pcompose-diff pcompose-pose-pX pcompose-const}$
by ($\text{simp add: pcompose-diff pcompose-pX}$)
have b : $(pX :: \text{rat poly}) ^ n = \text{Polynomial.monom } 1 \ n$ **by** ($\text{simp add: monom-power}$)
show $?thesis$ **unfolding** $\text{pcompose-mult pcompose-power-left a b}$ **by** simp
qed

lemma tay-sum : $\text{tay } (\sum d \in I. f \ d) \ r \ i = (\sum d \in I. \text{tay } (f \ d) \ r \ i)$
by ($\text{simp add: tay-def pcompose-sum coeff-sum}$)

lemma tay-pX-pow : $\text{tay } (pX ^ d) \ r \ i = (\text{if } i \leq d \text{ then of-nat } (d \text{ choose } i) * r ^ (d - i) \text{ else } 0)$
proof –
have c : $pX + [:r:] = [:r, 1:]$ **by** ($\text{simp add: monom-Suc}$)
have e : $\text{tay } (pX ^ d) \ r \ i = \text{Polynomial.coeff } ([:r, 1:] ^ d) \ i$
unfolding $\text{tay-def pcompose-power-left pcompose-pX c}$ **by** simp
show $?thesis$
proof ($\text{cases } i \leq d$)
case True
thus $?thesis$ **unfolding** e **using** $\text{coeff-linear-poly-power}[OF \ \text{True}, \text{ of } r \ 1 :: \text{rat}]$
by simp
next
case False
have dg : $\text{degree } ([:r, 1:] ^ d) < i$ **using** $\text{degree-power-le}[of \ [:r, 1:] \ d] \ \text{False}$ **by** simp
have $\text{Polynomial.coeff } ([:r, 1:] ^ d) \ i = 0$ **using** $\text{coeff-eq-0}[OF \ dg]$ **by** blast
thus $?thesis$ **unfolding** e **using** False **by** simp
qed
qed

lemma tay-eq-sum : $\text{degree } P < h \implies$
 $\text{tay } P \ r \ i = (\sum d < h. \text{Polynomial.coeff } P \ d * \text{tay } (pX ^ d) \ r \ i)$
proof –
assume dg : $\text{degree } P < h$
have m : $\text{Polynomial.monom } c \ d = \text{Polynomial.smult } c \ (pX ^ d)$ **for** $c :: \text{rat}$ **and** d
using $\text{smult-monom monom-power}$ **by** ($\text{simp add: smult-monom monom-power}$)

```

have s1: (∑ d ≤ h - 1. Polynomial.monom (Polynomial.coeff P d) d) = P
  using poly-as-sum-of-monom's[of P h - 1] dg by simp
have s2: {..h - 1} = {..

```

```

lemma tay-0-eq-poly: tay P r 0 = poly P r
  unfolding tay-def using poly-0-coeff-0[symmetric] poly-pcompose poly-monom
by simp

```

$(X - r)^{\wedge n} Q$ has vanishing Taylor coefficients of order $< n$ at r , and the shifted ones are those of Q .

```

lemma tay-pow-mult: tay ((pX - [:r:]) ^ n * Q) r (n + i) = tay Q r i
  unfolding tay-def tay-shift-lib coeff-monom-mult by simp

```

```

lemma tay-pow-mult-lt: i < n ⟹ tay ((pX - [:r:]) ^ n * Q) r i = 0
  unfolding tay-def tay-shift-lib coeff-monom-mult by simp

```

```

lemma vpGge-pcompose: prime p ⟹ vpGge p P 0 ⟹
  vpGge p Q 0 ⟹ vpGge p (pcompose P Q) 0

```

proof –

```

assume p: prime p and hP: vpGge p P 0 and hQ: vpGge p Q 0
show ?thesis using hP
proof (induction P rule: pCons-induct)
  case 0
  show ?case using vpGge-zero by simp
next
  case (pCons a P)
  have vge p (Polynomial.coeff (pCons a P) 0) 0 using pCons.prem's unfolding
vpGge-iff by blast
  hence a: vge p a 0 by simp
  have vpGge p P 0 unfolding vpGge-iff
  proof
    fix k
    have vge p (Polynomial.coeff (pCons a P) (Suc k)) 0 using pCons.prem's
unfolding vpGge-iff
    by blast
    thus vge p (Polynomial.coeff P k) 0 by simp
  qed
qed

```

```

    hence vpGge p (pcompose P Q) 0 using pCons.IH by simp
    hence vpGge p (Q * pcompose P Q) (0 + 0) by (rule vpGge-mul[OF p hQ])
    hence vpGge p ([:a:] + Q * pcompose P Q) 0 using vpGge-add[OF p vpGge-C[OF
a]] by simp
    thus ?case using pcompose-pCons by metis
qed
qed

```

```

lemma vge-tay: prime p  $\implies$  vpGge p P 0  $\implies$  vge p r 0  $\implies$ 
vge p (tay P r i) 0

```

```

proof -
  assume p: prime p and hP: vpGge p P 0 and hr: vge p r 0
  have X0: vpGge p (pX :: rat poly) 0 unfolding vpGge-iff using vge-one vge-zero
  by (simp add: coeff-monom)
  have vpGge p (pX + [:r:]) 0 using vpGge-add[OF p X0 vpGge-C[OF hr]] by
simp
  hence vpGge p (pcompose P (pX + [:r:])) 0 by (rule vpGge-pcompose[OF p hP])
  thus ?thesis unfolding tay-def vpGge-iff by simp
qed

```

```

end

```

```

theory Unimodular-Determinants
imports Rational-Padic-Valuations
begin

```

4 Determinants of p -integral matrices and a unimodularity criterion

The determinant of a matrix with p -integral entries is p -integral; modulo p the determinant of a matrix that is lower triangular modulo p is the product of its diagonal; hence if $M V = T$ with M, V p -integral and T lower triangular modulo p with unit diagonal, then $\det M$ is a p -adic unit. This is the tool behind the unimodularity of the inner and outer bases of the irrationality proof, but it is stated for arbitrary matrices.

The determinant of a matrix with p -integral entries is p -integral.

lemma *vge-det-integral*:

```

assumes prime p A  $\in$  carrier-mat n n  $\bigwedge i j. i < n \implies$ 
j < n  $\implies$  vge p (A $$ (i, j)) 0
shows vge p (Determinant.det A) 0
proof -
  have p: prime p and A: A  $\in$  carrier-mat n n and Ai:  $\bigwedge i j. i < n \implies$ 
j < n  $\implies$  vge p (A $$ (i, j)) 0
  using assms by auto
  have d: Determinant.det A =  $(\sum \sigma \in$ 
 $\{\sigma. \sigma \text{ permutes } \{0..<n\}\}. \text{signof } \sigma * (\prod i =$ 

```



```

    0..<n. A $$ (i, σ i)))
  using det-def'[OF A] by simp
have vge p (∑ σ ∈ {σ. σ permutes {0..<n}}. signof σ *
  (∏ i = 0..<n. A $$ (i, σ i))) 0
proof (rule vge-sum[OF p])
  show finite {σ. σ permutes {0..<n}} using finite-permutations by blast
  fix σ assume σ ∈ {σ. σ permutes {0..<n}}
  hence s: σ permutes {0..<n} by simp
  have sg: vge p (signof σ) 0 using vge-of-int[OF p] by simp
  have pr: vge p (∏ i = 0..<n. A $$ (i, σ i)) 0
    using vge-prod0[OF p, of {0..<n} λi. A $$ (i,
      σ i)] Ai permutes-in-image[OF s] by simp
  show vge p (signof σ * (∏ i = 0..<n. A $$ (i, σ i))) 0
    using vge-mul[OF p sg pr] by simp
qed
thus ?thesis using d by simp
qed

```

A permutation of $\{0..<n\}$ that is not the identity moves some i downwards (stated for *nat*; a version with an unaxiomatised *ord* type would be false).

lemma *permutes-not-id-exists-lt-nat*:

```

  fixes σ :: nat ⇒ nat
  assumes σ permutes {0..<n} σ ≠ id
  shows ∃ i < n. σ i < i
proof (rule ccontr)
  assume nl: ¬ (∃ i < n. σ i < i)
  have s: σ permutes {0..<n} using assms(1) .
  have le: ∀ i ∈ {0..<n}. i ≤ σ i using nl by force
  have sm: (∑ i ∈ {0..<n}. i) = (∑ i ∈ {0..<n}. σ i)
    using sum.permute[OF s, of λi. i] by (simp add: o-def)
  have eq: ∀ i ∈ {0..<n}. σ i = i
  proof (rule ccontr)
    assume ¬ (∀ i ∈ {0..<n}. σ i = i)
    then obtain j where j: j ∈ {0..<n} σ j ≠ j by blast
    have ∃ a ∈ {0..<n}. a < σ a using j le by force
    hence (∑ i ∈ {0..<n}. i) < (∑ i ∈ {0..<n}. σ i)
      using sum-strict-mono-ex1[of {0..<n} λi. i σ] le by blast
    thus False using sm by linarith
  qed
  have σ = id
  proof (rule ext)
    fix i
    show σ i = id i using eq permutes-not-in[OF s, of i] by auto
  qed
  thus False using assms(2) by simp
qed

```

Modulo p , the determinant of a matrix whose entries below the diagonal are divisible by p is the product of the diagonal.

lemma *det-cong-diag*:

assumes p : prime p **and** T : $T \in \text{carrier-mat } n \ n$
and int : $\bigwedge i \ j. i < n \implies j < n \implies \text{vge } p \ (T \ \$\$ (i, j)) \ 0$
and low : $\bigwedge i \ j. j < i \implies i < n \implies \text{vge } p \ (T \ \$\$ (i, j)) \ 1$
shows $\text{vge } p \ (\text{Determinant.det } T - (\prod i < n. T \ \$\$ (i, i))) \ 1$
proof –
let $?S = \{\sigma. \sigma \text{ permutes } \{0..<n\}\}$
let $?f = \lambda \sigma. \text{signof } \sigma * (\prod i = 0..<n. T \ \$\$ (i, \sigma i))$
have fS : $\text{finite } ?S$ **using** *finite-permutations* **by** *blast*
have idS : $\text{id} \in ?S$ **using** *permutes-id* **by** *simp*
have $\text{Determinant.det } T = ?f \ \text{id} + (\sum \sigma \in ?S - \{\text{id}\}. ?f \ \sigma)$
using *det-def* $[OF \ T] \text{sum.remove} [OF \ fS \ \text{idS}, \text{of } ?f]$ **by** *simp*
moreover **have** $?f \ \text{id} = (\prod i < n. T \ \$\$ (i, i))$ **using** *sign-id atLeast0LessThan*
by *(simp add: atLeast0LessThan)*
moreover **have** $\text{vge } p \ (\sum \sigma \in ?S - \{\text{id}\}. ?f \ \sigma) \ 1$
proof *(rule vge-sum [OF p])*
show $\text{finite } (?S - \{\text{id}\})$ **using** fS **by** *simp*
fix σ **assume** $\sigma \in ?S - \{\text{id}\}$
hence s : $\sigma \text{ permutes } \{0..<n\}$ **and** ns : $\sigma \neq \text{id}$ **by** *auto*
obtain i **where** i : $i < n \ \sigma i < i$ **using** *permutes-not-id-exists-lt-nat* $[OF \ s \ ns]$
by *blast*
have si : $\bigwedge j. j < n \implies \sigma j < n$ **using** *permutes-in-image* $[OF \ s]$
by *simp*
have pr : $(\prod j = 0..<n. T \ \$\$ (j, \sigma j)) = T \ \$\$ (i, \sigma i) * (\prod j \in \{0..<n\} - \{i\}. T \ \$\$ (j, \sigma j))$
using *prod.remove* $[of \ \{0..<n\} \ i \ \lambda j. T \ \$\$ (j, \sigma j)] \ i$ **by** *simp*
have $v1$: $\text{vge } p \ (T \ \$\$ (i, \sigma i)) \ 1$ **using** *low* $[OF \ i(2) \ i(1)]$ **by** *simp*
have $\text{vge } p \ (\prod j \in \{0..<n\} - \{i\}. T \ \$\$ (j, \sigma j)) \ 0$
using *vge-prod0* $[OF \ p, \text{of } \{0..<n\} - \{i\} \ \lambda j. T \ \$\$ (j, \sigma j)] \ \text{int } si$ **by** *simp*
hence v : $\text{vge } p \ (T \ \$\$ (i, \sigma i) * (\prod j \in \{0..<n\} - \{i\}. T \ \$\$ (j, \sigma j))) \ (1 + 0)$ **by** *(rule vge-mul [OF p v1])*
have sg : $\text{vge } p \ (\text{signof } \sigma) \ 0$ **using** *vge-of-int* $[OF \ p]$ **by** *simp*
have $\text{vge } p \ (\text{signof } \sigma * (\prod j = 0..<n. T \ \$\$ (j, \sigma j))) \ (0 + (1 + 0))$
unfolding pr **by** *(rule vge-mul [OF p sg v])*
thus $\text{vge } p \ (\text{signof } \sigma * (\prod i = 0..<n. T \ \$\$ (i, \sigma i))) \ 1$ **by** *simp*
qed
ultimately show $?thesis$ **by** *simp*
qed

The unimodularity criterion: if $M \ V = T$ with M, V p -integral and T upper triangular modulo p with unit diagonal, then $\text{det } M$ is a p -adic unit.

lemma *unimodular-criterion*:

assumes p : prime p **and** M : $M \in \text{carrier-mat } n \ n$ **and** V : $V \in \text{carrier-mat } n \ n$
and Mi : $\bigwedge i \ j. i < n \implies j < n \implies \text{vge } p \ (M \ \$\$ (i, j)) \ 0$
and Vi : $\bigwedge i \ j. i < n \implies j < n \implies \text{vge } p \ (V \ \$\$ (i, j)) \ 0$
and low : $\bigwedge i \ j. j < i \implies i < n \implies \text{vge } p \ ((M * V) \ \$\$ (i, j)) \ 1$
and diag : $\bigwedge i. i < n \implies (M * V) \ \$\$ (i, i) \neq 0 \wedge \text{vp } p \ ((M * V) \ \$\$ (i, i)) = 0$

```

shows Determinant.det M ≠ 0 ∧ vp p (Determinant.det M) = 0
proof -
  define T where T = M * V
  have Tc: T ∈ carrier-mat n n using M V by (simp add: T-def)
  have Ti: ∧i j. i < n ⇒ j < n ⇒ vge p (T $$ (i, j)) 0
  proof -
    fix i j assume i: i < n and j: j < n
    have e: T $$ (i, j) = (∑ k ∈ {0..

```

```

    by (rule vp-add-eq-of-lt[OF p w(1) False])
    thus ?thesis using du w by simp
  qed
qed
have vM: vge p (Determinant.det M) 0 by (rule vge-det-integral[OF p M Mi])
have vV: vge p (Determinant.det V) 0 by (rule vge-det-integral[OF p V Vi])
have dM: Determinant.det M ≠ 0 and dV: Determinant.det V ≠ 0 using dT
detT
  by auto
have pe: prime-elem (int p) using p prime-nat-int-transfer by simp
have s: vp p (Determinant.det T) = vp p (Determinant.det M) + vp p (Determinant.det
V)
  unfolding detT vp-def using qmultiplicity-mult[OF pe dM dV] by simp
have gM: 0 ≤ vp p (Determinant.det M) using vM dM unfolding vge-def by
simp
have gV: 0 ≤ vp p (Determinant.det V) using vV dV unfolding vge-def by
simp
show ?thesis using dM s gM gV dT by simp
qed
end

theory Legendre-Factorial-Valuation
imports Rational-Padic-Valuations Prime-Number-Theorem.Prime-Counting-Functions
begin

```

5 Legendre's formula for the p -adic valuation of $n!$

$v_p(n!) = \sum_{a \geq 1} \lfloor n / p^a \rfloor$, derived from the Legendre identity of the AFP entry `Prime_Number_Theorem`, in the two forms used later: with the *legendre-aux* function of that entry and with an explicit finite range of summation.

Legendre's formula $v_p(n!) = \sum_{a \geq 1} \lfloor n / p^a \rfloor$; exists as *IMO2019-Q4.multiplicity-prime-fact* with *legendre-aux*.

lemma *legendre-fact*:

```

  fixes p :: nat
  assumes prime p
  shows multiplicity p (fact n) = legendre-aux (real n) p
proof (cases p ≤ n)
  case True
  have fact n = (∏ p | prime p ∧ p ≤ n. p ^ legendre-aux n p)
    using legendre-identity'[of real n] by simp
  also have multiplicity p ... = (∑ q | prime q ∧
    q ≤ n. multiplicity p (q ^ legendre-aux n q))
    using assms by (subst prime-elem-multiplicity-prod-distrib) auto
  also have ... = (∑ q ∈ {p}. legendre-aux n q)
    using assms ⟨p ≤ n⟩ prime-multiplicity-other[of p]
    by (intro sum.mono-neutral-cong-right)

```

```

      (auto simp: prime-elem-multiplicity-power-distrib prime-multiplicity-prime
split: if-splits)
    finally show ?thesis by simp
next
  case False
  hence multiplicity p (fact n) = 0
  using assms by (intro not-dvd-imp-multiplicity-0) (auto simp: prime-dvd-fact-iff)
  moreover from False have legendre-aux (real n) p = 0
  by (intro legendre-aux-eq-0) auto
  ultimately show ?thesis by simp
qed

```

Legendre's formula with any upper limit $B \geq n$ on the sum.

```

lemma legendre-sum:
  assumes prime p n ≤ B
  shows multiplicity p (fact n) = (∑ a = 1..B. n div p ^ a)
proof -
  have p2: 2 ≤ p using prime-ge-2-nat[OF assms(1)] .
  have multiplicity p (fact n) = (∑ m | 0 < m ∧
    real (p ^ m) ≤ real n. nat ⌊real n / real (p ^ m)⌋)
  using legendre-fact[OF assms(1), of n] assms(1) by (simp add: legendre-aux-def)
  also have ... = (∑ m | 0 < m ∧ p ^ m ≤ n. n div p ^ m)
  proof (rule sum.cong)
    show {m. 0 < m ∧ real (p ^ m) ≤ real n} = {m. 0 < m ∧ p ^ m ≤ n}
    by (simp only: of-nat-le-iff)
  next
    fix m show nat ⌊real n / real (p ^ m)⌋ = n div p ^ m
    by (simp only: floor-divide-of-nat-eq nat-int)
  qed
  also have ... = (∑ a = 1..B. n div p ^ a)
  proof (rule sum.mono-neutral-left)
    show {m. 0 < m ∧ p ^ m ≤ n} ⊆ {1..B}
    proof
      fix m assume m: m ∈ {m. 0 < m ∧ p ^ m ≤ n}
      have m < 2 ^ m by (rule less-exp)
      also have (2::nat) ^ m ≤ p ^ m using p2 by (rule power-mono) simp
      finally show m ∈ {1..B} using m assms(2) by auto
    qed
  next
    show ∀ i ∈ {1..B} - {m. 0 < m ∧ p ^ m ≤ n}. n div p ^ i = 0
    proof
      fix i assume i ∈ {1..B} - {m. 0 < m ∧ p ^ m ≤ n}
      hence n < p ^ i by auto
      thus n div p ^ i = 0 by (rule div-less)
    qed
  qed simp
  finally show ?thesis .
qed

```

end

theory *Prime-Sum-Limits*
imports *Rational-Padic-Valuations Prime-Number-Theorem.Prime-Number-Theorem*
HOL-Real-Asymp.Real-Asymp
begin

6 Riemann-sum limits of sums over primes from the prime number theorem

From the prime number theorem in the form $\vartheta(x) \sim x$ (the AFP's ϑ -asymptotics): for a bounded function φ on $[a, b]$ that is continuous off a countable set, $\sum_{aK < p \leq bK} \varphi(p/K) \ln p \leq (\int_a^b \varphi + \varepsilon) K$ for all large K , and the version in the variable $x = K/p$. The proof uses Chebyshev's ϑ over the cells of a uniform partition, step majorants of φ and dominated convergence as the mesh goes to zero. Also the bookkeeping $\psi - \vartheta$ for the valuations of small primes.

The prime number theorem in the form used (from *Prime-Number-Theorem. ϑ -asymptotics*).

lemma *pnt-theta*: $((\lambda x. \text{primes-theta } x / x) \longrightarrow 1)$ *at-top*
using ϑ -asymptotics
by (rule *asympt-equivD-strong*[*OF* - *eventually-mono*[*OF* *eventually-gt-at-top*[*of* *0*]]]) *auto*

6.1 Chebyshev's ϑ over cells

$\vartheta(x) = \sum -\{p \leq x\} \log p$ as a sum over a set of naturals.

lemma *theta-eq-sum*: $0 \leq x \implies \text{primes-theta } x = (\sum p \in \{p. \text{prime } p \wedge \text{real } p \leq x\}. \ln (\text{real } p))$
by (*simp add: primes-theta-def prime-sum-upto-def*)

lemma *finite-primes-le*: *finite* $\{p. \text{prime } p \wedge \text{real } p \leq x\}$

proof (rule *finite-subset*[*of* - $\{..nat \lfloor x \rfloor\}$])
show $\{p. \text{prime } p \wedge \text{real } p \leq x\} \subseteq \{..nat \lfloor x \rfloor\}$
by (*auto intro: le-nat-floor*)
qed *simp*

The sum of $\log p$ over the primes in $(c, d]$ is $\vartheta(d) - \vartheta(c)$.

lemma *sum-cell*:
assumes $0 \leq c \leq d$
shows $(\sum p \in \{p. \text{prime } p \wedge c < \text{real } p \wedge \text{real } p \leq d\}. \ln (\text{real } p)) = \text{primes-theta } d - \text{primes-theta } c$
proof -
have $\{p. \text{prime } p \wedge \text{real } p \leq d\} = \{p. \text{prime } p \wedge \text{real } p \leq c\} \cup \{p. \text{prime } p \wedge c < \text{real } p \wedge \text{real } p \leq d\}$

using *assms* **by** *auto*
moreover have $\{p. \text{prime } p \wedge \text{real } p \leq c\} \cap \{p. \text{prime } p \wedge c < \text{real } p \wedge \text{real } p \leq d\} = \{\}$ **by** *auto*
ultimately show *?thesis*
unfolding *theta-eq-sum*[*OF* *assms*(1)] *theta-eq-sum*[*OF* *order-trans*[*OF* *assms*]]
using *sum.union-disjoint*[*of* $\{p. \text{prime } p \wedge \text{real } p \leq c\}$ $\{p. \text{prime } p \wedge c < \text{real } p \wedge \text{real } p \leq d\}$ $\lambda p. \ln (\text{real } p)$]
finite-primes-le[*of* *c*] *finite-subset*[*OF* - *finite-primes-le*[*of* *d*], *of* $\{p. \text{prime } p \wedge c < \text{real } p \wedge \text{real } p \leq d\}$]
by *fastforce*
qed

lemma *theta-ratio-eq*: $0 < c \implies 0 < K \implies$
 $c * (\text{primes-theta } (c * \text{real } K) / (c * \text{real } K)) = \text{primes-theta } (c * \text{real } K) / \text{real } K$
by (*simp add: field-simps*)

$\vartheta(cK)/K \longrightarrow c$ along $K \in \mathbb{N}$.

lemma *theta-tendsto*:
assumes $0 \leq c$
shows $((\lambda K :: \text{nat. primes-theta } (c * \text{real } K) / \text{real } K) \longrightarrow c)$ *at-top*
proof (*cases c = 0*)
case *True*
thus *?thesis* **by** (*simp add: primes-theta-def prime-sum-upto-def*)
next
case *False*
hence *c0*: $0 < c$ **using** *assms* **by** *simp*
have *lim*: $((\lambda K :: \text{nat. primes-theta } (c * \text{real } K) / (c * \text{real } K)) \longrightarrow 1)$ *at-top*
proof –
have *cK*: *filterlim* $(\lambda K :: \text{nat. } c * \text{real } K)$ *at-top at-top*
by (*rule filterlim-tendsto-pos-mult-at-top*[*OF* *tendsto-const c0 filterlim-real-sequentially*])
show *?thesis* **using** *filterlim-compose*[*OF* *pnt-theta cK*] **by** *simp*
qed
have $((\lambda K :: \text{nat. } c * (\text{primes-theta } (c * \text{real } K) / (c * \text{real } K))) \longrightarrow c * 1)$ *at-top*
by (*rule tendsto-mult-left*[*OF* *lim*])
thus *?thesis*
using *Lim-transform-eventually*[*OF* - *eventually-mono*[*OF* *eventually-gt-at-top*[*of* $0 :: \text{nat}$]
theta-ratio-eq[*OF* *c0*]]] **by** *auto*
qed

lemma *cell-tendsto*:
assumes $0 \leq c \leq d$
shows $((\lambda K :: \text{nat. } (\text{primes-theta } (d * \text{real } K) - \text{primes-theta } (c * \text{real } K)) / \text{real } K) \longrightarrow d - c)$ *at-top*
using *tendsto-diff*[*OF* *theta-tendsto*[*OF* *assms*(2)] *theta-tendsto*[*OF* *assms*(1)]]
by (*simp add: diff-divide-distrib*)

Eventually $\vartheta(dK) - \vartheta(cK) \leq (d - c + \delta) K$ resp. $\geq (d - c - \delta) K$.

lemma *cell-upper*:

assumes $0 \leq c \ 0 \leq d \ 0 < \delta$

shows $\forall_F K :: \text{nat in at-top. primes-theta } (d * \text{real } K) - \text{primes-theta } (c * \text{real } K) \leq (d - c + \delta) * \text{real } K$

proof –

have $\forall_F K :: \text{nat in at-top. } (\text{primes-theta } (d * \text{real } K) - \text{primes-theta } (c * \text{real } K)) / \text{real } K < d - c + \delta$

using *order-tendstoD(2)[OF cell-tendsto[OF assms(1,2)], of d - c + δ] assms(3)*

by *simp*

moreover have $\forall_F K :: \text{nat in at-top. } 0 < K$ **by** (*rule eventually-gt-at-top*)

ultimately show *?thesis*

by *eventually-elim (simp add: divide-less-eq)*

qed

lemma *cell-lower*:

assumes $0 \leq c \ 0 \leq d \ 0 < \delta$

shows $\forall_F K :: \text{nat in at-top. } (d - c - \delta) * \text{real } K \leq \text{primes-theta } (d * \text{real } K) - \text{primes-theta } (c * \text{real } K)$

proof –

have $\forall_F K :: \text{nat in at-top. } d - c - \delta < (\text{primes-theta } (d * \text{real } K) - \text{primes-theta } (c * \text{real } K)) / \text{real } K$

using *order-tendstoD(1)[OF cell-tendsto[OF assms(1,2)], of d - c - δ] assms(3)*

by *simp*

moreover have $\forall_F K :: \text{nat in at-top. } 0 < K$ **by** (*rule eventually-gt-at-top*)

ultimately show *?thesis*

by *eventually-elim (simp add: less-divide-eq)*

qed

6.2 Small primes: $\psi - \vartheta$ and the valuation of 24

$\psi(n) = \sum -\{p \leq n\} \lfloor \log p \rfloor \log p$ ($\text{nat-log } p \ n = \text{floorlog } p \ n - 1 = \lfloor \log p \rfloor$).

lemma *card-powers-le*: $\text{prime } p \implies 0 < n \implies \text{card } \{i. 0 < i \wedge p \wedge i \leq n\} = \text{nat-log } p \ n$

proof –

assume *p: prime p and n: 0 < n*

have *p1: 1 < p* **using** *p* **by** (*rule prime-gt-1-nat*)

have $\{i. 0 < i \wedge p \wedge i \leq n\} = \{1..\text{floorlog } p \ n - 1\}$

proof (*intro equalityI subsetI*)

fix *i* **assume** $i \in \{i. 0 < i \wedge p \wedge i \leq n\}$

hence $i: 0 < i \wedge p \wedge i \leq n$ **by** *auto*

have $\text{Suc } i \leq \text{floorlog } p \ n$ **by** (*rule floorlog-ge-SucI[OF i(2) p1]*)

thus $i \in \{1..\text{floorlog } p \ n - 1\}$ **using** *i(1)* **by** *auto*

next

fix *i* **assume** $i \in \{1..\text{floorlog } p \ n - 1\}$

hence $i: 1 \leq i \wedge i + 1 \leq \text{floorlog } p \ n$ **by** *auto*

have $p \wedge (i + 1 - 1) \leq n$ **by** (*rule floorlog-geD[OF i(2)]*) *simp*


```

    thus  $i \in \{i. 0 < i \wedge p \wedge i \leq n\}$  using  $i(1)$  by simp
  qed
  thus ?thesis by (simp add: nat-log-def)
qed

lemma psi-eq-sum:
   $0 < n \implies \text{primes-psi}(\text{real } n) = (\sum p \in \{p. \text{prime } p \wedge p \leq n\}. \text{real}(\text{nat-log } p \ n) * \ln(\text{real } p))$ 
proof -
  assume  $n: 0 < n$ 
  define  $P$  where  $P = \{p. \text{prime } p \wedge p \leq n\}$ 
  define  $I$  where  $I \ p = \{i. 0 < i \wedge p \wedge i \leq n\}$  for  $p$ 
  have  $fP$ : finite  $P$  unfolding  $P$ -def by simp
  have  $fI$ : finite  $(I \ p)$  if  $p: p \in P$  for  $p$ 
  proof (rule finite-subset[of -  $\{..n\}$ ])
    have  $p2$ :  $2 \leq p$  using  $p$  by (simp add:  $P$ -def prime-ge-2-nat)
    show  $I \ p \subseteq \{..n\}$ 
  proof
    fix  $i$  assume  $i \in I \ p$ 
    hence  $i$ :  $p \wedge i \leq n$  by (simp add:  $I$ -def)
    have  $i < 2 \wedge i$  by simp
    also have  $2 \wedge i \leq p \wedge i$  using  $p2$  by (rule power-mono) simp
    finally show  $i \in \{..n\}$  using  $i$  by simp
  qed
qed
simp
have  $\text{primes-psi}(\text{real } n) = (\sum (p, i) \mid \text{prime } p \wedge i > 0 \wedge \text{real}(p \wedge i) \leq \text{real } n. \ln(\text{real } p))$ 
  unfolding  $\text{primes-psi-def}$  by (rule sum-upto-primepows) (simp-all add: man-goldt-non-primepow)
also have  $\{(p, i). \text{prime } p \wedge i > 0 \wedge \text{real}(p \wedge i) \leq \text{real } n\} = \text{Sigma } P \ I$ 
proof (intro equalityI subsetI)
  fix  $x$  assume  $x \in \{(p, i). \text{prime } p \wedge i > 0 \wedge \text{real}(p \wedge i) \leq \text{real } n\}$ 
  then obtain  $p \ i$  where  $x: x = (p, i)$   $\text{prime } p \ 0 < i \ p \wedge i \leq n$  by auto
  have  $p \leq p \wedge i$  using  $x(2,3)$   $\text{prime-ge-1-nat}[OF \ x(2)]$  by (intro self-le-power)
simp-all
  thus  $x \in \text{Sigma } P \ I$  using  $x$  by (simp add:  $P$ -def  $I$ -def)
next
  fix  $x$  assume  $x \in \text{Sigma } P \ I$ 
  thus  $x \in \{(p, i). \text{prime } p \wedge i > 0 \wedge \text{real}(p \wedge i) \leq \text{real } n\}$ 
    by (force simp:  $P$ -def  $I$ -def)
qed
also have  $(\sum (p, i) \in \text{Sigma } P \ I. \ln(\text{real } p)) = (\sum p \in P. \sum i \in I \ p. \ln(\text{real } p))$ 
  by (rule sum.Sigma[symmetric, OF  $fP$ ]) (use  $fI$  in blast)
also have  $\dots = (\sum p \in P. \text{real}(\text{nat-log } p \ n) * \ln(\text{real } p))$ 
proof (rule sum.cong[OF refl])
  fix  $p$  assume  $p: p \in P$ 
  have  $\text{card}(I \ p) = \text{nat-log } p \ n$  unfolding  $I$ -def using  $p \ n$ 
    by (intro card-powers-le) (simp-all add:  $P$ -def)

```

thus $(\sum i \in I p. \ln (\text{real } p)) = \text{real } (\text{nat-log } p \ n) * \ln (\text{real } p)$ **by** *simp*
qed
 finally **show** *?thesis* **unfolding** *P-def* .
qed

$$\sum -\{p \leq N\} (\lfloor \log\text{-}p \ N \rfloor - 1) \log p = \psi(N) - \vartheta(N) \leq 2 \log N \sqrt{N}.$$

lemma *psi-minus-theta-sum*:

assumes $2 \leq N$
shows $(\sum p \in \{p. \text{prime } p \wedge p \leq N\}. (\text{real } (\text{nat-log } p \ N) - 1) * \ln (\text{real } p)) \leq 2 * \ln (\text{real } N) * \text{sqrt } (\text{real } N)$
proof –
have $(\sum p \in \{p. \text{prime } p \wedge p \leq N\}. (\text{real } (\text{nat-log } p \ N) - 1) * \ln (\text{real } p)) =$
 $\text{primes-psi } (\text{real } N) - \text{primes-theta } (\text{real } N)$
unfolding *psi-eq-sum*[of *N*] *theta-eq-sum*[of *real N*] **using** *assms*
by (*simp add: sum-subtractf left-diff-distrib psi-eq-sum theta-eq-sum*)
also have $\dots \leq 2 * \ln (\text{real } N) * \text{sqrt } (\text{real } N)$ **using** *assms*
by (*intro psi-minus-θ-bound simp*)
finally show *?thesis* .
qed

lemma *eventually-psi-theta-small*:

assumes $0 < \delta$
shows $\forall_F K :: \text{nat in at-top. } 2 * \ln (\text{real } (5 * K)) * \text{sqrt } (\text{real } (5 * K)) \leq \delta * \text{real } K$
proof –
have $((\lambda K :: \text{nat. } 2 * \ln (5 * \text{real } K) * \text{sqrt } (5 * \text{real } K) / \text{real } K) \longrightarrow 0) \text{ at-top}$
by *real-asymp*
hence $\forall_F K :: \text{nat in at-top. } 2 * \ln (5 * \text{real } K) * \text{sqrt } (5 * \text{real } K) / \text{real } K < \delta$
using *assms* **by** (*rule order-tendstoD*)
moreover have $\forall_F K :: \text{nat in at-top. } 0 < K$ **by** (*rule eventually-gt-at-top*)
ultimately show *?thesis* **by** *eventually-elim (simp add: divide-less-eq)*
qed

lemma *mult24-log-le*:

$\text{prime } p \implies \text{real } (\text{multiplicity } p \ (24::\text{nat})) * \ln (\text{real } p) \leq (\text{if } p \leq 24 \text{ then } \ln 24 \text{ else } 0)$
proof –
assume *p: prime p*
have *pos: 0 < p* **using** *p* **by** (*rule prime-gt-0-nat*)
show *?thesis*
proof (*cases p ≤ 24*)
case *True*
have $p \wedge \text{multiplicity } p \ 24 \text{ dvd } (24::\text{nat})$ **by** (*rule multiplicity-dvd*)
hence $p \wedge \text{multiplicity } p \ 24 \leq (24::\text{nat})$ **by** (*rule dvd-imp-le*) *simp*
hence $\text{le}': \text{real } (p \wedge \text{multiplicity } p \ 24) \leq 24$ **by** *linarith*

```

have pp: 0 < real (p ^ multiplicity p 24) using pos by simp
have real (multiplicity p (24::nat)) * ln (real p) = ln (real (p ^ multiplicity p
24))
  using pos by (simp add: ln-realpow)
also have ... ≤ ln 24 using le' pp by (simp add: ln-le-cancel-iff)
finally show ?thesis using True by simp
next
case False
have ¬ p dvd (24::nat) using False by (fastforce dest: dvd-imp-le)
thus ?thesis using False by (simp add: not-dvd-imp-multiplicity-0)
qed
qed

```

6.3 Step majorants on a uniform partition

Grid points of the uniform partition of $[a, b]$ into n cells.

definition $cx :: real \Rightarrow real \Rightarrow nat \Rightarrow nat \Rightarrow real$
where
 $cx\ a\ b\ n\ i = a + (b - a) * real\ i / real\ n$

The supremum of φ over the closed i -th cell.

definition $cM :: (real \Rightarrow real) \Rightarrow real \Rightarrow real \Rightarrow nat$
 $\Rightarrow nat \Rightarrow real$ **where**
 $cM\ \varphi\ a\ b\ n\ i = Sup\ (\varphi\ ' (\{cx\ a\ b\ n\ i..cx\ a\ b\ n\ (i + 1)\} \cap \{a..b\}))$

The step majorant of $\varphi + B$.

definition $cG :: (real \Rightarrow real) \Rightarrow real \Rightarrow real \Rightarrow real$
 $\Rightarrow nat \Rightarrow real \Rightarrow real$ **where**
 $cG\ \varphi\ a\ b\ B\ n\ y = (\sum\ i < n. indicator\ \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\ y * (cM\ \varphi\ a\ b\ n\ i + B))$

lemma $cx\ mono: a \leq b \implies i \leq j \implies cx\ a\ b\ n\ i \leq cx\ a\ b\ n\ j$
unfolding $cx\ def$ **by** (intro add-left-mono divide-right-mono mult-left-mono)
 $simp\ all$

lemma $cx\ zero: cx\ a\ b\ n\ 0 = a$ **by** (simp add: $cx\ def$)
lemma $cx\ self: 0 < n \implies cx\ a\ b\ n\ n = b$ **by** (simp add: $cx\ def$)
lemma $cx\ succ\ sub: 0 < n \implies cx\ a\ b\ n\ (i + 1) - cx\ a\ b\ n\ i = (b - a) / real\ n$
by (simp add: $cx\ def$ field-simps)
lemma $cx\ ge: a \leq b \implies a \leq cx\ a\ b\ n\ i$ **using** $cx\ mono[of\ a\ b\ 0\ i\ n]$
by (simp add: $cx\ zero$)
lemma $cx\ le: a \leq b \implies 0 < n \implies i \leq n \implies$
 $cx\ a\ b\ n\ i \leq b$ **using** $cx\ mono[of\ a\ b\ i\ n\ n]$ **by** (simp add: $cx\ self$)

lemma $cell\ unique:$
assumes $a \leq b\ y \in \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\ y \in$
 $\{cx\ a\ b\ n\ j <..cx\ a\ b\ n\ (j + 1)\}$
shows $i = j$

```

proof (rule ccontr)
  assume ne:  $i \neq j$ 
  show False
  proof (cases  $i < j$ )
    case True
      have  $cx\ a\ b\ n\ (i + 1) \leq cx\ a\ b\ n\ j$  using True by (intro cx-mono[OF assms(1)])
    simp
      thus False using assms(2,3) by auto
    next
      case False
      hence  $j < i$  using ne by simp
      hence  $cx\ a\ b\ n\ (j + 1) \leq cx\ a\ b\ n\ i$  by (intro cx-mono[OF assms(1)]) simp
      thus False using assms(2,3) by force
  qed
qed

```

lemma cell-exists:

```

  assumes  $a < b\ 0 < n\ y \in \{a..b\}$ 
  shows  $\exists i < n.\ y \in \{cx\ a\ b\ n\ i..cx\ a\ b\ n\ (i + 1)\}$ 
proof -
  have  $w: 0 < b - a$  using assms(1) by simp
  have  $nr: 0 < real\ n$  using assms(2) by simp
  have  $y: a < y\ y \leq b$  using assms(3) by auto
  define  $t$  where  $t = (y - a) * real\ n / (b - a)$ 
  have  $t0: 0 < t$  unfolding t-def using y w nr by simp
  have  $tn: t \leq real\ n$  unfolding t-def using y w nr by (simp add: field-simps)
  define  $k$  where  $k = nat\ \lceil t \rceil$ 
  have  $k1: 1 \leq k$  unfolding k-def using t0 by linarith
  have  $kn: k \leq n$  unfolding k-def using tn by simp
  have  $kt1: real\ k - 1 < t$  unfolding k-def using t0 by linarith
  have  $kt2: t \leq real\ k$  unfolding k-def using t0 by simp
  have  $e: real\ (k - 1) = real\ k - 1$  using k1 by simp
  have  $e2: k - 1 + 1 = k$  using k1 by simp
  have  $lo: cx\ a\ b\ n\ (k - 1) < y$  unfolding cx-def e using kt1 w nr by (simp add:
    t-def field-simps)
  have  $hi: y \leq cx\ a\ b\ n\ (k - 1 + 1)$  unfolding cx-def e2 using kt2 w nr
    by (simp add: t-def field-simps)
  have  $k - 1 < n$  using k1 kn by linarith
  thus ?thesis using lo hi by auto
qed

```

lemma cM-bdd:

```

  fixes  $\varphi :: real \Rightarrow real$ 
  assumes  $\bigwedge y. y \in \{a..b\} \implies |\varphi\ y| \leq B$ 
  shows bdd-above ( $\varphi\ '(\{cx\ a\ b\ n\ i..cx\ a\ b\ n\ (i + 1)\} \cap \{a..b\})$ )
proof (rule bdd-aboveI[of - B])
  fix  $z$  assume  $z \in \varphi\ '(\{cx\ a\ b\ n\ i..cx\ a\ b\ n\ (i + 1)\} \cap \{a..b\})$ 
  then obtain  $y$  where  $y: y \in \{a..b\}\ z = \varphi\ y$  by auto
  show  $z \leq B$  using assms[OF y(1)] y(2) abs-ge-self[of  $\varphi\ y$ ] by linarith

```

qed

lemma *cM-le*:

assumes $\bigwedge y. y \in \{a..b\} \implies$
 $|\varphi y| \leq B \ 0 \leq B \ a \leq b \ 0 < n \ i < n$
 shows $cM \ \varphi \ a \ b \ n \ i \leq B$

proof –

have *mem*: $cx \ a \ b \ n \ i \in \{cx \ a \ b \ n \ i..cx \ a \ b \ n \ (i + 1)\} \cap \{a..b\}$
 using *cx-mono*[*OF assms*(3), *of i i + 1 n*] *cx-ge*[*OF assms*(3), *of n i*] *cx-le*[*OF assms*(3,4),
of i] *assms*(5) **by** *auto*
 have *Sup* ($\varphi \ ' (\{cx \ a \ b \ n \ i..cx \ a \ b \ n \ (i + 1)\} \cap \{a..b\})$) $\leq B$
 proof (rule *cSup-least*)
 show $\varphi \ ' (\{cx \ a \ b \ n \ i..cx \ a \ b \ n \ (i + 1)\} \cap \{a..b\}) \neq \{\}$ **using** *mem* **by** *blast*
 fix *x* **assume** $x \in \varphi \ ' (\{cx \ a \ b \ n \ i..cx \ a \ b \ n \ (i + 1)\} \cap \{a..b\})$
 then **obtain** *y* **where** $y \in \{a..b\} \ x = \varphi \ y$ **by** *auto*
 show $x \leq B$ **using** *assms*(1)[*OF y*(1)] *y*(2) **by** *linarith*
 qed
 thus ?thesis **unfolding** *cM-def* .
 qed

qed

lemma *cM-ge*:

assumes $\bigwedge y. y \in \{a..b\} \implies$
 $|\varphi y| \leq B \ a \leq b \ 0 < n \ i < n$
 shows $-B \leq cM \ \varphi \ a \ b \ n \ i$

proof –

have *mem*: $cx \ a \ b \ n \ i \in \{cx \ a \ b \ n \ i..cx \ a \ b \ n \ (i + 1)\} \cap \{a..b\}$
 using *cx-mono*[*OF assms*(2), *of i i + 1 n*] *cx-ge*[*OF assms*(2), *of n i*] *cx-le*[*OF assms*(2,3),
of i] *assms*(4) **by** *auto*
 have $\varphi \ (cx \ a \ b \ n \ i) \leq cM \ \varphi \ a \ b \ n \ i$ **unfolding** *cM-def*
by (rule *cSup-upper*[*OF imageI*[*OF mem*] *cM-bdd*[*OF assms*(1)]])
 moreover have $-B \leq \varphi \ (cx \ a \ b \ n \ i)$ **using** *assms*(1)[*of cx a b n i*] *mem* **by**
fastforce
 ultimately show ?thesis **by** *linarith*
 qed

lemma *le-cM*:

assumes $\bigwedge y. y \in \{a..b\} \implies$
 $|\varphi y| \leq B \ y \in \{cx \ a \ b \ n \ i..cx \ a \ b \ n \ (i + 1)\} \cap \{a..b\}$
 shows $\varphi \ y \leq cM \ \varphi \ a \ b \ n \ i$
 unfolding *cM-def* **by** (rule *cSup-upper*[*OF - cM-bdd*[*OF assms*(1)]]) (use *assms*(2))
in *auto*

lemma *cG-eval*:

assumes $a < b \ 0 < n \ y \in \{a<..b\}$
 shows $\exists i < n. y \in \{cx \ a \ b \ n \ i<..cx \ a \ b \ n \ (i + 1)\} \wedge$
 $cG \ \varphi \ a \ b \ n \ y = cM \ \varphi \ a \ b \ n \ i + B$

proof –

obtain i **where** $i: i < n \ y \in \{cx \ a \ b \ n \ i < .. cx \ a \ b \ n \ (i + 1)\}$ **using** $cell\text{-}exists[OF \ assms]$
by $blast$
have $cG \ \varphi \ a \ b \ B \ n \ y = cM \ \varphi \ a \ b \ n \ i + B$
unfolding $cG\text{-}def$
proof –
have $z: indicator \ \{cx \ a \ b \ n \ j < .. cx \ a \ b \ n \ (j + 1)\} \ y * (cM \ \varphi \ a \ b \ n \ j + B) = 0$
if $j: j \in \{.. < n\} - \{i\}$ **for** j
proof –
have $y \notin \{cx \ a \ b \ n \ j < .. cx \ a \ b \ n \ (j + 1)\}$
proof
assume $y \in \{cx \ a \ b \ n \ j < .. cx \ a \ b \ n \ (j + 1)\}$
hence $j = i$ **by** $(rule \ cell\text{-}unique[OF \ less\text{-}imp\text{-}le[OF \ assms(1)] - i(2)])$
thus $False$ **using** j **by** $simp$
qed
thus $?thesis$ **by** $simp$
qed
have $(\sum \ j < n. indicator \ \{cx \ a \ b \ n \ j < .. cx \ a \ b \ n \ (j + 1)\} \ y * (cM \ \varphi \ a \ b \ n \ j + B)) =$
 $indicator \ \{cx \ a \ b \ n \ i < .. cx \ a \ b \ n \ (i + 1)\} \ y * (cM \ \varphi \ a \ b \ n \ i + B) + (\sum \ j \in$
 $\{.. < n\} - \{i\}. indicator \ \{cx \ a \ b \ n \ j < .. cx \ a \ b \ n \ (j + 1)\} \ y * (cM \ \varphi \ a \ b \ n \ j$
 $+ B))$
by $(rule \ sum.remove) \ (use \ i \ in \ auto)$
also have $(\sum \ j \in \{.. < n\} - \{i\}. indicator \ \{cx \ a \ b \ n \ j < .. cx \ a \ b \ n \ (j + 1)\} \ y * (cM \ \varphi \ a \ b \ n \ j + B)) = 0$
by $(rule \ sum.neutral) \ (use \ z \ in \ blast)$
also have $indicator \ \{cx \ a \ b \ n \ i < .. cx \ a \ b \ n \ (i + 1)\} \ y * (cM \ \varphi \ a \ b \ n \ i + B) =$
 $cM \ \varphi \ a \ b \ n \ i + B$ **using** $i(2)$ **by** $simp$
finally show $(\sum \ j < n. indicator \ \{cx \ a \ b \ n \ j < .. cx \ a \ b \ n \ (j + 1)\} \ y * (cM \ \varphi \ a \ b \ n \ j +$
 $B)) = cM \ \varphi \ a \ b \ n \ i + B$ **by** $simp$
qed
thus $?thesis$ **using** i **by** $blast$
qed

lemma $cG\text{-}bounds$:

assumes $a < b \ \bigwedge y. y \in \{a..b\} \implies$
 $|\varphi \ y| \leq B \ 0 \leq B \ 0 < n \ y \in \{a < .. b\}$
shows $\varphi \ y + B \leq cG \ \varphi \ a \ b \ B \ n \ y \wedge 0 \leq cG \ \varphi \ a \ b \ B \ n \ y \wedge$
 $cG \ \varphi \ a \ b \ B \ n \ y \leq 2 * B$

proof –

obtain i **where** $i: i < n \ y \in \{cx \ a \ b \ n \ i < .. cx \ a \ b \ n \ (i + 1)\}$ **and** $e: cG \ \varphi \ a \ b$
 $B \ n \ y =$
 $cM \ \varphi \ a \ b \ n \ i + B$
using $cG\text{-}eval[OF \ assms(1,4,5)]$ **by** $blast$
have $\varphi \ y \leq cM \ \varphi \ a \ b \ n \ i$ **using** $i \ assms(5)$ **by** $(intro \ le\text{-}cM[OF \ assms(2)]) \ auto$
moreover have $cM \ \varphi \ a \ b \ n \ i \leq B$
by $(rule \ cM\text{-}le[OF \ assms(2,3)] \ less\text{-}imp\text{-}le[OF \ assms(1)] \ assms(4) \ i(1))$
moreover have $- B \leq cM \ \varphi \ a \ b \ n \ i$

by (rule cM-ge[OF assms(2) less-imp-le[OF assms(1)] assms(4) i(1)])
ultimately show ?thesis unfolding e by linarith
qed

lemma cG-measurable [measurable]: $cG \varphi a b B n \in \text{borel-measurable borel}$
unfolding cG-def by measurable

At a continuity point of φ inside (a, b) , the majorants converge to $\varphi y + B$.

lemma cG-tendsto:

assumes $a < b \wedge y. y \in \{a..b\} \implies$
 $|\varphi y| \leq B \wedge y \in \{a < .. < b\} \text{ isCont } \varphi y$
shows $((\lambda n. cG \varphi a b B n y) \longrightarrow \varphi y + B)$ at-top
proof (rule tendstoI)
fix $\eta :: \text{real}$ assume $\eta: 0 < \eta$
obtain δ where $\delta: 0 < \delta \wedge u. \text{dist } u y < \delta \implies$
 $\text{dist } (\varphi u) (\varphi y) < \eta / 2$
proof –
have $h: \forall r > 0. \exists s > 0. \forall x. x \neq y \wedge$
 $\text{dist } x y < s \implies \text{dist } (\varphi x) (\varphi y) < r$
using assms(4) unfolding isCont-def LIM-eq dist-norm .
have $\exists s > 0. \forall x. x \neq y \wedge \text{dist } x y < s \implies$
 $\text{dist } (\varphi x) (\varphi y) < \eta / 2$
by (rule h[rule-format]) (use η in simp)
then obtain s where $s: 0 < s \wedge x. x \neq y \implies$
 $\text{dist } x y < s \implies \text{dist } (\varphi x) (\varphi y) < \eta / 2$ by blast
have $\text{dist } (\varphi u) (\varphi y) < \eta / 2$ if $\text{dist } u y < s$ for u
proof (cases $u = y$)
case True thus ?thesis using η by simp
next
case False thus ?thesis using $s(2)$ that by blast
qed
thus ?thesis using $s(1)$ that by blast
qed
obtain $N :: \text{nat}$ where $N: (b - a) / \delta < \text{real } N$ using reals-Archimedean2 by blast
show $\forall_F n$ in at-top. $\text{dist } (cG \varphi a b B n y) (\varphi y + B) < \eta$
proof (rule eventually-at-top-linorderI[of $N + 1$])
fix $n :: \text{nat}$ assume $n: N + 1 \leq n$
have $n0: 0 < n$ using n by simp
obtain i where $i: i < n \wedge y \in \{cx a b n i < .. cx a b n (i + 1)\}$ and $e: cG \varphi a b B n y = cM \varphi a b n i + B$
using cG-eval[OF assms(1) $n0$, of $y \varphi B$] assms(3) by auto
have low: $\varphi y \leq cM \varphi a b n i$ using i assms(3) by (intro le-cM[OF assms(2)])
auto
have width: $cx a b n (i + 1) - cx a b n i < \delta$
proof –
have $b - a < \text{real } N * \delta$ using $N \delta(1)$ by (simp add: divide-less-eq)
moreover have $\text{real } N * \delta \leq \text{real } n * \delta$ using $n \delta(1)$ by simp
ultimately have $b - a < \text{real } n * \delta$ by linarith

```

    thus ?thesis unfolding cx-succ-sub[OF n0] using n0
    by (simp add: pos-divide-less-eq mult.commute)
qed
have up: cM  $\varphi$  a b n i  $\leq \varphi$  y +  $\eta$  / 2
    unfolding cM-def
proof (rule cSup-least)
  show  $\varphi$  ' ( $\{cx\ a\ b\ n\ i..cx\ a\ b\ n\ (i + 1)\} \cap \{a..b\}$ )  $\neq \{\}$ 
    using i(2) assms(3) by force
  fix z assume z  $\in \varphi$  ' ( $\{cx\ a\ b\ n\ i..cx\ a\ b\ n\ (i + 1)\} \cap \{a..b\}$ )
  then obtain u where u: u  $\in \{cx\ a\ b\ n\ i..cx\ a\ b\ n\ (i + 1)\} \cap \{a..b\}$  z =
     $\varphi\ u$  by auto
  have dist u y <  $\delta$  using u(1) i(2) width by (fastforce simp: dist-real-def)
  hence dist ( $\varphi\ u$ ) ( $\varphi\ y$ ) <  $\eta$  / 2 by (rule  $\delta$ (2))
  thus z  $\leq \varphi\ y + \eta$  / 2 using u(2) abs-ge-self[of  $\varphi\ u - \varphi\ y$ ]
    unfolding dist-real-def by linarith
qed
show dist (cG  $\varphi$  a b B n y) ( $\varphi\ y + B$ ) <  $\eta$  unfolding e dist-real-def
    using low up  $\eta$  by simp
qed
qed

lemma set-integral-sum-nat:
  fixes f :: nat  $\Rightarrow$  real  $\Rightarrow$  real
  assumes  $\bigwedge i. i \in I \implies \text{set-integrable lborel } A\ (f\ i)$ 
  shows set-integrable lborel A ( $\lambda x. \sum i \in I. f\ i\ x$ )  $\wedge$ 
    set-lebesgue-integral lborel A ( $\lambda x. \sum i \in I. f\ i\ x$ ) = ( $\sum i \in$ 
      I. set-lebesgue-integral lborel A (f i))
proof -
  have i:  $\bigwedge i. i \in I \implies$ 
    integrable lborel ( $\lambda x. \text{indicator } A\ x *_R f\ i\ x$ ) using assms
    unfolding set-integrable-def .
  have e: ( $\lambda x. \text{indicator } A\ x *_R (\sum i \in I. f\ i\ x)$ ) =
    ( $\lambda x. \sum i \in I. \text{indicator } A\ x *_R f\ i\ x$ )
    by (simp add: sum-distrib-left)
  show ?thesis unfolding set-integrable-def set-lebesgue-integral-def e
    using Bochner-Integration.integrable-sum[OF i] Bochner-Integration.integral-sum[OF
i]
    by (rule conjI)
qed

lemma integral-cG:
  assumes a < b 0 < n
  shows interval-lebesgue-integral lborel (ereal a) (ereal b) (cG  $\varphi$  a b B n) = ( $\sum$ 
i <
  n. (cM  $\varphi$  a b n i + B) * ((b - a) / real n))
proof -
  have ab: a  $\leq b$  using assms(1) by simp
  define C where C i = {cx a b n i <..cx a b n (i + 1)} for i
  define c where c i = cM  $\varphi$  a b n i + B for i

```


have $cx3: a \leq cx\ a\ b\ n\ i \wedge cx\ a\ b\ n\ (i + 1) \leq b \wedge$
 $cx\ a\ b\ n\ i \leq cx\ a\ b\ n\ (i + 1)$ **if** $i < n$ **for** i
using $cx\text{-}ge[OF\ ab, of\ n\ i]$ $cx\text{-}le[OF\ ab\ assms(2), of\ i + 1]$ $cx\text{-}mono[OF\ ab,$
 $of\ i\ i + 1\ n]$ **that by auto**
have $nb: \{b\} \in \text{null-sets } lborel$ **by** ($simp\ add: \text{finite-imp-null-set-lborel}$)
have $meas: \text{measure } lborel\ (C\ i \cap \{a <..<b\}) = (b - a) / \text{real } n$ **if** $i: i < n$ **for** i
proof –
have $C\ i \cap \{a <..<b\} = C\ i - \{b\}$ **using** $cx3[OF\ i]$ **unfolding** $C\text{-def}$ **by**
fastforce
hence $\text{measure } lborel\ (C\ i \cap \{a <..<b\}) = \text{measure } lborel\ (C\ i - \{b\})$ **by** $simp$
also have $\dots = \text{measure } lborel\ (C\ i)$ **unfolding** $C\text{-def}$
by ($rule\ \text{measure-Diff-null-set}[OF - nb]$) $simp$
also have $\dots = cx\ a\ b\ n\ (i + 1) - cx\ a\ b\ n\ i$ **unfolding** $C\text{-def}$ **using** $cx3[OF$
 $i]$ **by** $simp$
also have $\dots = (b - a) / \text{real } n$ **by** ($rule\ cx\text{-}succ\text{-}sub[OF\ assms(2)]$)
finally show $?thesis$.
qed
have $one: \text{set-lebesgue-integral } lborel\ \{a <..<b\}\ (\lambda y. \text{indicator } (C\ i)\ y * c\ i) =$
 $c\ i * ((b - a) / \text{real } n)$ **if** $i: i < n$ **for** i
proof –
have $(\lambda y. \text{indicator } \{a <..<b\}\ y *_{\mathbb{R}} (\text{indicator } (C\ i)\ y * c\ i)) =$
 $(\lambda y. c\ i * \text{indicator } (C\ i \cap \{a <..<b\})\ y)$
by ($auto\ simp: \text{fun-eq-iff indicator-def}$)
hence $\text{set-lebesgue-integral } lborel\ \{a <..<b\}\ (\lambda y. \text{indicator } (C\ i)\ y * c\ i) =$
 $c\ i * \text{measure } lborel\ (C\ i \cap \{a <..<b\})$
unfolding $\text{set-lebesgue-integral-def}$ **by** $simp$
thus $?thesis$ **unfolding** $meas[OF\ i]$.
qed
have $ints: \text{set-integrable } lborel\ \{a <..<b\}\ (\lambda y. \text{indicator } (C\ i)\ y * c\ i)$
if $i: i \in \{..<n\}$ **for** i
proof –
have $(\lambda y. \text{indicator } \{a <..<b\}\ y *_{\mathbb{R}} (\text{indicator } (C\ i)\ y * c\ i)) =$
 $(\lambda y. c\ i * \text{indicator } (C\ i \cap \{a <..<b\})\ y)$
by ($auto\ simp: \text{fun-eq-iff indicator-def}$)
moreover have $\text{integrable } lborel\ (\text{indicator } (C\ i \cap \{a <..<b\}) :: \text{real} \Rightarrow \text{real})$
proof –
have $\text{emeasure } lborel\ (C\ i \cap \{a <..<b\}) \leq \text{emeasure } lborel\ \{a <..<b\}$
by ($rule\ \text{emeasure-mono}$) ($simp\text{-all } add: C\text{-def}$)
also have $\dots = \text{ennreal } (b - a)$ **using** ab **by** $simp$
finally have $\text{emeasure } lborel\ (C\ i \cap \{a <..<b\}) < \infty$
using $\text{ennreal-less-top}[of\ b - a]$ **by** ($auto\ intro: \text{order.strict-trans1}$)
thus $?thesis$ **by** ($simp\ add: \text{integrable-indicator-iff } C\text{-def}$)
qed
ultimately show $?thesis$ **unfolding** $\text{set-integrable-def}$ **by** $simp$
qed
have $\text{interval-lebesgue-integral } lborel\ (\text{ereal } a)\ (\text{ereal } b)\ (cG\ \varphi\ a\ b\ B\ n) =$
 $\text{set-lebesgue-integral } lborel\ \{a <..<b\}\ (\lambda y. \sum\ i < n. \text{indicator } (C\ i)\ y * c\ i)$
using ab **unfolding** $cG\text{-def } C\text{-def } c\text{-def}$ **by** ($simp\ add: \text{interval-lebesgue-integral-le-eq}$)
also have $\dots = (\sum\ i < n. \text{set-lebesgue-integral } lborel\ \{a <..<b\}\ (\lambda y. \text{indicator}$

```

    (C i) y * c i))
  by (rule conjunct2[OF set-integral-sum-nat[OF ints]])
also have ... = (∑ i < n. c i * ((b - a) / real n))
  by (rule sum.cong[OF refl]) (rule one, simp)
finally show ?thesis unfolding c-def .
qed

```

6.4 Dominated convergence for the majorants

```

lemma ae-cG-tendsto:
  assumes a < b ∧ y. y ∈ {a..b} ⇒
    |φ y| ≤ B countable D
  and ∧ y. y ∈ {a<..

```

φ (restricted to (a, b)) is Borel measurable and integrable.

```

lemma phi-measurable:
  fixes φ :: real ⇒ real
  assumes countable D ∧ y. y ∈ {a<..

```

```

    hence eventually ( $\lambda y. y \in \{..<a\} \cup \{b<..\}$ ) (nhds x)
    by (intro eventually-nhds-in-open) auto
    hence e: eventually ( $\lambda y. \text{indicator } \{a<..b\} y * \varphi y = 0$ ) (nhds x)
    by eventually-elim auto
    have isCont ( $\lambda y. 0::\text{real}$ ) x by simp
    thus ?thesis using isCont-cong[OF e] by simp
  qed
qed
qed

```

lemma phi-integrable:

```

  fixes  $\varphi :: \text{real} \Rightarrow \text{real}$ 
  assumes  $a < b \wedge y. y \in \{a..b\} \implies$ 
     $|\varphi y| \leq B$  countable D
    and  $\wedge y. y \in \{a<..b\} \implies y \notin D \implies \text{isCont } \varphi y$ 
  shows interval-lebesgue-integrable lborel (ereal a) (ereal b)  $\varphi$ 
proof -
  have integrable lborel ( $\lambda y. \text{indicator } \{a<..b\} y *_R \varphi y$ )
proof (rule Bochner-Integration.integrable-bound)
  show integrable lborel ( $\lambda y. \text{indicator } \{a<..b\} y *_R B$ )
    using interval-integral-const(1)[of a b B] assms(1)
    unfolding interval-lebesgue-integrable-def set-integrable-def by force
  show ( $\lambda y. \text{indicator } \{a<..b\} y *_R \varphi y$ )  $\in$  borel-measurable lborel
    using phi-measurable[of D a b  $\varphi$ , OF assms(3,4)] by fastforce
  show AE y in lborel. norm (indicator  $\{a<..b\} y *_R \varphi y$ )  $\leq$ 
    norm (indicator  $\{a<..b\} y *_R B$ )
proof (rule AE-I2)
  fix y :: real
  show norm (indicator  $\{a<..b\} y *_R \varphi y$ )  $\leq$ 
    norm (indicator  $\{a<..b\} y *_R B$ )
    using assms(2)[of y] by (force simp: indicator-def)
qed
qed
thus ?thesis unfolding interval-lebesgue-integrable-def set-integrable-def using
assms(1) by force
qed

```

lemma integral-cG-tendsto:

```

  assumes  $a < b \wedge y. y \in \{a..b\} \implies$ 
     $|\varphi y| \leq B$   $0 \leq B$  countable D
    and  $\wedge y. y \in \{a<..b\} \implies y \notin D \implies \text{isCont } \varphi y$ 
  shows (( $\lambda n. \text{interval-lebesgue-integral lborel (ereal a) (ereal b) (cG } \varphi a b B n)$ )
     $\longrightarrow \text{interval-lebesgue-integral lborel (ereal a) (ereal b) } (\lambda y. \varphi y + B)$ ) at-top
proof -
  have ab:  $\text{ereal } a \leq \text{ereal } b$  using assms(1) by simp
  define s where  $s n y = \text{indicator } \{a<..b\} y *_R cG \varphi a b B n$  for n y
  define g where  $g y = \text{indicator } \{a<..b\} y *_R (\varphi y + B)$  for y
  define w where  $w y = \text{indicator } \{a<..b\} y *_R (2 * B)$  for y

```

```

have gm:  $g \in \text{borel-measurable lborel}$ 
proof -
  have  $(\lambda y. \text{indicator } \{a <..< b\} y * \varphi y) \in \text{borel-measurable borel}$ 
    by (rule phi-measurable[of  $D$   $a$   $b$   $\varphi$ , OF assms(4,5)])
  moreover have  $(\lambda y. \text{indicator } \{a <..< b\} y * B :: \text{real}) \in \text{borel-measurable borel}$ 
    by measurable
  ultimately have  $(\lambda y. \text{indicator } \{a <..< b\} y * \varphi y + \text{indicator } \{a <..< b\} y * B) \in$ 
     $\text{borel-measurable borel}$  by measurable
  thus ?thesis unfolding g-def by (simp add: distrib-left)
qed
have sm:  $s \ n \in \text{borel-measurable lborel}$  for  $n$  unfolding s-def by measurable
have wi:  $\text{integrable lborel } w$ 
  using interval-integral-const(1)[of  $a$   $b$   $2 * B$ ] assms(1)
  unfolding interval-lebesgue-integrable-def set-integrable-def w-def by simp
have AE  $y$  in lborel.  $y \in \{a <..< b\} \longrightarrow$ 
   $((\lambda n. cG \ \varphi \ a \ b \ B \ n \ y) \longrightarrow \varphi \ y + B)$  at-top
  using assms(1,2,4,5) by (rule ae-cG-tendsto)
hence lim: AE  $y$  in lborel.  $(\lambda n. s \ n \ y) \longrightarrow g \ y$ 
proof (eventually-elim)
  case (elim  $y$ )
  show ?case
  proof (cases  $y \in \{a <..< b\}$ )
    case True thus ?thesis using elim by (simp add: s-def g-def)
  next
    case False thus ?thesis by (simp add: s-def g-def)
  qed
qed
have bound: AE  $y$  in lborel.  $\text{norm } (s \ n \ y) \leq w \ y$  for  $n$ 
proof (rule AE-I2)
  fix  $y :: \text{real}$ 
  show  $\text{norm } (s \ n \ y) \leq w \ y$ 
  proof (cases  $y \in \{a <..< b\} \wedge 0 < n$ )
    case True
    have  $0 \leq cG \ \varphi \ a \ b \ B \ n \ y \wedge cG \ \varphi \ a \ b \ B \ n \ y \leq 2 * B$ 
      using cG-bounds[of  $a$   $b$   $\varphi \ B \ n \ y$ , OF assms(1,2,3)] True by auto
    thus ?thesis using True by (simp add: s-def w-def)
  next
    case False
    thus ?thesis using assms(3) by (auto simp: s-def w-def cG-def)
  qed
qed
have  $(\lambda n. \text{integral}^L \text{ lborel } (s \ n)) \longrightarrow \text{integral}^L$ 
   $\text{lborel } g$ 
  by (rule integral-dominated-convergence[OF gm sm wi lim bound])
thus ?thesis unfolding interval-lebesgue-integral-le-eq[OF ab] einterval-eq-Icc
  set-lebesgue-integral-def s-def g-def .
qed

```

The change of variables $x = 1/y$: $\int_{-1/M}^{-1/3} y f(1/y) dy = \int_{-3}^{-M} f(x) dx$

$f(x)/x^3 \, dx$.

lemma *integral-inv-subst*:

fixes $f :: \text{real} \Rightarrow \text{real}$

assumes $3 \leq M \, (\lambda x. \text{indicator} \{3 <..<M\} \, x * f \, x) \in$

$\text{borel-measurable borel} \bigwedge x. x \in \{3..M\} \implies |f \, x| \leq B d$

shows $\text{interval-lebesgue-integral lborel} \, (\text{ereal} \, (1 / M)) \, (\text{ereal} \, (1 / 3)) \, (\lambda y. y * f$
 $(1 /$
 $y))$

$= \text{interval-lebesgue-integral lborel} \, (\text{ereal} \, 3) \, (\text{ereal} \, M) \, (\lambda x. f \, x / x^3)$

proof (*cases* $M = 3$)

case *True*

show *?thesis* **using** *True* **by** *simp*

next

case *False*

note $M = \text{assms}(1)$ **and** $hm = \text{assms}(2)$ **and** $Bnd = \text{assms}(3)$

have $M0: 0 < M$ **using** M **by** *simp*

have $M3: 3 < M$ **using** M *False* **by** *simp*

have $iM: 0 < 1 / M$ **using** $M0$ **by** *simp*

define h **where** $h \, x = \text{indicator} \{3 <..<M\} \, x * f \, x$ **for** $x :: \text{real}$

have $hm': h \in \text{borel-measurable borel}$ **using** hm **unfolding** $h\text{-def}[abs\text{-def}]$.

define F **where** $F \, u = - \, u * f \, (- \, 1 / u)$ **for** $u :: \text{real}$

define g **where** $g \, x = - \, 1 / x$ **for** $x :: \text{real}$

define g' **where** $g' \, x = 1 / x^2$ **for** $x :: \text{real}$

define c **where** $c = - \, (1 / 3 :: \text{real})$

define d **where** $d = - \, (1 / M)$

have $cd: c < d$ **unfolding** $c\text{-def} \, d\text{-def}$ **using** $M3$ **by** (*simp add: field-simps*)

have $gc: g \, 3 = c \, g \, M = d$ **unfolding** $g\text{-def} \, c\text{-def} \, d\text{-def}$ **by** *simp-all*

— reflection

have $r: \text{interval-lebesgue-integral lborel} \, (\text{ereal} \, (1 / M)) \, (\text{ereal} \, (1 / 3)) \, (\lambda y. y * f$
 $(1 /$
 $y))$

$= \text{interval-lebesgue-integral lborel} \, (\text{ereal} \, c) \, (\text{ereal} \, d) \, F$

using $\text{interval-integral-reflect}[of \, \text{ereal} \, (1 / M) \, \text{ereal} \, (1 / 3) \, \lambda y. y * f \, (1 / y)]$

unfolding $F\text{-def} \, c\text{-def} \, d\text{-def}$ **by** *simp*

— integrability of F on the closed interval

have $Fo: \text{set-integrable lborel} \, \{c <..<d\} \, F$

proof —

have $eq: (\lambda u. \text{indicator} \{c <..<d\} \, u *_R F \, u) =$

$(\lambda u. \text{indicator} \{c <..<d\} \, u * (- \, u * h \, (- \, 1 / u)))$

proof (*rule ext*)

fix $u :: \text{real}$

show $\text{indicator} \{c <..<d\} \, u *_R F \, u = \text{indicator} \{c <..<d\} \, u * (- \, u * h \, (- \, 1$
 $/ u))$

proof (*cases* $u \in \{c <..<d\}$)

case *True*

hence $ud: u < - \, (1 / M)$ **unfolding** $d\text{-def}$ **by** *simp*

hence $u0: u < 0$ **using** iM **by** *linarith*

have $3 < - \, 1 / u - 1 / u < M$ **using** *True* $u0 \, M0$ **unfolding** $c\text{-def} \, d\text{-def}$
by (*auto simp: field-simps*)

```

      thus ?thesis unfolding F-def h-def using True by simp
    qed simp
  qed
  have mm: (λu. indicator {c<.. $d$ } u * (− u * h (− 1 / u))) ∈
    borel-measurable lborel
    using hm' by simp
  show ?thesis unfolding set-integrable-def eq
  proof (rule Bochner-Integration.integrable-bound)
    show integrable lborel (λu. indicator {c<.. $d$ } u *R Bd)
      using interval-integral-const(1)[of c d Bd] cd
      unfolding interval-lebesgue-integrable-def set-integrable-def by simp
    show (λu. indicator {c<.. $d$ } u * (− u * h (− 1 / u))) ∈ borel-measurable
      lborel
      by (rule mm)
    show AE u in lborel. norm (indicator {c<.. $d$ } u * (− u * h (− 1 / u))) ≤
      norm (indicator {c<.. $d$ } u *R Bd)
    proof (rule AE-I2)
      fix u :: real
      show norm (indicator {c<.. $d$ } u * (− u * h (− 1 / u))) ≤
        norm (indicator {c<.. $d$ } u *R Bd)
      proof (cases u ∈ {c<.. $d$ })
        case True
          hence ud: u < − (1 / M) − (1 / 3) < u unfolding c-def d-def by
            simp-all
          hence u0: u < 0 − 1 ≤ u using iM by linarith+
          have i: 3 < − 1 / u − 1 / u < M using True u0 M0 unfolding c-def
            d-def
          by (auto simp: field-simps)
          have fb: |f (− 1 / u)| ≤ Bd using Bnd i by simp
          have |− u * h (− 1 / u)| = |u| * |f (− 1 / u)|
            unfolding h-def using i by (simp add: abs-mult)
          also have ... ≤ 1 * Bd by (rule mult-mono) (use u0 fb in auto)
          finally show ?thesis using True by fastforce
        case False
          qed simp
      qed
    qed
  qed
  have Fc: set-integrable lborel {g 3..g M} F
  proof −
    have {g 3..g M} = {c<.. $d$ } ∪ {c} ∪ {d} unfolding gc using cd by force
    moreover have set-integrable lborel ({c<.. $d$ } ∪ {c} ∪ {d}) F
      by (intro set-integrable-Un Fo set-borel-integrable-singleton) simp-all
    ultimately show ?thesis by simp
  qed
  have derivg: (g has-real-derivative g' x) (at x) if x ∈ {3..M} for x
  proof −
    have x ≠ 0 using that by simp
    thus ?thesis unfolding g-def g'-def
      by (fastforce simp: power2-eq-square intro: derivative-eq-intros)
  qed

```

```

qed
have contg': continuous-on {3..M} g' unfolding g'-def by (fastforce intro: continuous-intros)
have nn: 0 ≤ g' x if x ∈ {3..M} for x unfolding g'-def by simp
have interval-lebesgue-integral lborel (ereal (g 3)) (ereal (g M)) F =
  interval-lebesgue-integral lborel (ereal 3) (ereal M) (λx. F (g x) * g' x)
by (rule interval-integral-substitution(2)[OF Fc derivg contg' nn M])
moreover have interval-lebesgue-integral lborel (ereal 3) (ereal M) (λx. F (g x)
*
  g' x) = interval-lebesgue-integral lborel (ereal 3) (ereal M) (λx. f x / x ^ 3)
proof (rule interval-integral-cong)
  fix x assume x ∈ einterval (min (ereal 3) (ereal M)) (max (ereal 3) (ereal M))
  hence x: 3 < x using M by simp
  show F (g x) * g' x = f x / x ^ 3 unfolding F-def g-def g'-def using x
  by (simp add: field-simps power2-eq-square power3-eq-cube)
qed
ultimately show ?thesis unfolding r unfolding gc by simp
qed

```

6.5 The Riemann-sum bounds

lemma *split2-aux*: $c * ((x + d) * k) = c * x * k + d * k * (c :: \text{real})$
 by (simp add: algebra-simps)

Outer range, variable $y = p/K$: Riemann-sum upper bound for a bounded, almost-everywhere continuous φ (step majorants + the PNT cell estimates).

theorem *outer-prime-sum-bound*:

```

assumes a: 0 < a and ab: a ≤ b and Bd: ⋀y. y ∈ {a..b} ⇒
  |ϕ y| ≤ Bd and D: countable D
and cont: ⋀y. y ∈ {a<..b} ⇒ y ∉ D ⇒
  isCont ϕ y and e: 0 < ε
shows ∀F K in at-top.
  (∑ p ∈ {p. p ≤ nat (floor (b * real K)) ∧ prime p ∧
    a * real K < real p ∧ real p ≤ b * real K}.
    ϕ (real p / real K) * ln (real p))
  ≤ (interval-lebesgue-integral lborel (ereal a) (ereal b) ϕ + ε) * real K
proof (cases a = b)
case True
show ?thesis
proof (rule always-eventually, rule allI)
  fix K :: nat
  have e0: {p. p ≤ nat (floor (b * real K)) ∧ prime p ∧ a * real K < real p ∧
    real p ≤ b * real K} = {}
  using True by simp
  show (∑ p ∈ {p. p ≤ nat (floor (b * real K)) ∧ prime p ∧
    a * real K < real p ∧ real p ≤ b * real K}.
    ϕ (real p / real K) * ln (real p)) ≤
    (interval-lebesgue-integral lborel (ereal a) (ereal b) ϕ + ε) * real K
  unfolding e0 using True e by auto

```

```

qed
next
case False
hence  $ab'$ :  $a < b$  using  $ab$  by simp
have  $B0$ :  $0 \leq Bd$  using  $Bd[of\ a]$   $ab$  by auto
define  $I$  where  $I = \text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) \varphi$ 
have  $IB$ :  $\text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (\lambda y. \varphi\ y + Bd) =$ 
 $I + (b - a) * Bd$ 
unfolding  $I\text{-def}$ 
proof -
have  $\phi I$ :  $\text{interval-lebesgue-integrable lborel } (ereal\ a) (ereal\ b) \varphi$  using  $ab'$   $Bd$ 
 $D\ cont$ 
by (rule  $\phi I\text{-integrable}$ )
show  $\text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (\lambda y. \varphi\ y + Bd) =$ 
 $\text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) \varphi + (b - a) * Bd$ 
using  $\text{interval-lebesgue-integral-add}(2)[OF\ \phi I\ \text{interval-integral-const}(1)[of\ a$ 
 $b\ Bd]]$ 
 $\text{interval-integral-const}(2)[of\ a\ b\ Bd]$  by simp
qed
obtain  $n$  where  $n0$ :  $0 < n$  and  $n1$ :  $\text{interval-lebesgue-integral lborel } (ereal\ a)$ 
 $(ereal\ b) (cG$ 
 $\varphi\ a\ b\ Bd\ n) < I + (b - a) * Bd + \varepsilon / 2$ 
proof -
have  $((\lambda n. \text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (cG\ \varphi\ a\ b\ Bd\ n))$ 
 $\longrightarrow \text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (\lambda y.$ 
 $\varphi\ y + Bd))\ at\ top$ 
using  $ab'$   $Bd\ B0\ D\ cont$  by (rule  $\text{integral-cG-tendsto}$ )
hence  $\lim$ :  $((\lambda n. \text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (cG\ \varphi\ a\ b$ 
 $Bd\ n)) \longrightarrow I + (b - a) * Bd)\ at\ top$ 
unfolding  $IB$  .
have  $\forall_F n\ in\ at\ top. \text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (cG$ 
 $\varphi\ a\ b\ Bd\ n) < I + (b - a) * Bd + \varepsilon / 2$ 
using  $\text{order-tendstoD}(2)[OF\ \lim, of\ I + (b - a) * Bd + \varepsilon / 2]\ e$  by auto
moreover have  $\forall_F n :: nat\ in\ at\ top. 0 < n$  by (rule  $\text{eventually-gt-at-top}$ )
ultimately have  $\forall_F n\ in\ at\ top. 0 < n \wedge$ 
 $\text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (cG\ \varphi\ a\ b\ Bd\ n) <$ 
 $I + (b - a) * Bd + \varepsilon / 2$ 
by  $\text{eventually-elim simp}$ 
then obtain  $n$  where  $0 < n$   $\text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b)$ 
 $(cG\ \varphi\ a$ 
 $b\ Bd\ n) < I + (b - a) * Bd + \varepsilon / 2$ 
by (auto simp:  $\text{eventually-at-top-linorder}$ )
thus  $?thesis$  using  $that$  by blast
qed
define  $T$  where  $T = (\sum\ i < n. cM\ \varphi\ a\ b\ n\ i + Bd)$ 
define  $\delta$  where  $\delta = \varepsilon / (2 * ((2 * real\ n + 1) * Bd + 1))$ 
have  $d0$ :  $0 < \delta$ 
proof -
have  $0 \leq (2 * real\ n + 1) * Bd$  by (intro  $\text{mult-nonneg-nonneg } B0$ ) simp

```


hence $0 < 2 * ((2 * \text{real } n + 1) * Bd + 1)$ **by** *simp*
 thus *?thesis* **unfolding** δ -def **using** *e* **by** *simp*
qed
 have $dB: (2 * \text{real } n + 1) * Bd * \delta \leq \varepsilon / 2$
proof –
 have $0 \leq (2 * \text{real } n + 1) * Bd$ **by** (*intro mult-nonneg-nonneg B0*) *simp*
 hence $0 < 2 * ((2 * \text{real } n + 1) * Bd + 1)$ **by** *simp*
 hence $eq: \delta * (2 * ((2 * \text{real } n + 1) * Bd + 1)) = \varepsilon$ **unfolding** δ -def
by *simp*
 have $(2 * \text{real } n + 1) * Bd * \delta \leq ((2 * \text{real } n + 1) * Bd + 1) * \delta$ **using** *d0*
by *simp*
 also have $\dots = \varepsilon / 2$ **using** *eq* **by** *force*
 finally **show** *?thesis* .
qed
 have $\forall_F K :: \text{nat in at-top. } \forall i < n.$
 $\text{primes-theta } (cx\ a\ b\ n\ (i + 1) * \text{real } K) - \text{primes-theta } (cx\ a\ b\ n\ i * \text{real } K)$
 $\leq$
 $((b - a) / \text{real } n + \delta) * \text{real } K$
proof –
 have $\forall i \in \{..<n\}. \forall_F K :: \text{nat in at-top.}$
 $\text{primes-theta } (cx\ a\ b\ n\ (i + 1) * \text{real } K) - \text{primes-theta } (cx\ a\ b\ n\ i * \text{real } K)$
 $\leq$
 $((b - a) / \text{real } n + \delta) * \text{real } K$
proof
 fix *i* **assume** $i \in \{..<n\}$
 have $0 \leq cx\ a\ b\ n\ i\ 0 \leq cx\ a\ b\ n\ (i + 1)$
using *cx-ge[OF ab, of n i] cx-ge[OF ab, of n i + 1]* **by** *linarith+*
 hence $\forall_F K :: \text{nat in at-top. primes-theta } (cx\ a\ b\ n\ (i + 1) * \text{real } K) -$
 $\text{primes-theta } (cx\ a\ b\ n\ i * \text{real } K)$
 $\leq (cx\ a\ b\ n\ (i + 1) - cx\ a\ b\ n\ i) * \text{real } K$ **by** (*rule cell-upper[OF - -*
d0])
 thus $\forall_F K :: \text{nat in at-top. primes-theta } (cx\ a\ b\ n\ (i + 1) * \text{real } K) -$
 $\text{primes-theta } (cx\ a\ b\ n\ i * \text{real } K)$
 $\leq ((b - a) / \text{real } n + \delta) * \text{real } K$ **unfolding** *cx-succ-sub[OF n0]* .
qed
 hence $\forall_F K :: \text{nat in at-top. } \forall i \in \{..<n\}.$
 $\text{primes-theta } (cx\ a\ b\ n\ (i + 1) * \text{real } K) - \text{primes-theta } (cx\ a\ b\ n\ i * \text{real } K)$
 $\leq$
 $((b - a) / \text{real } n + \delta) * \text{real } K$
by (*rule eventually-ball-finite[OF finite-lessThan]*)
thus *?thesis* **by** (*rule eventually-mono*) *simp*
qed
moreover have $\forall_F K :: \text{nat in at-top. } (b - a - \delta) * \text{real } K \leq$
 $\text{primes-theta } (b * \text{real } K) - \text{primes-theta } (a * \text{real } K)$
by (*rule cell-lower*) (*use a ab d0 in auto*)
ultimately show *?thesis*
using *eventually-gt-at-top[of 0 :: nat]*
proof (*eventually-elim*)
 case (*elim K*)

```

have Kr: 0 < real K using elim(3) by simp
let ?s = {p. p ≤ nat (floor (b * real K)) ∧ prime p ∧ a * real K < real p ∧
  real p ≤ b * real K}
have fin: finite ?s by simp
have seq: ?s = {p. prime p ∧ a * real K < real p ∧ real p ≤ b * real K}
  by (auto intro: le-nat-floor)
have tot: (∑ p ∈ ?s. ln (real p)) =
  primes-theta (b * real K) - primes-theta (a * real K)
unfolding seq by (rule sum-cell) (use a ab Kr in ⟨auto intro: mult-right-mono⟩)
have mem: ∧p. p ∈ ?s ⇒ real p / real K ∈ {a<..b}
  using Kr by (auto simp: less-divide-eq divide-le-eq)
have maj: (∑ p ∈ ?s. (φ (real p / real K) + Bd) * ln (real p)) ≤
  (∑ p ∈ ?s. cG φ a b Bd n (real p / real K) * ln (real p))
proof (rule sum-mono)
  fix p assume p: p ∈ ?s
  have φ (real p / real K) + Bd ≤ cG φ a b Bd n (real p / real K)
    using ab' Bd B0 n0 mem[OF p] by (rule cG-bounds[THEN conjunct1])
  moreover have 0 ≤ ln (real p) using p prime-ge-1-nat[of p] by (intro
    ln-ge-zero) auto
  ultimately show (φ (real p / real K) + Bd) * ln (real p) ≤
    cG φ a b Bd n (real p / real K) * ln (real p)
    by (rule mult-right-mono)
qed
have cGsum: (∑ p ∈ ?s. cG φ a b Bd n (real p / real K) * ln (real p)) ≤
  (∑ i < n. (cM φ a b n i + Bd) * (((b - a) / real n + δ) * real K))
proof -
  have (∑ p ∈ ?s. cG φ a b Bd n (real p / real K) * ln (real p))
    = (∑ i < n. (cM φ a b n i + Bd) * (∑ p ∈ {p ∈
      ?s. real p / real K ∈ {cx a b n i<..cx a b n (i + 1)}}. ln (real p)))
  proof -
    have (∑ p ∈ ?s. cG φ a b Bd n (real p / real K) * ln (real p)) =
      (∑ p ∈ ?s. ∑ i <
        n. indicator {cx a b n i<..cx a b n (i + 1)} (real p / real K) * (cM φ
        a b n i +
          Bd) * ln (real p))
    unfolding cG-def sum-distrib-right ..
    also have ... = (∑ i < n. ∑ p ∈
      ?s. indicator {cx a b n i<..cx a b n (i + 1)} (real p / real K) * (cM φ a
      b n i +
        Bd) * ln (real p))
    by (rule sum.swap)
    also have ... = (∑ i < n. (cM φ a b n i + Bd) * (∑ p ∈ {p ∈
      ?s. real p / real K ∈ {cx a b n i<..cx a b n (i + 1)}}. ln (real p)))
    proof (rule sum.cong[OF refl])
      fix i assume i ∈ {..<n}
      have (∑ p ∈ ?s. indicator {cx a b n i<..cx a b n (i + 1)} (real p / real
        K) *
        (cM φ a b n i + Bd) * ln (real p)) =
        (∑ p ∈ ?s. if real p / real K ∈

```

$\{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}$ then $(cM\ \varphi\ a\ b\ n\ i + Bd) * \ln\ (real\ p)$ else 0)

by (rule sum.cong[OF refl]) (simp add: indicator-def)

also have ... = $(\sum p \in \{p \in ?s.\ real\ p / real\ K \in \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\}.\ (cM\ \varphi\ a\ b\ n\ i + Bd) * \ln\ (real\ p))$

by (rule sum.inter-filter[OF fin, symmetric])

also have ... = $(cM\ \varphi\ a\ b\ n\ i + Bd) * (\sum p \in \{p \in ?s.\ real\ p / real\ K \in \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\}.\ \ln\ (real\ p))$

by (rule sum-distrib-left[symmetric])

finally show $(\sum p \in ?s.\ indicator\ \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\ (real\ p /$

$real\ K) * (cM\ \varphi\ a\ b\ n\ i + Bd) * \ln\ (real\ p)) =$

$(cM\ \varphi\ a\ b\ n\ i + Bd) * (\sum p \in \{p \in ?s.\ real\ p / real\ K \in \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\}.\ \ln\ (real\ p))$.

qed

finally show ?thesis .

qed

also have ... $\leq (\sum i < n.\ (cM\ \varphi\ a\ b\ n\ i + Bd) * (((b - a) / real\ n + \delta) * real\ K))$

proof (rule sum-mono, rule mult-left-mono)

fix i assume i: $i \in \{..<n\}$

show $0 \leq cM\ \varphi\ a\ b\ n\ i + Bd$

proof -

have $-Bd \leq cM\ \varphi\ a\ b\ n\ i$

using Bd less-imp-le[OF ab'] n0 i[unfolded lessThan-iff] by (rule cM-ge)

thus ?thesis by linarith

qed

— chained facts instead of OF: OF with the $\bigwedge y$ -premise unifies higher-order and fails

have $(\sum p \in \{p \in ?s.\ real\ p / real\ K \in \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\}.\ \ln\ (real\ p))$

$\leq (\sum p \in \{p.\ prime\ p \wedge cx\ a\ b\ n\ i * real\ K < real\ p \wedge real\ p \leq cx\ a\ b\ n\ (i + 1) * real\ K\}.\ \ln\ (real\ p))$

proof (rule sum-mono2)

show finite $\{p.\ prime\ p \wedge cx\ a\ b\ n\ i * real\ K < real\ p \wedge real\ p \leq cx\ a\ b\ n\ (i + 1) * real\ K\}$

by (rule finite-subset[OF - finite-primes-le[of $cx\ a\ b\ n\ (i + 1) * real\ K$]])

auto

show $\{p \in ?s.\ real\ p / real\ K \in \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\} \subseteq \{p.\ prime\ p \wedge cx\ a\ b\ n\ i * real\ K < real\ p \wedge real\ p \leq cx\ a\ b\ n\ (i + 1) * real\ K\}$

using Kr by (auto simp: less-divide-eq divide-le-eq)

fix p assume p $\in \{p.\ prime\ p \wedge cx\ a\ b\ n\ i * real\ K < real\ p \wedge real\ p \leq cx\ a\ b\ n\ (i + 1) * real\ K\}$ -

$\{p \in ?s.\ real\ p / real\ K \in \{cx\ a\ b\ n\ i <..cx\ a\ b\ n\ (i + 1)\}\}$

hence prime p by simp

thus $0 \leq \ln\ (real\ p)$ using prime-ge-1-nat[of p] by (intro ln-ge-zero) auto

qed

also have ... = primes-theta $(cx\ a\ b\ n\ (i + 1) * real\ K) - primes-theta$

$(cx\ a\ b\ n\ i\ * \text{real } K)$
proof (*rule sum-cell*)
show $0 \leq cx\ a\ b\ n\ i\ * \text{real } K$ **using** *cx-ge*[*OF ab, of n i*] *a Kr* **by** *auto*
show $cx\ a\ b\ n\ i\ * \text{real } K \leq cx\ a\ b\ n\ (i + 1) * \text{real } K$
using *cx-mono*[*OF ab, of i i + 1 n*] *Kr* **by** *simp*
qed
also have $\dots \leq ((b - a) / \text{real } n + \delta) * \text{real } K$ **using** *elim*(1) *i* **by** *auto*
finally show $(\sum p \in \{p \in ?s. \text{real } p / \text{real } K \in \{cx\ a\ b\ n\ i <.. cx\ a\ b\ n\ (i + 1)\}\}. \ln(\text{real } p)) \leq ((b - a) / \text{real } n + \delta) * \text{real } K$.
qed
finally show *?thesis* .
qed
have *split*: $(\sum p \in ?s. \varphi(\text{real } p / \text{real } K) * \ln(\text{real } p)) = (\sum p \in ?s. (\varphi(\text{real } p / \text{real } K) + Bd) * \ln(\text{real } p)) - Bd * (\sum p \in ?s. \ln(\text{real } p))$
by (*simp add: sum-subtractf sum-distrib-left sum.distrib algebra-simps*)
have *split2*: $(\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * (((b - a) / \text{real } n + \delta) * \text{real } K)) = \text{interval-lebesgue-integral lborel } (ereal\ a) (ereal\ b) (cG\ \varphi\ a\ b\ Bd\ n) * \text{real } K + \delta * \text{real } K * T$
unfolding *integral-cG*[*OF ab' n0*] *T-def*
proof –
have $(\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * (((b - a) / \text{real } n + \delta) * \text{real } K)) = (\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * ((b - a) / \text{real } n) * \text{real } K + \delta * \text{real } K * (cM\ \varphi\ a\ b\ n\ i + Bd))$
by (*rule sum.cong*[*OF refl*] (*rule split2-aux*))
also have $\dots = (\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * ((b - a) / \text{real } n) * \text{real } K) + (\sum i < n. \delta * \text{real } K * (cM\ \varphi\ a\ b\ n\ i + Bd))$
by (*rule sum.distrib*)
also have $(\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * ((b - a) / \text{real } n) * \text{real } K) = (\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * ((b - a) / \text{real } n)) * \text{real } K$
by (*rule sum-distrib-right*[*symmetric*])
also have $(\sum i < n. \delta * \text{real } K * (cM\ \varphi\ a\ b\ n\ i + Bd)) = \delta * \text{real } K * (\sum i < n. cM\ \varphi\ a\ b\ n\ i + Bd)$
by (*rule sum-distrib-left*[*symmetric*])
finally show $(\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * (((b - a) / \text{real } n + \delta) * \text{real } K)) = (\sum i < n. (cM\ \varphi\ a\ b\ n\ i + Bd) * ((b - a) / \text{real } n)) * \text{real } K + \delta * \text{real } K * (\sum i < n. cM\ \varphi\ a\ b\ n\ i + Bd)$.
qed
have $T \leq \text{real } n * (2 * Bd)$
unfolding *T-def*
proof –
have $(\sum i < n. cM\ \varphi\ a\ b\ n\ i + Bd) \leq (\sum i < n. 2 * Bd)$
proof (*rule sum-mono*)

```

    fix i assume i: i ∈ {.. $n$ }
    have cM  $\varphi$  a b n i ≤ Bd
    using Bd B0 less-imp-le[OF ab] n0 i[unfolded lessThan-iff] by (rule cM-le)
    thus cM  $\varphi$  a b n i + Bd ≤ 2 * Bd by linarith
  qed
  thus ( $\sum$  i < n. cM  $\varphi$  a b n i + Bd) ≤ real n * (2 * Bd) by fastforce
  qed
  hence m1:  $\delta$  * real K * T ≤  $\delta$  * real K * (real n * (2 * Bd))
    by (rule mult-left-mono) (use d0 Kr in simp)
  have m2: Bd * ((b - a -  $\delta$ ) * real K) ≤
    Bd * (primes-theta (b * real K) - primes-theta (a * real K))
    by (rule mult-left-mono[OF elim(2) B0])
  have m3: interval-lebesgue-integral lborel (ereal a) (ereal b) (cG  $\varphi$  a b Bd n) *
    real K ≤ (I + (b - a) * Bd +  $\varepsilon$  / 2) * real K
    by (rule mult-right-mono[OF less-imp-le[OF n1]]) (use Kr in simp)
  have m4: (2 * real n + 1) * Bd *  $\delta$  * real K ≤  $\varepsilon$  / 2 * real K
    by (rule mult-right-mono[OF dB]) (use Kr in simp)
  show ?case
    unfolding split tot I-def[symmetric]
    using maj cGsum split2 m1 m2 m3 m4
    by (simp only: algebra-simps) argo
    —  $\sum \varphi \ln p \leq (I + (b-a)B + \varepsilon/2) K + \delta K n 2B - B(b-a-\delta) K =$ 
     $(I + \varepsilon/2) K + (2n+1) B \delta K \leq (I + \varepsilon) K$ 
  qed
  qed

```

Inner range, variable $x = K/p$: substitute $y = 1/x$ in the outer bound.

theorem inner-prime-sum-bound:

```

assumes M:  $3 < M$  and Bd:  $\bigwedge x. x \in \{3..real\ M\} \implies$ 
   $|f\ x| \leq Bd$  and D: countable D
and cont:  $\bigwedge x. x \in \{3<.. $real\ M\} \implies x \notin D \implies$ 
  isCont f x and e:  $0 < \varepsilon$$ 
```

shows $\forall_F K$ in at-top.

```

( $\sum$  p ∈ {p. p ≤ K ∧ prime p ∧ K < p * M ∧
  3 * p ≤ K}. real p * f (real K / real p) * ln (real p))
  ≤ (interval-lebesgue-integral lborel (ereal 3) (ereal (real M)) ( $\lambda x. f\ x /$ 
     $x \wedge 3$ ) +  $\varepsilon$ ) * real K  $\wedge^2$ 

```

proof —

```

have Mr:  $3 < real\ M$  using M by simp
have M0:  $0 < real\ M$  using Mr by simp
let ? $\varphi$  =  $\lambda y. y * f\ (1 / y)$ 
have B0:  $0 \leq Bd$  using Bd[of 3] Mr by auto
have phiB:  $\bigwedge y. y \in \{1 / real\ M..1 / 3\} \implies |?\varphi\ y| \leq Bd$ 

```

proof —

```

fix y assume y:  $y \in \{1 / real\ M..1 / 3\}$ 
have iM:  $0 < 1 / real\ M$  using M0 by simp
have 1 / real M ≤ y using y by simp
hence y0:  $0 < y$  using iM by linarith
have y1:  $y \leq 1$  using y by simp

```

```

have  $3 \leq 1 / y$  using  $y \ y0$  by (simp add: le-divide-eq)
moreover have  $1 / y \leq \text{real } M$  using  $y \ y0 \ M0$  by (simp add: field-simps)
ultimately have  $|f(1 / y)| \leq Bd$  using  $Bd$  by simp
have  $|y * f(1 / y)| = y * |f(1 / y)|$  using  $y0$  by (simp add: abs-mult)
also have  $\dots \leq 1 * Bd$  by (rule mult-mono[OF  $y1 \ fb$ ]) (use  $B0$  in simp-all)
finally show  $|\varphi y| \leq Bd$  by simp
qed
have  $\text{phiC}: \bigwedge y. y \in \{1 / \text{real } M <.. < 1 / 3\} \implies$ 
 $y \notin (\lambda x. 1 / x) \text{ `` } D \implies \text{isCont } \varphi y$ 
proof -
  fix  $y$  assume  $y: y \in \{1 / \text{real } M <.. < 1 / 3\}$  and  $yD: y \notin (\lambda x. 1 / x) \text{ `` } D$ 
  have  $iM: 0 < 1 / \text{real } M$  using  $M0$  by simp
  have  $1 / \text{real } M < y$  using  $y$  by simp
  hence  $y0: 0 < y$  using  $iM$  by linarith
  have  $3 < 1 / y$  using  $y \ y0$  by (simp add: less-divide-eq)
  moreover have  $1 / y < \text{real } M$  using  $y \ y0 \ M0$  by (simp add: field-simps)
  moreover have  $1 / y \notin D$ 
  proof
    assume  $1 / y \in D$ 
    hence  $1 / (1 / y) \in (\lambda x. 1 / x) \text{ `` } D$  by (rule imageI)
    thus False using  $yD$  by simp
  qed
  ultimately have  $c2: \text{isCont } f(1 / y)$  using  $\text{cont}$  by simp
  have  $c1: \text{isCont } (\lambda y. 1 / y) \ y$  using  $y0$  by (auto intro: continuous-intros)
  have  $\text{isCont } (\lambda y. f(1 / y)) \ y$  using  $\text{isCont-o2}[OF \ c1 \ c2]$  by simp
  thus  $\text{isCont } \varphi y$  by (auto intro: continuous-intros)
qed
have  $ho: \forall_F K \text{ in } \text{at-top.}$ 
 $(\sum p \in \{p. p \leq \text{nat } (\text{floor } (1 / 3 * \text{real } K)) \wedge \text{prime } p \wedge$ 
 $1 / \text{real } M * \text{real } K < \text{real } p \wedge \text{real } p \leq 1 / 3 * \text{real } K\}.$ 
 $\varphi(\text{real } p / \text{real } K) * \ln(\text{real } p))$ 
 $\leq (\text{interval-lebesgue-integral lborel } (\text{ereal } (1 / \text{real } M)) (\text{ereal } (1 / 3))) \varphi +$ 
 $\varepsilon) * \text{real } K$ 
  by (rule outer-prime-sum-bound[OF - - phiB countable-image[OF  $D$ ] phiC e])
(use  $M0 \ Mr$  in
   $\langle \text{auto simp: divide-le-eq} \rangle$ )
  have  $\text{subst}: \text{interval-lebesgue-integral lborel } (\text{ereal } (1 / \text{real } M)) (\text{ereal } (1 / 3))$ 
 $\varphi$ 
 $= \text{interval-lebesgue-integral lborel } (\text{ereal } 3) (\text{ereal } (\text{real } M)) (\lambda x. f x / x \wedge 3)$ 
  proof -
    have  $m: (\lambda x. \text{indicator } \{3 <.. < \text{real } M\} \ x * f x) \in \text{borel-measurable borel}$ 
    using  $D \ \text{cont}$  by (rule phi-measurable)
    show  $\varphi$ thesis using  $\text{less-imp-le}[OF \ Mr] \ m \ Bd$  by (rule integral-inv-subst)
  qed
show  $\varphi$ thesis
  using  $ho \ \text{eventually-gt-at-top}[of \ 0 :: \text{nat}]$ 
proof (eventually-elim)
  case (elim  $K$ )
  have  $Kr: 0 < \text{real } K$  using  $\text{elim}(2)$  by simp

```

```

have seq: {p. p ≤ nat (floor (1 / 3 * real K)) ∧ prime p ∧
  1 / real M * real K < real p ∧ real p ≤ 1 / 3 * real K}
  = {p. p ≤ K ∧ prime p ∧ K < p * M ∧ 3 * p ≤ K}
proof (intro equalityI subsetI)
  fix p assume p ∈ {p. p ≤ nat (floor (1 / 3 * real K)) ∧ prime p ∧
    1 / real M * real K < real p ∧ real p ≤ 1 / 3 * real K}
  hence p: prime p 1 / real M * real K < real p real p ≤ 1 / 3 * real K by
auto
  have real K < real p * real M using p(2) M0 by (simp add: field-simps)
  hence b1: K < p * M by (metis of-nat-mult of-nat-less-iff)
  have 3 * real p ≤ real K using p(3) by simp
  hence b2: 3 * p ≤ K by simp
  show p ∈ {p. p ≤ K ∧ prime p ∧ K < p * M ∧ 3 * p ≤ K}
    using p(1) b1 b2 by simp
next
  fix p assume p ∈ {p. p ≤ K ∧ prime p ∧ K < p * M ∧ 3 * p ≤ K}
  hence p: prime p K < p * M 3 * p ≤ K by auto
  have a1: real K < real p * real M using p(2) by (metis of-nat-mult
of-nat-less-iff)
  have a2: 3 * real p ≤ real K using p(3) by simp
  have c1: 1 / real M * real K < real p using a1 M0 by (simp add: field-simps)
  have c2: real p ≤ 1 / 3 * real K using a2 by simp
  have c3: p ≤ nat (floor (1 / 3 * real K)) using c2 by (rule le-nat-floor)
  show p ∈ {p. p ≤ nat (floor (1 / 3 * real K)) ∧ prime p ∧
    1 / real M * real K < real p ∧ real p ≤ 1 / 3 * real K}
    using p(1) c1 c2 c3 by simp
qed
have trm: ∧p. prime p ⇒
  real p * f (real K / real p) * ln (real p) =
  real K * (?φ (real p / real K) * ln (real p))
proof -
  fix p :: nat assume p: prime p
  have e1: 1 / (real p / real K) = real K / real p by simp
  have e2: real K * (real p / real K) = real p using Kr by simp
  show real p * f (real K / real p) * ln (real p) =
    real K * (?φ (real p / real K) * ln (real p))
    unfolding e1 using e2 by (auto simp: mult.assoc[symmetric])
qed
have (∑ p ∈ {p. p ≤ K ∧ prime p ∧ K < p * M ∧
  3 * p ≤ K}. real p * f (real K / real p) * ln (real p))
  = real K * (∑ p ∈ {p. p ≤ nat (floor (1 / 3 * real K)) ∧ prime p ∧
    1 / real M * real K < real p ∧ real p ≤ 1 / 3 * real K}.
    ?φ (real p / real K) * ln (real p))
  unfolding seq sum-distrib-left
  by (rule sum.cong[OF refl]) (rule trm, simp)
— by rule, not simp: simp normalises 1/(p/K) back to K/p and loops
also have ... ≤ real K * ((interval-lebesgue-integral lborel (ereal (1 /
  real M)) (ereal (1 / 3))) ?φ + ε) * real K
  by (rule mult-left-mono[OF elim(1)]) (use Kr in simp)

```

```

    finally show ?case unfolding subst
      by (simp add: power2-eq-square mult-ac)
  qed
qed
end

```

```

theory Piecewise-Interval-Integrals
  imports HOL-Analysis.Analysis
begin

```

7 Interval integrals of piecewise polynomial and rational functions

Integrability and the value of *interval-lebesgue-integral* for functions that agree on an open interval with a quadratic polynomial, with $c_0 + c_1/x$, or with a constant, and the chaining of adjacent pieces. Together these give exact integrals of functions defined piecewise through *floor* and *frac*.

7.1 Generic piece lemmas

The integrand may be changed on the open interval.

```

lemma interval-integral-cong-Ioo:
  fixes f g :: real  $\Rightarrow$  real
  assumes  $a \leq b \wedge x. x \in \{a <..< b\} \implies f x = g x$ 
  shows interval-lebesgue-integral lborel (ereal a) (ereal b) f =
    interval-lebesgue-integral lborel (ereal a) (ereal b) g
proof -
  have set-lebesgue-integral lborel  $\{a <..< b\} f =$  set-lebesgue-integral lborel  $\{a <..< b\}$ 
    g
    using assms(2) set-lebesgue-integral-cong[of  $\{a <..< b\}$  lborel f g] by auto
  thus ?thesis unfolding interval-lebesgue-integral-def using assms(1) by auto
qed

```

```

lemma interval-integrable-cong-Ioo:
  fixes f g :: real  $\Rightarrow$  real
  assumes  $a \leq b \wedge x. x \in \{a <..< b\} \implies$ 
     $f x = g x$  interval-lebesgue-integrable lborel (ereal a) (ereal b) f
  shows interval-lebesgue-integrable lborel (ereal a) (ereal b) g
proof -
  have set-integrable lborel  $\{a <..< b\} f =$  set-integrable lborel  $\{a <..< b\} g$ 
    using assms(2) set-integrable-cong[of lborel lborel  $\{a <..< b\}$   $\{a <..< b\}$  f g] by
    fastforce
  thus ?thesis using assms(1,3) unfolding interval-lebesgue-integrable-def by
    simp
qed

```


A continuous integrand is integrable on a bounded interval.

lemma *interval-integrable-continuous:*

fixes $f :: \text{real} \Rightarrow \text{real}$
assumes $a \leq b$ *continuous-on* $\{a..b\}$ f
shows *interval-lebesgue-integrable lborel* $(\text{ereal } a)$ $(\text{ereal } b)$ f
using *interval-integrable-continuous-on* $[OF \text{ assms}]$ **by** *meson*

Quadratic piece: $\int -a^2 (c0 + c1 x + c2 x^2) dx$.

lemma *piece-quad:*

fixes $f :: \text{real} \Rightarrow \text{real}$
assumes $a \leq b \wedge x. x \in \{a..b\} \implies f x = c0 + c1 * x + c2 * x^2$
shows *interval-lebesgue-integrable lborel* $(\text{ereal } a)$ $(\text{ereal } b)$ $f \wedge$
interval-lebesgue-integral lborel $(\text{ereal } a)$ $(\text{ereal } b)$ f
 $= (c0 * b + c1 * b^2 / 2 + c2 * b^3 / 3) - (c0 * a + c1 * a^2 / 2 + c2 * a^3 / 3)$

proof –

let $?g = \lambda x. c0 + c1 * x + c2 * x^2$
let $?F = \lambda x. c0 * x + c1 * x^2 / 2 + c2 * x^3 / 3$
have *cont: continuous-on* $\{a..b\}$ $?g$ **by** (*intro continuous-intros*)
have I : *interval-lebesgue-integrable lborel* $(\text{ereal } a)$ $(\text{ereal } b)$ $?g$
by (*rule interval-integrable-continuous* $[OF \text{ assms}(1) \text{ cont}]$)
have D : $\wedge x. x \in \{a..b\} \implies$
 $(?F \text{ has-vector-derivative } ?g x) \text{ (at } x \text{ within } \{a..b\})$
proof –
fix x **assume** $x: x \in \{a..b\}$
have $((\lambda x. c0 * x + c1 * x^2 / 2 + c2 * x^3 / 3) \text{ has-real-derivative } (c0$
 $+ c1 * x +$
 $c2 * x^2)) \text{ (at } x \text{ within } \{a..b\})$
using x **by** (*fastforce simp: power2-eq-square intro: derivative-eq-intros*)
thus $(?F \text{ has-vector-derivative } ?g x) \text{ (at } x \text{ within } \{a..b\})$
unfolding *has-real-derivative-iff-has-vector-derivative* .

qed

have *interval-lebesgue-integral lborel* $(\text{ereal } a)$ $(\text{ereal } b)$ $?g = ?F b - ?F a$

proof –

have $mm1$: $\min a b = a$ **using** $\text{assms}(1)$ **by** *simp*
have $mm2$: $\max a b = b$ **using** $\text{assms}(1)$ **by** *simp*
show $?thesis$ **using** *interval-integral-FTC-finite* $[of a b ?g ?F]$ *cont D unfolding*
 $mm1 mm2$
by *presburger*

qed

moreover **have** *interval-lebesgue-integrable lborel* $(\text{ereal } a)$ $(\text{ereal } b)$ f
by (*rule interval-integrable-cong-Ioo* $[OF \text{ assms}(1) - I]$) (*simp add: assms(2)*)
moreover **have** *interval-lebesgue-integral lborel* $(\text{ereal } a)$ $(\text{ereal } b)$ $f =$
interval-lebesgue-integral lborel $(\text{ereal } a)$ $(\text{ereal } b)$ $?g$
by (*rule interval-integral-cong-Ioo* $[OF \text{ assms}(1)]$) (*rule assms(2)*)
ultimately show $?thesis$ **by** *simp*

qed

Rational piece: $\int -a \wedge b (A x + B) / x^3 dx$ for $0 < a$.

lemma *piece-rat*:

```

fixes  $f :: \text{real} \Rightarrow \text{real}$ 
assumes  $0 < a \leq b \wedge x. x \in \{a..b\} \Rightarrow f x = A * x + B$ 
shows  $\text{interval-lebesgue-integrable lborel (ereal a) (ereal b) } (\lambda x. f x / x^3) \wedge$ 
 $\text{interval-lebesgue-integral lborel (ereal a) (ereal b) } (\lambda x. f x / x^3)$ 
 $= (- A / b - B / (2 * b^2)) - (- A / a - B / (2 * a^2))$ 
proof -
  let  $?g = \lambda x. (A * x + B) / x^3$ 
  let  $?F = \lambda x. - A / x - B / (2 * x^2)$ 
  have cont:  $\text{continuous-on } \{a..b\} \ ?g$  using assms(1) by (intro continuous-intros)
  auto
  have I:  $\text{interval-lebesgue-integrable lborel (ereal a) (ereal b) } ?g$ 
    by (rule interval-integrable-continuous[OF assms(2) cont])
  have D:  $\bigwedge x. x \in \{a..b\} \Rightarrow$ 
     $(?F \text{ has-vector-derivative } ?g x) \text{ (at } x \text{ within } \{a..b\})$ 
  proof -
    fix  $x$  assume  $x: x \in \{a..b\}$ 
    have  $x \neq 0$  using assms(1)  $x$  by auto
    hence  $((\lambda x. - A / x - B / (2 * x^2)) \text{ has-real-derivative } ((A * x + B) /$ 
 $x^3)) \text{ (at } x \text{ within } \{a..b\})$ 
    by (fastforce simp: power2-eq-square power3-eq-cube field-simps intro: deriva-
tive-eq-intros)
    thus  $(?F \text{ has-vector-derivative } ?g x) \text{ (at } x \text{ within } \{a..b\})$ 
    unfolding has-real-derivative-iff-has-vector-derivative .
  qed
  have  $\text{interval-lebesgue-integral lborel (ereal a) (ereal b) } ?g = ?F b - ?F a$ 
  proof -
    have mm1:  $\min a b = a$  using assms(2) by simp
    have mm2:  $\max a b = b$  using assms(2) by simp
    show ?thesis using interval-integral-FTC-finite[of a b ?g ?F] cont D unfolding
mm1 mm2
    by presburger
  qed
  moreover have  $\text{interval-lebesgue-integrable lborel (ereal a) (ereal b) } (\lambda x. f x / x$ 
 $^3)$ 
    by (rule interval-integrable-cong-Ioo[OF assms(2) - I]) (simp add: assms(3))
  moreover have  $\text{interval-lebesgue-integral lborel (ereal a) (ereal b) } (\lambda x. f x / x$ 
 $^3) =$ 
 $\text{interval-lebesgue-integral lborel (ereal a) (ereal b) } ?g$ 
    by (rule interval-integral-cong-Ioo[OF assms(2)]) (simp add: assms(3))
  ultimately show ?thesis by simp
qed

```

Adjacent pieces add up.

lemma *chain*:

```

fixes  $f :: \text{real} \Rightarrow \text{real}$ 
assumes  $a \leq b \leq c$ 
and  $\text{interval-lebesgue-integrable lborel (ereal a) (ereal b) } f \wedge$ 

```

$interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ a)\ (ereal\ b)\ f = v$
and $interval\text{-}lebesgue\text{-}integrable\ lborel\ (ereal\ b)\ (ereal\ c)\ f \wedge$
 $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ b)\ (ereal\ c)\ f = w$
shows $interval\text{-}lebesgue\text{-}integrable\ lborel\ (ereal\ a)\ (ereal\ c)\ f \wedge$
 $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ a)\ (ereal\ c)\ f = v + w$
proof –
have $I: interval\text{-}lebesgue\text{-}integrable\ lborel\ (ereal\ a)\ (ereal\ c)\ f$
proof –
have $I1: set\text{-}integrable\ lborel\ \{a < .. < b\}\ f$ **using** $assms(1,3)$
unfolding $interval\text{-}lebesgue\text{-}integrable\text{-}def$ **by** $force$
have $I2: set\text{-}integrable\ lborel\ \{b < .. < c\}\ f$ **using** $assms(2,4)$
unfolding $interval\text{-}lebesgue\text{-}integrable\text{-}def$ **by** $force$
have $set\text{-}integrable\ lborel\ (\{a < .. < b\} \cup \{b < .. < c\})\ f$ **using** $set\text{-}integrable\text{-}Un[OF\ I1\ I2]$
by $simp$
moreover **have** $set\text{-}integrable\ lborel\ (\{a < .. < b\} \cup \{b < .. < c\})\ f =$
 $set\text{-}integrable\ lborel\ \{a < .. < c\}\ f$
using $set\text{-}integrable\text{-}discrete\text{-}difference[where\ X = \{b\}\ and\ A =$
 $\{a < .. < b\} \cup \{b < .. < c\}\ and\ B = \{a < .. < c\}\ and\ M = lborel\ and\ f = f]$
 $assms(1,2)$
by $force$
ultimately **have** $set\text{-}integrable\ lborel\ \{a < .. < c\}\ f$ **by** $meson$
thus $?thesis$ **unfolding** $interval\text{-}lebesgue\text{-}integrable\text{-}def$ **using** $assms(1,2)$ **by**
 $force$
qed
have $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ a)\ (ereal\ c)\ f$
 $= interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ a)\ (ereal\ b)\ f +$
 $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ b)\ (ereal\ c)\ f$
using $interval\text{-}integral\text{-}sum[of\ ereal\ a\ ereal\ b\ ereal\ c\ f]\ I\ assms(1,2)$
by $(simp\ add: min\text{-}def\ max\text{-}def)$
thus $?thesis$ **using** $I\ assms(3,4)$ **by** $simp$
qed

lemma $const\text{-}piece:$

fixes $h :: real \Rightarrow real$
assumes $a \leq b \wedge z. z \in \{a < .. < b\} \implies h\ z = c$
shows $interval\text{-}lebesgue\text{-}integrable\ lborel\ (ereal\ a)\ (ereal\ b)\ h \wedge$
 $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ a)\ (ereal\ b)\ h = (b - a) * c$
using $piece\text{-}quad[OF\ assms(1), of\ h\ c\ 0\ 0]\ assms(2)$ **by** $(simp\ add: algebra\text{-}simps)$

lemma $floor\text{-}inv\text{-}eq:$

fixes $y :: real$
assumes $0 < y\ of\text{-}int\ k * y \leq 1\ 1 < (of\text{-}int\ k + 1) * y$
shows $floor\ (1 / y) = k$
unfolding $floor\text{-}eq\text{-}iff$ **using** $assms$ **by** $(auto\ simp: le\text{-}divide\text{-}eq\ divide\text{-}less\text{-}eq)$

lemma $frac\text{-}nat:$

fixes $u :: real$
assumes $real\ k \leq u\ u < real\ k + 1$

```

  shows  $\text{frac } u = u - \text{real } k$ 
proof -
  have  $\text{floor } u = \text{int } k$  using assms by (auto simp: floor-eq-iff)
  thus ?thesis by (fastforce simp: frac-def)
qed

```

end

```

theory Laplace-Sin-Cot-Series
  imports HOL-Analysis.Analysis Cotangent-PFD-Formula.Cotangent-PFD-Formula
begin

```

8 Gamma-type integrals, the Laplace transform of $\sin$ and the $\coth$ series

Integrals $\int_0^\infty y^k e^{-cy} dy$ on the open half-line by the fundamental theorem of calculus, the interchange of $\int$ and $\sum$ for non-negative series, the Mittag-Leffler expansion of $\coth$ (from the AFP's cotangent partial-fraction formula), the series $\sum_l u / (u^2 + (2\pi l)^2)$, the Laplace transform $\int_0^\infty e^{-cy} \sin(ay) dy$, and the integrals $\Psi(a, c) = \int_0^\infty e^{-cy} y / (y^2 + a^2) dy$, $\Phi(a, c) = \int_0^\infty y^5 e^{-cy} / (y^2 + a^2) dy$ and the geometric pole sums.

8.1 Gamma integrals and the interchange of $\int$ and $\sum$

```

lemma set-bm-cont: continuous-on UNIV  $f \implies$ 
  set-borel-measurable lborel  $\{0 < ..\}$  ( $f :: \text{real} \Rightarrow \text{real}$ )
  unfolding set-borel-measurable-def
  by (intro borel-measurable-scaleR borel-measurable-indicator) (simp-all add:
    borel-measurable-continuous-onI)

lemma ftc-Ioi-nonneg:
  fixes  $f F :: \text{real} \Rightarrow \text{real}$ 
  assumes  $f: f \in \text{borel-measurable borel}$ 
    and  $D: \bigwedge x. 0 \leq x \implies (F \text{ has-real-derivative } f x) \text{ (at } x)$ 
    and  $nn: \bigwedge x. 0 \leq x \implies 0 \leq f x$ 
    and  $T: (F \longrightarrow T) \text{ at-top}$ 
  shows set-integrable lborel  $\{0 < ..\} f \wedge \text{set-lebesgue-integral lborel } \{0 < ..\} f = T$ 
  -  $F 0$ 
proof -
  have mono:  $F 0 \leq F x$  if  $x: 0 \leq x$  for  $x$ 
    by (rule DERIV-nonneg-imp-nondecreasing[OF  $x$ ]) (use  $D nn$  in blast)
  have ev: eventually  $(\lambda x. F 0 \leq F x)$  at-top
    using eventually-ge-at-top[of  $0 :: \text{real}$ ] by eventually-elim (rule mono)
  have T0:  $F 0 \leq T$  by (rule tendsto-lowerbound[OF  $T ev$ ]) simp
  define  $g$  where  $g = (\lambda x. \text{indicator } \{0 ..\} x * f x :: \text{real})$ 

```

```

have nnI: ( $\int^+ x \in \{0..\}$ . ennreal (f x)  $\partial$ lborel) = ennreal (T - F 0)
  by (rule nn-integral-FTC-atLeast[OF f D nn T])
have e: ( $\int^+ x$ . ennreal (g x)  $\partial$ lborel) = ennreal (T - F 0)
  unfolding nnI[symmetric] g-def by (intro nn-integral-cong) (simp split: split-indicator)
have gm: g  $\in$  borel-measurable lborel unfolding g-def using f by measurable
have hb: has-bochner-integral lborel g (T - F 0)
  by (rule has-bochner-integral-nn-integral[OF gm - e]) (auto intro!: AE-I2 simp:
g-def nn T0
  split: split-indicator)
have gs: ( $\lambda x$ . indicator {0..} x  $\ast_R$  f x) = g by (simp add: g-def fun-eq-iff)
have i0: set-integrable lborel {0..} f
  unfolding set-integrable-def gs by (rule integrable.intros[OF hb])
have v0: set-lebesgue-integral lborel {0..} f = T - F 0
  unfolding set-lebesgue-integral-def gs by (rule has-bochner-integral-integral-eq[OF
hb])
have i1: set-integrable lborel {0<..} f by (rule set-integrable-subset[OF i0]) auto
have v1: set-lebesgue-integral lborel {0<..} f = set-lebesgue-integral lborel {0..}
f
proof (rule set-integral-cong-set)
  show set-borel-measurable lborel {0..} f set-borel-measurable lborel {0<..} f
    using i0 i1 unfolding set-integrable-def set-borel-measurable-def
    by (auto dest: borel-measurable-integrable)
  show AE x in lborel. ((x::real)  $\in$  {0..}) = (x  $\in$  {0<..})
    using AE-lborel-singleton[of 0] by eventually-elim (auto simp: le-less)
qed
show ?thesis using i1 v1 v0 by simp
qed

lemma gam-deriv:
  fixes c :: real
  shows (( $\lambda y$ . exp (- (c * y)) * ( $\sum k \leq$ 
    n. (c * y)  $\wedge$  k / fact k)) has-real-derivative
    - c * exp (- (c * y)) * ((c * y)  $\wedge$  n / fact n)) (at y)
proof (induction n)
  case 0
  show ?case by (auto intro!: derivative-eq-intros)
next
  case (Suc n)
  have e': (( $\lambda y$ . exp (- (c * y))) has-real-derivative - c * exp (- (c * y))) (at y)
    by (auto intro!: derivative-eq-intros)
  have p0: (( $\lambda y$ . y  $\wedge$  Suc n) has-real-derivative of-nat (Suc n) * y  $\wedge$  n) (at y)
    using DERIV-pow[of Suc n y] by simp
  have p: (( $\lambda y$ . (c * y)  $\wedge$  Suc n / fact (Suc n)) has-real-derivative c * ((c * y)  $\wedge$ 
n /
  fact n)) (at y)
  proof -
    have h1: (( $\lambda y$ . (c  $\wedge$  Suc n / fact (Suc n)) * y  $\wedge$  Suc n) has-real-derivative (c  $\wedge$ 
Suc n /
  fact (Suc n)) * (of-nat (Suc n) * y  $\wedge$  n)) (at y)

```

by (rule DERIV-cmult[OF p0])
 have h2: $(\lambda y. (c \wedge \text{Suc } n / \text{fact } (\text{Suc } n)) * y \wedge \text{Suc } n) =$
 $(\lambda y. (c * y) \wedge \text{Suc } n / \text{fact } (\text{Suc } n))$
 by (simp add: fun-eq-iff power-mult-distrib)
 have f: $\text{fact } (\text{Suc } n) = (\text{of-nat } (\text{Suc } n) * \text{fact } n :: \text{real})$ by (rule fact-Suc)
 have h3: $(c \wedge \text{Suc } n / \text{fact } (\text{Suc } n)) * (\text{of-nat } (\text{Suc } n) * y \wedge n) = c * ((c * y) \wedge n / \text{fact } n)$
 unfolding f by (simp add: power-mult-distrib field-simps del: of-nat-Suc)
 show ?thesis using h1 unfolding h2 h3 .
 qed
 have d: $((\lambda y. \exp(-(c * y)) * ((c * y) \wedge \text{Suc } n / \text{fact } (\text{Suc } n))) \text{ has-real-derivative } (-(c * \exp(-(c * y)) * ((c * y) \wedge \text{Suc } n / \text{fact } (\text{Suc } n)) + c * \exp(-(c * y)) * ((c * y) \wedge n / \text{fact } n))) (at y))$
 by (rule DERIV-cong[OF DERIV-mult[OF e' p]]) (simp add: algebra-simps)
 show ?case unfolding sum.atMost-Suc distrib-left
 by (rule DERIV-cong[OF DERIV-add[OF Suc.IH d]]) (simp add: algebra-simps)
 qed

lemma gam-lim:

fixes $c :: \text{real}$
 assumes $c: 0 < c$
 shows $((\lambda y. \exp(-(c * y)) * (\sum k \leq n. (c * y) \wedge k / \text{fact } k)) \longrightarrow 0) \text{ at-top}$
 proof -
 have cy: filterlim $(\lambda y. c * y)$ at-top at-top
 by (rule filterlim-tendsto-pos-mult-at-top[OF tendsto-const c filterlim-ident])
 have $((\lambda y. \sum k \leq n. (c * y) \wedge k / \exp(c * y) / \text{fact } k) \longrightarrow 0) \text{ at-top}$
 proof (rule tendsto-null-sum)
 fix $k :: \text{nat}$ assume $k \in \{..n\}$
 have $((\lambda y. (c * y) \wedge k / \exp(c * y)) \longrightarrow 0) \text{ at-top}$
 using filterlim-compose[OF tendsto-power-div-exp-0 cy] by (simp add: o-def)
 thus $((\lambda y. (c * y) \wedge k / \exp(c * y) / \text{fact } k) \longrightarrow 0) \text{ at-top}$
 by (rule tendsto-divide-zero)
 qed
 moreover have $(\lambda y. \sum k \leq n. (c * y) \wedge k / \exp(c * y) / \text{fact } k) =$
 $(\lambda y. \exp(-(c * y)) * (\sum k \leq n. (c * y) \wedge k / \text{fact } k))$
 by (simp add: fun-eq-iff sum-distrib-left exp-minus field-simps)
 ultimately show ?thesis by simp
 qed

lemma exp-neg-lim: $0 < (c :: \text{real}) \implies$
 $((\lambda y. \exp(-(c * y))) \longrightarrow 0) \text{ at-top}$
 using gam-lim[of c 0] by simp

lemma gam-both:

fixes $c :: \text{real}$
 assumes $c: 0 < c$

shows *set-integrable lborel* $\{0 < ..\}$ $(\lambda y. y^{\wedge} n * \exp(- (c * y))) \wedge$
set-lebesgue-integral lborel $\{0 < ..\}$ $(\lambda y. y^{\wedge} n * \exp(- (c * y))) =$
 $\text{fact } n / c^{\wedge} (n + 1)$
proof –
define F **where** $F = (\lambda y. - (\text{fact } n / c^{\wedge} (n + 1)) * (\exp(- (c * y)) * (\sum k$
 $\leq$
 $n. (c * y)^{\wedge} k / \text{fact } k)))$
have D : $(F \text{ has-real-derivative } y^{\wedge} n * \exp(- (c * y)))$ **(at y) for y**
proof –
have $(F \text{ has-real-derivative } - (\text{fact } n / c^{\wedge} (n + 1)) * (- c * \exp(- (c * y)))$
 $* ((c * y)^{\wedge} n /$
 $\text{fact } n)))$ **(at y)**
unfolding $F\text{-def}$ **by** $(\text{intro DERIV-cmult gam-deriv})$
moreover have $- (\text{fact } n / c^{\wedge} (n + 1)) * (- c * \exp(- (c * y))) * ((c * y)$
 $^{\wedge} n / \text{fact } n)) =$
 $y^{\wedge} n * \exp(- (c * y))$
using c **by** $(\text{simp add: power-mult-distrib field-simps})$
ultimately show $?thesis$ **by** simp
qed
have $F0$: $F\ 0 = - (\text{fact } n / c^{\wedge} (n + 1))$
by $(\text{simp add: F-def atMost-atLeast0 sum.atLeast-Suc-atMost[of 0 n] power-0-left})$
have T : $(F \longrightarrow 0)$ **at-top** **unfolding** $F\text{-def}$
by $(\text{rule tendsto-mult-right-zero[OF gam-lim[OF c]])}$
have m : $(\lambda y. y^{\wedge} n * \exp(- (c * y))) \in \text{borel-measurable borel}$ **by** measurable
have *set-integrable lborel* $\{0 < ..\}$ $(\lambda y. y^{\wedge} n * \exp(- (c * y))) \wedge$
set-lebesgue-integral lborel $\{0 < ..\}$ $(\lambda y. y^{\wedge} n * \exp(- (c * y))) = 0 - F\ 0$
by $(\text{rule ftc-Ioi-nonneg[OF m D - T]})$ simp
thus $?thesis$ **unfolding** $F0$ **by** simp
qed

lemma *gam-integrable*: $0 < (c::\text{real}) \implies$
set-integrable lborel $\{0 < ..\}$ $(\lambda y. y^{\wedge} n * \exp(- (c * y)))$
by $(\text{rule gam-both[THEN conjunct1]})$

lemma *gam*: $0 < (c::\text{real}) \implies$
set-lebesgue-integral lborel $\{0 < ..\}$ $(\lambda y. y^{\wedge} n * \exp(- (c * y))) = \text{fact } n / c^{\wedge}$
 $(n + 1)$
by $(\text{rule gam-both[THEN conjunct2]})$

Interchange of $\int$ and $\sum$ for nonnegative integrable terms with summable integrals.

lemma *set-integral-sum-fin*:
fixes $f :: \text{nat} \Rightarrow \text{real} \Rightarrow \text{real}$
assumes $\bigwedge i. i \in I \implies \text{set-integrable lborel } A\ (f\ i)$
shows *set-integrable lborel* $A\ (\lambda x. \sum i \in I. f\ i\ x) \wedge$
set-lebesgue-integral lborel $A\ (\lambda x. \sum i \in I. f\ i\ x) = (\sum i \in$
 $I. \text{set-lebesgue-integral lborel } A\ (f\ i))$
proof –
have i : $\bigwedge i. i \in I \implies$

```

    integrable lborel ( $\lambda x. \text{indicator } A \ x \ *_R \ f \ i \ x$ ) using assms
    unfolding set-integrable-def .
  have e: ( $\lambda x. \text{indicator } A \ x \ *_R \ (\sum i \in I. f \ i \ x)$ ) =
    ( $\lambda x. \sum i \in I. \text{indicator } A \ x \ *_R \ f \ i \ x$ )
    by (simp add: sum-distrib-left)
  show ?thesis unfolding set-integrable-def set-lebesgue-integral-def e
    using Bochner-Integration.integrable-sum[OF i] Bochner-Integration.integral-sum[OF
i]
    by (rule conjI)
qed

```

```

lemma set-integrable-cong-Ioi:
  set-integrable lborel  $\{0 < ..\}$   $f \implies (\bigwedge x. 0 < x \implies$ 
     $f \ x = g \ x) \implies \text{set-integrable lborel } \{0 < ..\} (g :: \text{real} \Rightarrow \text{real})$ 
  using set-integrable-cong[of lborel lborel  $\{0 < ..\}$   $\{0 < ..\} f g]$  by simp

```

```

lemma set-integral-cong-Ioi:
  ( $\bigwedge x. 0 < x \implies f \ x = g \ x$ )  $\implies$ 
    set-lebesgue-integral lborel  $\{0 < ..\} f =$ 
    set-lebesgue-integral lborel  $\{0 < ..\} (g :: \text{real} \Rightarrow \text{real})$ 
  by (rule set-lebesgue-integral-cong) auto

```

```

lemma integral-suminf-nn:
  fixes  $f :: \text{nat} \Rightarrow \text{real} \Rightarrow \text{real}$ 
  assumes  $\bigwedge n. \text{set-integrable lborel } \{0 < ..\} (f \ n) \wedge n \ y. 0 < y \implies$ 
     $0 \leq f \ n \ y$ 
    and  $\bigwedge y. 0 < y \implies$ 
      summable ( $\lambda n. f \ n \ y$ ) summable ( $\lambda n. \text{set-lebesgue-integral lborel } \{0 < ..\} (f$ 
n)
  shows set-integrable lborel  $\{0 < ..\} (\lambda y. \sum n. f \ n \ y) \wedge$ 
    set-lebesgue-integral lborel  $\{0 < ..\} (\lambda y. \sum n. f \ n \ y) =$ 
    ( $\sum n. \text{set-lebesgue-integral lborel } \{0 < ..\} (f \ n)$ )
proof -
  define  $g$  where  $g = (\lambda n \ x. \text{indicator } \{0 < ..\} \ x \ *_R \ f \ n \ x)$ 
  have  $g_i: \text{integrable lborel } (g \ n) \text{ for } n$  using assms(1)[of n] unfolding set-integrable-def
g-def .
  have  $g_{nn}: 0 \leq g \ n \ x \text{ for } n \ x$  using assms(2)[of x n] by (simp add: g-def
indicator-def)
  have  $g_{nn}': \text{norm } (g \ n \ x) = g \ n \ x \text{ for } n \ x$  using g_{nn}[of n x] by simp
  have  $g_s: \text{summable } (\lambda n. g \ n \ x) \text{ for } x$ 
    using assms(3)[of x] by (cases  $0 < x$ ) (simp-all add: g-def indicator-def)
  have  $AE: AE \ x \text{ in lborel. summable } (\lambda n. \text{norm } (g \ n \ x))$  unfolding  $g_{nn}'$  using
gs by simp
  have  $S: \text{summable } (\lambda n. \text{LINT } x | \text{lborel. norm } (g \ n \ x))$ 
proof -
  have ( $\lambda n. \text{LINT } x | \text{lborel. norm } (g \ n \ x)) =$ 
    ( $\lambda n. \text{set-lebesgue-integral lborel } \{0 < ..\} (f \ n)$ )
    unfolding  $g_{nn}'$  by (simp add: set-lebesgue-integral-def g-def)
  thus ?thesis using assms(4) by simp

```



```

qed
have i: integrable lborel ( $\lambda x. \sum n. g\ n\ x$ ) by (rule integrable-suminf[OF gi AE
S])
have v: lebesgue-integral lborel ( $\lambda x. \sum n. g\ n\ x$ ) =
  ( $\sum n. \text{lebesgue-integral lborel } (g\ n)$ ) by (rule integral-suminf[OF gi AE S])
have sumg: ( $\lambda x. \text{indicator } \{0 < ..\} x *_{\mathbb{R}} (\sum n. f\ n\ x)$ ) =
  ( $\lambda x. \sum n. g\ n\ x$ )
proof (rule ext)
  fix x :: real
  show indicator  $\{0 < ..\} x *_{\mathbb{R}} (\sum n. f\ n\ x) = (\sum n. g\ n\ x)$ 
  proof (cases  $0 < x$ )
    case True thus ?thesis by (simp add: g-def)
  next
    case False thus ?thesis by (simp add: g-def)
  qed
qed
show ?thesis using i v unfolding set-integrable-def set-lebesgue-integral-def sumg
g-def
  by (rule conjI)
qed

```

Converse interchange: if the pointwise sum is integrable, the integrals sum to its integral.

```

lemma suminf-integral-nn:
  fixes f :: nat  $\Rightarrow$  real  $\Rightarrow$  real
  assumes  $\bigwedge n. \text{set-integrable lborel } \{0 < ..\} (f\ n) \wedge n\ y. 0 < y \implies$ 
     $0 \leq f\ n\ y$ 
  and  $\bigwedge y. 0 < y \implies$ 
    summable ( $\lambda n. f\ n\ y$ ) set-integrable lborel  $\{0 < ..\} (\lambda y. \sum n. f\ n\ y)$ 
  shows ( $\lambda n. \text{set-lebesgue-integral lborel } \{0 < ..\} (f\ n)$ ) sums set-lebesgue-integral
  lborel
     $\{0 < ..\} (\lambda y. \sum n. f\ n\ y)$ 
proof -
  have nn-int:  $0 \leq \text{set-lebesgue-integral lborel } \{0 < ..\} (f\ n)$  for n
  using set-integral-mono[of lborel  $\{0 < ..\} \lambda x. 0\ f\ n]$  assms(1,2)
  by (simp add: set-integrable-def)
  have bd:  $(\sum n < N. \text{set-lebesgue-integral lborel } \{0 < ..\} (f\ n)) \leq$ 
    set-lebesgue-integral lborel  $\{0 < ..\} (\lambda y. \sum n. f\ n\ y)$  for N
  proof -
    have fs: set-integrable lborel  $\{0 < ..\} (\lambda x. \sum n < N. f\ n\ x) \wedge$ 
      set-lebesgue-integral lborel  $\{0 < ..\} (\lambda x. \sum n < N. f\ n\ x) = (\sum n <$ 
        N. set-lebesgue-integral lborel  $\{0 < ..\} (f\ n))$ 
    by (rule set-integral-sum-fin) (rule assms(1))
    have set-lebesgue-integral lborel  $\{0 < ..\} (\lambda x. \sum n < N. f\ n\ x) \leq$ 
      set-lebesgue-integral lborel  $\{0 < ..\} (\lambda y. \sum n. f\ n\ y)$ 
    proof (rule set-integral-mono[OF conjunct1[OF fs] assms(4)])
      fix x :: real assume  $x \in \{0 < ..\}$ 
      thus  $(\sum n < N. f\ n\ x) \leq (\sum n. f\ n\ x)$  using assms(2,3)
      by (intro sum-le-suminf) auto
    qed
  qed

```

```

qed
thus ?thesis using fs by simp
qed
have sm: summable (λn. set-lebesgue-integral lborel {0<..} (f n))
  by (rule summableI-nonneg-bounded[OF nn-int bd])
have (∑ n. set-lebesgue-integral lborel {0<..} (f n)) =
  set-lebesgue-integral lborel {0<..} (λy. ∑ n. f n y)
  using integral-suminf-nn[OF assms(1-3) sm] by simp
thus ?thesis using summable-sums[OF sm] by simp
qed

```

Mittag-Leffler for $\coth$ (from *Cotangent-PFD-Formula.cot-pfd-formula-real*).

lemma *coth-series*:

$0 < v \implies (\lambda n. 2 * v / (v ^ 2 + (real n + 1) ^ 2)) \text{ sums } (pi * \cosh (pi * v) / \sinh (pi * v) - 1 / v)$

proof –

assume $v: 0 < v$

define z **where** $z = i * \text{complex-of-real } v$

define x **where** $x = pi * v$

have $zI: z \notin \mathbb{Z}$

proof

assume $z \in \mathbb{Z}$

then obtain m **where** $z = \text{of-int } m$ **by** (*elim Ints-cases*)

hence $\text{Im } z = 0$ **by** *simp*

thus *False* **using** v **by** (*simp add: z-def*)

qed

have $sh0: \sinh x \neq 0$ **using** v **by** (*simp add: x-def*)

have $cosr: \cos (i * \text{complex-of-real } x) = \text{complex-of-real } (\cosh x)$

using $\cosh\text{-real}[of\ x]$ **by** (*simp add: cosh-def exp-minus*)

have $sinr: \sin (i * \text{complex-of-real } x) = i * \text{complex-of-real } (\sinh x)$

proof –

have $\text{complex-of-real } (\sinh x) = -i * \sin (i * \text{complex-of-real } x)$

using $\sinh\text{-real}[of\ x]$ **by** (*simp add: sinh-def exp-minus*)

hence $i * \text{complex-of-real } (\sinh x) = \sin (i * \text{complex-of-real } x)$ **by** *simp*

thus ?thesis **by** *simp*

qed

have $piz: \text{complex-of-real } pi * z = i * \text{complex-of-real } x$ **by** (*simp add: z-def x-def*)

have $\text{complex-of-real } pi * \cot (\text{complex-of-real } pi * z) = 1 / z + \cot\text{-pfd } z$

by (*rule cot-pfd-formula-complex[OF zI]*)

hence $\cot\text{-pfd } z = \text{complex-of-real } pi * (\text{complex-of-real } (\cosh x) / (i * \text{complex-of-real } (\sinh x))) - 1 / z$

unfolding piz $\cot\text{-def}$ $cosr$ $sinr$ **by** (*simp add: algebra-simps*)

hence $c: i * \cot\text{-pfd } z = \text{complex-of-real } (pi * \cosh x / \sinh x - 1 / v)$

using v $sh0$ **by** (*simp add: z-def field-simps*)

have $t: i * (2 * z / (z^2 - (\text{complex-of-nat } (Suc\ n))^2)) = \text{complex-of-real } (2 * v / (v ^ 2 + (real\ n + 1) ^ 2))$ **for** n

proof –

have $d: z^2 - (\text{complex-of-nat } (Suc\ n))^2 =$

$- \text{complex-of-real } (v ^ 2 + (real\ n + 1) ^ 2)$

by (simp add: z-def power2-eq-square algebra-simps)
 have d0: complex-of-real $(v^2 + (\text{real } n + 1)^2) \neq 0$
 by (metis add-pos-nonneg of-real-eq-0-iff less-irrefl zero-less-power v zero-le-power2)
 have g: $i * (2 * (i * c) / - D) = 2 * c / D$ if $D \neq 0$ for $c D :: \text{complex}$
 using that by (simp add: field-simps)
 show ?thesis unfolding d unfolding z-def g[OF d0]
 by (simp only: of-real-divide of-real-mult of-real-numeral)
 qed
 have $(\lambda n. i * (2 * z / (z^2 - (\text{complex-of-nat } (\text{Suc } n))^2))) \text{ sums } (i * \text{cot-pfd } z)$
 by (rule sums-mult[OF sums-cot-pfd-complex])
 hence $(\lambda n. \text{complex-of-real } (2 * v / (v^2 + (\text{real } n + 1)^2))) \text{ sums } \text{complex-of-real } (\pi * \cosh x / \sinh x - 1 / v)$
 unfolding t c .
 thus ?thesis unfolding sums-of-real-iff x-def .
 qed

8.2 The S -series and the Laplace transform of $\sin$

lemma S -series:

$0 < u \implies (\lambda l. u / (u^2 + (2 * \pi * (\text{real } l + 1))^2)) \text{ sums } (1 / 4 + 1 / (2 * (\exp u - 1)) - 1 / (2 * u))$
 proof –
 assume u: $0 < u$
 define v where $v = u / (2 * \pi)$
 have v: $0 < v$ using u by (simp add: v-def)
 have cs: $(\lambda n. 1 / (4 * \pi) * (2 * v / (v^2 + (\text{real } n + 1)^2))) \text{ sums } (1 / (4 * \pi) * (\pi * \cosh (\pi * v) / \sinh (\pi * v) - 1 / v))$
 by (rule sums-mult[OF coth-series[OF v]])
 have t: $1 / (4 * \pi) * (2 * v / (v^2 + (\text{real } n + 1)^2)) = u / (u^2 + (2 * \pi * (\text{real } n + 1))^2)$ for n
 proof –
 have d2: $0 < u^2 + (2 * \pi * (\text{real } n + 1))^2$ using u by (simp add: add-pos-pos)
 have $1 / (4 * \pi) * (2 * v / (v^2 + (\text{real } n + 1)^2)) = v / (2 * \pi * (v^2 + (\text{real } n + 1)^2))$
 by (simp add: field-simps)
 also have $2 * \pi * (v^2 + (\text{real } n + 1)^2) = (u^2 + (2 * \pi * (\text{real } n + 1))^2) / (2 * \pi)$
 by (simp add: v-def field-simps power2-eq-square)
 also have $v / ((u^2 + (2 * \pi * (\text{real } n + 1))^2) / (2 * \pi)) = u / (u^2 + (2 * \pi * (\text{real } n + 1))^2)$
 using d2 by (simp add: v-def field-simps)
 finally show ?thesis .
 qed
 define w where $w = \exp (u / 2)$
 have w1: $1 < w$ using u by (simp add: w-def)
 have ww: $\exp u = w^2$ unfolding w-def using exp-double[of u / 2] by simp

```

have pv: pi * v = u / 2 by (simp add: v-def)
have ch: cosh (u / 2) / sinh (u / 2) = (w ^ 2 + 1) / (w ^ 2 - 1)
proof -
  have w0: w ≠ 0 using w1 by simp
  have c1: cosh (u / 2) = (w + inverse w) / 2 by (simp add: cosh-def exp-minus
w-def)
  have s1: sinh (u / 2) = (w - inverse w) / 2 by (simp add: sinh-def exp-minus
w-def)
  have cosh (u / 2) / sinh (u / 2) = (w + inverse w) / (w - inverse w)
    unfolding c1 s1 divide-divide-times-eq mult.commute[of - 2]
    by (rule nonzero-mult-divide-mult-cancel-left) simp
  also have ... = (w * (w + inverse w)) / (w * (w - inverse w)) using w0 by
simp
  also have w * (w + inverse w) = w * w + 1 using w0 by (simp add:
distrib-left)
  also have w * (w - inverse w) = w * w - 1 using w0 by (simp add:
right-diff-distrib)
  finally show ?thesis by (simp add: power2-eq-square)
qed
have val: 1 / (4 * pi) * (pi * cosh (pi * v) / sinh (pi * v) - 1 / v) =
  1 / 4 + 1 / (2 * (exp u - 1)) - 1 / (2 * u)
proof -
  have w2: w ^ 2 - 1 ≠ 0 using one-less-power[OF w1, of 2] by simp
  have ch': pi * cosh (u / 2) / sinh (u / 2) = pi * ((w ^ 2 + 1) / (w ^ 2 - 1))
    using ch by (metis times-divide-eq-right)
  show ?thesis unfolding pv ww ch' using u w2 by (simp add: v-def field-simps)
qed
show ?thesis using cs unfolding t val .
qed

```

```

lemma exp-integrable: 0 < (a::real) ⇒
  set-integrable lborel {0<..} (λu. exp (- (a * u)))
  using gam-integrable[of a 0] by simp

```

```

lemma laplace-sin-integrable: 0 < (a::real) ⇒
  set-integrable lborel {0<..} (λu. exp (- (a * u)) * sin (y * u))
proof -
  assume a: 0 < a
  show ?thesis
  proof (rule set-integrable-bound[OF exp-integrable[OF a]])
    show set-borel-measurable lborel {0<..} (λu. exp (- (a * u)) * sin (y * u))
      by (rule set-bm-cont) (intro continuous-intros)
    show AE x∈{0<..} in lborel. norm (exp (- (a * x)) * sin (y * x)) ≤
      norm (exp (- (a * x)))
      by (intro AE-I2 impI) (auto simp: abs-mult intro!: mult-left-le abs-sin-le-one)
  qed
qed

```

$$\int -0^\infty e^{\{-au\}} \sin(yu) \, du = y / (y^2 + a^2).$$

```

lemma laplace-sin:  $0 < (a::real) \implies$ 
  set-lebesgue-integral lborel  $\{0 < ..\}$   $(\lambda u. \exp(- (a * u)) * \sin(y * u)) =$ 
   $y / (y^2 + a^2)$ 
proof -
  assume  $a: 0 < a$ 
  have  $D: a^2 + y^2 \neq 0$  using  $a$  by (simp add: add-pos-nonneg)
  have  $Dp: 0 < a^2 + y^2$  using  $a$  by (simp add: add-pos-nonneg)
  define  $G$  where  $G = (\lambda u. (- \exp(- (a * u)) * (a * \sin(y * u) + y * \cos(y * u))) *$ 
   $\text{inverse}(a^2 + y^2) - \exp(- (a * u)) * \text{inverse } a)$ 
  have  $dG: (G \text{ has-real-derivative } \exp(- (a * u)) * (\sin(y * u) + 1)) (at u) \text{ for }$ 
 $u$ 
  unfolding  $G\text{-def}$ 
  by (rule DERIV-cong, (rule derivative-eq-intros refl)+) (use a D in <simp add:
field-simps
power2-eq-square>)
  define  $K$  where  $K = (a + |y|) * \text{inverse}(a^2 + y^2) + \text{inverse } a$ 
  have  $bnd: \text{norm}(G u) \leq \exp(- (a * u)) * K$  for  $u$ 
  proof -
  have  $X: |a * \sin(y * u) + y * \cos(y * u)| \leq a + |y|$ 
  proof -
  have  $|a * \sin(y * u) + y * \cos(y * u)| \leq$ 
 $|a| * |\sin(y * u)| + |y| * |\cos(y * u)|$ 
  using abs-triangle-ineq[of a * sin(y * u) y * cos(y * u)] by (simp add:
abs-mult)
  also have  $\dots \leq a + |y|$ 
  using  $a$  by (intro add-mono) (auto intro!: mult-left-le)
  finally show ?thesis .
qed
have  $\text{norm}(G u) \leq \exp(- (a * u)) * |a * \sin(y * u) + y * \cos(y * u)| *$ 
 $\text{inverse}(a^2 + y^2) + \exp(- (a * u)) * \text{inverse } a$ 
proof -
  have  $\text{norm}(G u) \leq |(- \exp(- (a * u)) * (a * \sin(y * u) + y * \cos(y * u))) *$ 
 $\text{inverse}(a^2 + y^2)| + |\exp(- (a * u)) * \text{inverse } a|$ 
  unfolding  $G\text{-def}$  real-norm-def by (rule abs-triangle-ineq4)
  also have  $|(- \exp(- (a * u)) * (a * \sin(y * u) + y * \cos(y * u))) * \text{inverse}$ 
 $(a^2 + y^2)| =$ 
 $\exp(- (a * u)) * |a * \sin(y * u) + y * \cos(y * u)| * \text{inverse}(a^2 + y^2)$ 
  using  $Dp$  by (simp add: abs-mult)
  also have  $|\exp(- (a * u)) * \text{inverse } a| = \exp(- (a * u)) * \text{inverse } a$  using
 $a$ 
  by (simp add: abs-mult)
  finally show ?thesis .
qed
also have  $\dots \leq \exp(- (a * u)) * (a + |y|) * \text{inverse}(a^2 + y^2) +$ 
 $\exp(- (a * u)) * \text{inverse } a$ 

```

```

    using X Dp by (intro add-mono mult-right-mono mult-left-mono) auto
    also have ... = exp (- (a * u)) * K by (simp add: K-def algebra-simps)
    finally show ?thesis .
qed
have T: (G ⟶ 0) at-top
  by (rule Lim-null-comparison[OF always-eventually[OF allI[OF bnd]] tend-
sto-mult-left-zero[OF
  exp-neg-lim[OF a]]])
have m: (λu. exp (- (a * u)) * (sin (y * u) + 1)) ∈ borel-measurable borel
  by (intro borel-measurable-continuous-onI continuous-intros)
have nn: 0 ≤ exp (- (a * u)) * (sin (y * u) + 1) for u
proof -
  have 0 ≤ sin (y * u) + 1 using sin-ge-minus-one[of y * u] by linarith
  thus ?thesis by simp
qed
have I1: set-integrable lborel {0<..} (λu. exp (- (a * u)) * (sin (y * u) + 1)) ∧
  set-lebesgue-integral lborel {0<..} (λu. exp (- (a * u)) * (sin (y * u) + 1)) =
  0 - G 0
  by (rule ftc-Ioi-nonneg[OF m dG nn T])
have G0: G 0 = - y * inverse (a ^ 2 + y ^ 2) - inverse a by (simp add:
G-def)
have I2: set-lebesgue-integral lborel {0<..} (λu. exp (- (a * u))) = inverse a
  using gam[OF a, of 0] by (simp add: inverse-eq-divide)
have f: (λu. exp (- (a * u)) * sin (y * u)) =
  (λu. exp (- (a * u)) * (sin (y * u) + 1) - exp (- (a * u)))
  by (simp add: fun-eq-iff algebra-simps)
have set-lebesgue-integral lborel {0<..} (λu. exp (- (a * u)) * sin (y * u)) =
  (0 - G 0) - inverse a
  unfolding f set-integral-diff(2)[OF conjunct1[OF I1] exp-integrable[OF a]] I2
  conjunct2[OF I1]
  ..
  also have ... = y / (y ^ 2 + a ^ 2) unfolding G0 by (simp add: field-simps
add.commute)
  finally show ?thesis .
qed

```

$$\Psi(a, c) = \int_{-\infty}^0 e^{-cy} \frac{y}{(y^2 + a^2)} dy.$$

definition *Psi* :: real ⇒ real ⇒ real **where**

Psi a c = set-lebesgue-integral lborel {0<..} (λy. exp (- (c * y)) * (y / (y ^ 2 + a ^ 2)))

lemma *psi-integrable*: 0 < (a::real) ⟹ 0 < c ⟹

set-integrable lborel {0<..} (λy. exp (- (c * y)) * (y / (y ^ 2 + a ^ 2)))

proof -

assume a: 0 < a and c: 0 < c

have Dp: 0 < y ^ 2 + a ^ 2 for y using a by (simp add: add-nonneg-pos)

have b: y / (y ^ 2 + a ^ 2) ≤ 1 / (2 * a) if y: 0 < y for y

proof -

have 0 ≤ (y - a) ^ 2 by simp

hence $2 * a * y \leq y^2 + a^2$ **by** (*simp add: power2-diff algebra-simps*)
 thus *?thesis* **using** *a Dp[of y]* **by** (*simp add: field-simps*)
qed
show *?thesis*
proof (*rule set-integrable-bound*)
 show *set-integrable lborel {0<..}* ($\lambda y. \exp(- (c * y)) * (1 / (2 * a))$)
 by (*rule set-integrable-mult-left[OF exp-integrable[OF c]]*)
 show *set-borel-measurable lborel {0<..}* ($\lambda y. \exp(- (c * y)) * (y / (y^2 + a^2))$)
 by (*rule set-bm-cont*) (*intro continuous-intros, metis Dp less-irrefl*)
 show *AE $x \in \{0<..\}$ in lborel. norm (exp (- (c * x)) * (x / (x^2 + a^2)))*
 $\leq$
 norm ($\exp(- (c * x)) * (1 / (2 * a))$)
proof (*rule AE-I2, rule impI*)
 fix *x :: real* **assume** $x \in \{0<..\}$
 hence *x: 0 < x* **by** *simp*
 have *le: exp (- (c * x)) * (x / (x^2 + a^2)) ≤ exp (- (c * x)) * (1 / (2 * a))*
 by (*rule mult-left-mono[OF b[OF x]] simp*)
 have *n1: norm (exp (- (c * x)) * (x / (x^2 + a^2))) = exp (- (c * x)) * (x / (x^2 + a^2))*
 using *x Dp[of x]* **by** (*simp add: abs-mult*)
 have *n2: norm (exp (- (c * x)) * (1 / (2 * a))) = exp (- (c * x)) * (1 / (2 * a))*
 using *a* **by** (*simp add: abs-mult*)
 show *norm (exp (- (c * x)) * (x / (x^2 + a^2))) ≤ norm (exp (- (c * x)) * (1 / (2 * a)))*
 unfolding *n1 n2* **by** (*rule le*)
qed
qed
qed

The symmetry $\Psi(a, c) = \Psi(c, a)$, by Fubini on $e^{\{-cy\}} e^{\{-au\}} \sin(yu)$.

lemma *psi-symm*: $0 < a \implies 0 < c \implies \Psi a c = \Psi c a$

proof –

assume *a: 0 < a* **and** *c: 0 < c*

define *K* **where** $K = (\lambda y u :: real.$

*if 0 < y ∧ 0 < u then exp (- (c * y)) * (exp (- (a * u)) * sin (y * u)) else 0)*

have *Km: (λ(y, u). K y u) ∈ borel-measurable (lborel ⊗_M lborel)*

unfolding *K-def* **by** *measurable*

have *Kslice: (λu. K y u) = (λu. indicator {0<..} u *_R (exp (- (c * y)) * (exp (- (a * u)) * sin (y * u))))* **if** *y: 0 < y* **for** *y*

using *y* **by** (*auto simp: K-def fun-eq-iff indicator-def*)

have *Kslice2: (λy. K y u) = (λy. indicator {0<..} y *_R (exp (- (a * u)) * (exp (- (c * y)) * sin (u * y))))* **if** *u: 0 < u* **for** *u*

using *u* **by** (*auto simp: K-def fun-eq-iff indicator-def mult-ac*)

have *Kz: (λu. K y u) = (λu. 0)* **if** $\neg 0 < y$ **for** *y* **using** *that*

by (*auto simp: K-def fun-eq-iff*)

have *Kz2: (λy. K y u) = (λy. 0)* **if** $\neg 0 < u$ **for** *u* **using** *that*

```

    by (auto simp: K-def fun-eq-iff)
  have Ki: integrable lborel ( $\lambda u. K y u$ ) for y
  proof (cases  $0 < y$ )
    case True
      have set-integrable lborel  $\{0 < ..\}$  ( $\lambda u. \exp(- (c * y)) * (\exp(- (a * u)) * \sin$ 
        ( $y * u$ ))
        by (intro set-integrable-mult-right laplace-sin-integrable[OF a])
      thus ?thesis unfolding Kslice[OF True] set-integrable-def .
    qed (simp add: Kz)
  have Kn: ( $\int u. \text{norm } (K y u) \partial \text{lborel}$ )  $\leq \exp(- (c * y)) * (1 / a)$ 
    if y:  $0 < y$  for y
  proof -
    have ( $\int u. \text{norm } (K y u) \partial \text{lborel}$ )  $\leq$ 
      ( $\int u. \text{indicator } \{0 < ..\} u *_R (\exp(- (c * y)) * \exp(- (a * u))) \partial \text{lborel}$ )
    proof (rule integral-mono)
      show integrable lborel ( $\lambda u. \text{norm } (K y u)$ ) by (rule integrable-norm[OF Ki])
      show integrable lborel ( $\lambda u. \text{indicator } \{0 < ..\} u *_R (\exp(- (c * y)) * \exp(- (a * u)))$ )
        using set-integrable-mult-right[OF exp-integrable[OF a], of exp(- (c * y))]
        unfolding set-integrable-def .
      fix u :: real
      show  $\text{norm } (K y u) \leq \text{indicator } \{0 < ..\} u *_R (\exp(- (c * y)) * \exp(- (a * u)))$ 
        using y by (auto simp: K-def abs-mult indicator-def intro!: mult-left-le)
    qed
  also have ... = set-lebesgue-integral lborel  $\{0 < ..\}$  ( $\lambda u. \exp(- (c * y)) * \exp(- (a * u))$ )
    unfolding set-lebesgue-integral-def ..
  also have ... =  $\exp(- (c * y)) * \text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda u. \exp(- (a * u)))$ 
    by (rule set-integral-mult-right)
  also have  $\text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda u. \exp(- (a * u))) = 1 / a$ 
    using gam[OF a, of 0] by simp
  finally show ?thesis .
qed
have Kint: integrable ( $\text{lborel} \otimes_M \text{lborel}$ ) ( $\lambda(y, u). K y u$ )
proof (rule lborel-pair.Fubini-integrable[OF Km])
  show AE y in lborel. integrable lborel ( $\lambda u. \text{case } (y, u) \text{ of } (y, u) \Rightarrow K y u$ ) using Ki by simp
  show integrable lborel ( $\lambda y. \int u. \text{norm } (\text{case } (y, u) \text{ of } (y, u) \Rightarrow K y u) \partial \text{lborel}$ )
    proof (rule Bochner-Integration.integrable-bound)
      show integrable lborel ( $\lambda y. \text{indicator } \{0 < ..\} y *_R (\exp(- (c * y)) * (1 / a))$ )
        using set-integrable-mult-left[OF exp-integrable[OF c], of 1 / a]
        unfolding set-integrable-def .
      show ( $\lambda y. \int u. \text{norm } (\text{case } (y, u) \text{ of } (y, u) \Rightarrow K y u)$ )

```



```

    u)  $\Rightarrow$   $K \ y \ u \ \partial \text{lborel}$ )  $\in$  borel-measurable lborel
  using Km by measurable
show AE y in lborel. norm ( $\int u. \text{norm} \ (\text{case} \ (y, u) \ \text{of} \ (y,$ 
  u)  $\Rightarrow$   $K \ y \ u \ \partial \text{lborel}) \leq$ 
  norm (indicator {0<..}  $y *_{\mathbb{R}}$  ( $\exp \ (- \ (c * y)) * (1 / a)$ )))
proof (rule AE-I2)
  fix  $y :: \text{real}$ 
  show norm ( $\int u. \text{norm} \ (\text{case} \ (y, u) \ \text{of} \ (y,$ 
    u)  $\Rightarrow$   $K \ y \ u \ \partial \text{lborel}) \leq$ 
    norm (indicator {0<..}  $y *_{\mathbb{R}}$  ( $\exp \ (- \ (c * y)) * (1 / a)$ )))
  proof (cases  $0 < y$ )
    case True
    have  $0 \leq (\int u. \text{norm} \ (K \ y \ u \ \partial \text{lborel})$ 
      by (rule integral-nonneg-AE) simp
    thus ?thesis using Kn[OF True] True a by simp
  qed (simp add: Kz)
qed
qed
qed
have L:  $\Psi \ a \ c = (\int y. (\int u. K \ y \ u \ \partial \text{lborel}) \ \partial \text{lborel})$ 
proof -
  have ( $\lambda y. \text{indicator} \ \{0 < ..\} \ y *_{\mathbb{R}} \ (\exp \ (- \ (c * y)) * (y / (y^2 + a^2)))$ ) =
    ( $\lambda y. \int u. K \ y \ u \ \partial \text{lborel}$ )
  proof (rule ext)
    fix  $y :: \text{real}$ 
    show indicator {0<..}  $y *_{\mathbb{R}}$  ( $\exp \ (- \ (c * y)) * (y / (y^2 + a^2))$ ) =
      ( $\int u. K \ y \ u \ \partial \text{lborel}$ )
    proof (cases  $0 < y$ )
      case True
      have ( $\int u. K \ y \ u \ \partial \text{lborel}$ ) =
        set-lebesgue-integral lborel {0<..} ( $\lambda u. \exp \ (- \ (c * y)) * (\exp \ (- \ (a * u)) * \sin \ (y * u))$ )
      unfolding Kslice[OF True] set-lebesgue-integral-def ..
      also have ... =  $\exp \ (- \ (c * y)) * (y / (y^2 + a^2))$ 
      unfolding set-integral-mult-right laplace-sin[OF a] ..
      finally have ( $\int u. K \ y \ u \ \partial \text{lborel}$ ) =
         $\exp \ (- \ (c * y)) * (y / (y^2 + a^2))$  .
      thus ?thesis using True by simp
    next
      case False
      have ( $\int u. K \ y \ u \ \partial \text{lborel}$ ) = 0 unfolding Kz[OF False]
      by (rule integral-zero)
      thus ?thesis using False by simp
    qed
  qed
  thus ?thesis unfolding Psi-def set-lebesgue-integral-def by simp
qed
have R:  $\Psi \ c \ a = (\int u. (\int y. K \ y \ u \ \partial \text{lborel}) \ \partial \text{lborel})$ 
proof -

```

```

have (λu. indicator {0<..} u *R (exp (- (a * u)) * (u / (u ^ 2 + c ^ 2)))) =
  (λu. ∫ y. K y u ∂lborel)
proof (rule ext)
  fix u :: real
  show indicator {0<..} u *R (exp (- (a * u)) * (u / (u ^ 2 + c ^ 2))) =
    (∫ y. K y u ∂lborel)
  proof (cases 0 < u)
    case True
      have (∫ y. K y u ∂lborel) =
        set-lebesgue-integral lborel {0<..} (λy. exp (- (a * u)) * (exp (- (c * y)) *
          sin (u * y)))
        unfolding Kslice2[OF True] set-lebesgue-integral-def ..
      also have ... = exp (- (a * u)) * (u / (u ^ 2 + c ^ 2))
        unfolding set-integral-mult-right laplace-sin[OF c] ..
      finally have (∫ y. K y u ∂lborel) =
        exp (- (a * u)) * (u / (u ^ 2 + c ^ 2)) .
      thus ?thesis using True by simp
    case False
      have (∫ y. K y u ∂lborel) = 0 unfolding Kz2[OF False]
        by (rule integral-zero)
      thus ?thesis using False by simp
  qed
qed
thus ?thesis unfolding Psi-def set-lebesgue-integral-def by simp
qed
show ?thesis unfolding L R using lborel-pair.Fubini-integral[OF Kint] by simp
qed

```

$$\Phi(a, c) = \int -0^\infty y^5 e^{\{-cy\}} / (y^2 + a^2) dy.$$

definition *Phi* :: real ⇒ real ⇒ real **where**

$$Phi\ a\ c = set-lebesgue-integral\ lborel\ \{0<..\}\ (\lambda y. y^5 * exp\ (-\ (c * y)) / (y^2 + a^2))$$

lemma *phi-split*:

$$\begin{aligned} &\text{fixes } a\ y :: \text{real} \text{ assumes } a: 0 < a \\ &\text{shows } y^5 * exp\ (-\ (c * y)) / (y^2 + a^2) = \\ &\quad y^3 * exp\ (-\ (c * y)) - a^2 * (y^1 * exp\ (-\ (c * y))) + a^4 * (exp\ \\ &\quad (-\ (c * y)) * \\ &\quad (y / (y^2 + a^2))) \end{aligned}$$

proof –

$$\begin{aligned} &\text{have } D: y^2 + a^2 \neq 0 \text{ using } a \text{ by (simp add: add-nonneg-pos)} \\ &\text{have } y^5 = (y^3 - a^2 * y) * (y^2 + a^2) + a^4 * y \\ &\quad \text{by (simp add: algebra-simps eval-nat-numeral)} \\ &\text{hence } y^5 * exp\ (-\ (c * y)) / (y^2 + a^2) = \\ &\quad ((y^3 - a^2 * y) * (y^2 + a^2) + a^4 * y) * exp\ (-\ (c * y)) / (y^2 + a^2) \\ &\quad \text{by simp} \end{aligned}$$

```

also have ... = (y ^ 3 - a ^ 2 * y) * exp (- (c * y)) + a ^ 4 * (exp (- (c *
y)) * (y /
(y ^ 2 + a ^ 2)))
using D by (simp add: field-simps)
finally show ?thesis by (simp add: algebra-simps)
qed

lemma phi-integrable: 0 < (a::real) ==> 0 < c ==>
set-integrable lborel {0<..} (λy. y ^ 5 * exp (- (c * y)) / (y ^ 2 + a ^ 2))
proof -
assume a: 0 < a and c: 0 < c
have E: (λy. y ^ 5 * exp (- (c * y)) / (y ^ 2 + a ^ 2)) =
(λy. y ^ 3 * exp (- (c * y)) - a ^ 2 * (y ^ 1 * exp (- (c * y)))) + a ^ 4
*
(exp (- (c * y)) * (y / (y ^ 2 + a ^ 2))))
by (rule ext) (rule phi-split[OF a])
show ?thesis unfolding E
by (intro set-integral-add(1) set-integral-diff(1) set-integrable-mult-right gam-integrable
psi-integrable a c)
qed

lemma phi-eq: 0 < a ==> 0 < c ==>
Phi a c = 6 / c ^ 4 - a ^ 2 / c ^ 2 + a ^ 4 * Psi a c
proof -
assume a: 0 < a and c: 0 < c
have E: (λy. y ^ 5 * exp (- (c * y)) / (y ^ 2 + a ^ 2)) =
(λy. y ^ 3 * exp (- (c * y)) - a ^ 2 * (y ^ 1 * exp (- (c * y)))) + a ^ 4
*
(exp (- (c * y)) * (y / (y ^ 2 + a ^ 2))))
by (rule ext) (rule phi-split[OF a])
have i3: set-integrable lborel {0<..} (λy. y ^ 3 * exp (- (c * y)))
by (rule gam-integrable[OF c])
have i1: set-integrable lborel {0<..} (λy. a ^ 2 * (y ^ 1 * exp (- (c * y))))
by (intro set-integrable-mult-right gam-integrable[OF c])
have ip: set-integrable lborel {0<..} (λy. a ^ 4 * (exp (- (c * y)) * (y / (y ^ 2
+
a ^ 2))))
by (intro set-integrable-mult-right psi-integrable[OF a c])
have Phi a c = set-lebesgue-integral lborel {0<..} (λy. y ^ 3 * exp (- (c * y)))
- a ^ 2 * set-lebesgue-integral lborel {0<..} (λy. y ^ 1 * exp (- (c * y))) +
a ^ 4 *
Psi a c
unfolding Phi-def E set-integral-add(2)[OF set-integral-diff(1)[OF i3 i1] ip]
set-integral-diff(2)[OF i3 i1]
set-integral-mult-right Psi-def ..
also have ... = 6 / c ^ 4 - a ^ 2 / c ^ 2 + a ^ 4 * Psi a c
unfolding gam[OF c] by (simp add: fact-numeral power2-eq-square)
finally show ?thesis .
qed

```

lemma *phi-symm*: $0 < a \implies 0 < c \implies c^4 * \text{Phi } a \ c = a^4 * \text{Phi } c \ a$
proof –
 assume *a*: $0 < a$ and *c*: $0 < c$
 have $c^4 * \text{Phi } a \ c = 6 - a^2 * c^2 + a^4 * c^4 * \text{Psi } a \ c$
 unfolding *phi-eq*[*OF a c*] using *c* by (*simp add: field-simps eval-nat-numeral*)
 also have $\dots = a^4 * \text{Phi } c \ a$
 unfolding *phi-eq*[*OF c a*] *psi-symm*[*OF a c*] using *a* by (*simp add: field-simps eval-nat-numeral*)
 finally show ?thesis .
qed

8.3 The pole integral

lemma *geom-pt*: $0 < u \implies (\lambda n. u^4 * \exp(-(a + (\text{real } n + 1)) * u)) \text{ sums } (u^4 * \exp(-(a * u)) / (\exp u - 1))$
proof –
 assume *u*: $0 < u$
 define *x* where $x = \exp(-u)$
 have *x0*: $0 < x < 1$ using *u* by (*simp-all add: x-def*)
 have *g*: $(\lambda n. x^n) \text{ sums } (1 / (1 - x))$ by (*rule geometric-sums*) (use *x0* in *simp*)
 have *s*: $(\lambda n. (u^4 * \exp(-(a * u)) * x) * x^n) \text{ sums } ((u^4 * \exp(-(a * u)) * x) * (1 / (1 - x)))$
 by (*rule sums-mult*[*OF g*])
 have *e1*: $(\lambda n. (u^4 * \exp(-(a * u)) * x) * x^n) = (\lambda n. u^4 * \exp(-(a + (\text{real } n + 1)) * u))$
proof (*rule ext*)
 fix *n* :: nat
 have $x^n = \exp(\text{real } n * (-u))$ unfolding *x-def* by (*rule exp-of-nat-mult*[*symmetric*])
 hence $(u^4 * \exp(-(a * u)) * x) * x^n = u^4 * (\exp(-(a * u)) * \exp(-u) * \exp(\text{real } n * (-u)))$
 by (*simp add: x-def mult-ac*)
 also have $\exp(-(a * u)) * \exp(-u) * \exp(\text{real } n * (-u)) = \exp(-((a + (\text{real } n + 1)) * u))$
 by (*simp add: exp-add*[*symmetric*] *algebra-simps*)
 finally show $(u^4 * \exp(-(a * u)) * x) * x^n = u^4 * \exp(-((a + (\text{real } n + 1)) * u))$.
qed
 have *e2*: $(u^4 * \exp(-(a * u)) * x) * (1 / (1 - x)) = u^4 * \exp(-(a * u)) / (\exp u - 1)$
proof –
 have $x * (1 / (1 - x)) = 1 / (\exp u - 1)$
 unfolding *x-def exp-minus* using *u* by (*simp add: field-simps*)
 moreover have $(u^4 * \exp(-(a * u)) * x) * (1 / (1 - x)) = u^4 * \exp(-(a * u)) * (x * (1 / (1 - x)))$
 by (*simp only: mult.assoc*)
 ultimately show ?thesis by *simp*

```

qed
show ?thesis using s unfolding e1 e2 .
qed

lemma u-integral:
  assumes  $0 < a$  ( $\lambda n. 1 / (a + (\text{real } n + 1)) \wedge 5$ ) sums  $Z$ 
  shows set-integrable lborel  $\{0 < ..\}$  ( $\lambda u. u \wedge 4 * \exp(- (a * u)) * (1 / 4 + 1 /$ 
 $(2 * (exp\ u - 1)) - 1 / (2 * u))) \wedge$ 
 $set\text{-lebesgue-integral lborel } \{0 < ..\} (\lambda u. u \wedge 4 * \exp(- (a * u)) * (1 / 4 +$ 
 $1 / (2 * (exp\ u - 1)) - 1 / (2 * u)))$ 
 $= 6 / a \wedge 5 - 3 / a \wedge 4 + 12 * Z$ 
proof -
  note  $a = \text{assms}(1)$  and  $Z = \text{assms}(2)$ 
  define  $f$  where  $f = (\lambda n\ u. u \wedge 4 * \exp(- ((a + (\text{real } n + 1)) * u)))$ 
  have  $an: 0 < a + (\text{real } n + 1)$  for  $n$  using  $a$  by simp
  have  $fi: \text{set-integrable lborel } \{0 < ..\} (f\ n)$  for  $n$  unfolding  $f\text{-def}$ 
  by (rule gam-integrable[OF  $an$ ])
  have  $fv: \text{set-lebesgue-integral lborel } \{0 < ..\} (f\ n) = 24 * (1 / (a + (\text{real } n + 1))$ 
 $\wedge 5)$  for  $n$ 
  unfolding  $f\text{-def gam}[OF  $an$ ] by (simp add: fact-numeral)
  have  $fnn: 0 \leq f\ n\ u$  if  $0 < u$  for  $n\ u$  using that by (simp add:  $f\text{-def}$ )
  have  $fs: \text{summable } (\lambda n. f\ n\ u)$  if  $0 < u$  for  $u$ 
  using geom-pt[OF that, of  $a$ ] unfolding  $f\text{-def}$  by (rule sums-summable)
  have  $fsum: (\lambda n. \text{set-lebesgue-integral lborel } \{0 < ..\} (f\ n)) \text{ sums } (24 * Z)$ 
  unfolding  $fv$  by (rule sums-mult[OF  $Z$ ])
  have  $IG: \text{set-integrable lborel } \{0 < ..\} (\lambda u. \sum n. f\ n\ u) \wedge$ 
 $\text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda u. \sum n. f\ n\ u) =$ 
 $(\sum n. \text{set-lebesgue-integral lborel } \{0 < ..\} (f\ n))$ 
  by (rule integral-suminf-nn[OF  $fi\ fnn\ fs\ \text{sums-summable}[OF\ fsum]$ ])
  have  $G: (\sum n. f\ n\ u) = u \wedge 4 * \exp(- (a * u)) / (exp\ u - 1)$  if  $0 < u$  for  $u$ 
  using sums-unique[OF geom-pt[OF that, of  $a$ ]] unfolding  $f\text{-def}$  by simp
  have  $GI: \text{set-integrable lborel } \{0 < ..\} (\lambda u. u \wedge 4 * \exp(- (a * u)) / (exp\ u -$ 
 $1))$ 
  by (rule set-integrable-cong-Ioi[OF conjunct1[OF  $IG$ ]]) (rule  $G$ )
  have  $GV: \text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda u. u \wedge 4 * \exp(- (a * u)) / (exp\ u -$ 
 $1)) = 24 * Z$ 
  proof -
    have  $\text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda u. u \wedge 4 * \exp(- (a * u)) / (exp\ u -$ 
 $1)) =$ 
 $\text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda u. \sum n. f\ n\ u)$ 
    by (rule set-integral-cong-Ioi) (rule  $G$ [symmetric])
    also have  $\dots = 24 * Z$  using conjunct2[OF  $IG$ ] sums-unique[OF  $fsum$ ] by
    simp
  finally show ?thesis .
qed
have  $i4: \text{set-integrable lborel } \{0 < ..\} (\lambda u. 1 / 4 * (u \wedge 4 * \exp(- (a * u))))$$ 
```

```

    by (intro set-integrable-mult-right gam-integrable[OF a])
    have iG: set-integrable lborel {0<..} (λu. 1 / 2 * (u ^ 4 * exp (- (a * u)) /
    (exp u -
    1)))
    by (intro set-integrable-mult-right GI)
    have i3: set-integrable lborel {0<..} (λu. 1 / 2 * (u ^ 3 * exp (- (a * u))))
    by (intro set-integrable-mult-right gam-integrable[OF a])
    have H: u ^ 4 * exp (- (a * u)) * (1 / 4 + 1 / (2 * (exp u - 1)) - 1 / (2 *
    u)) =
      1 / 4 * (u ^ 4 * exp (- (a * u))) + 1 / 2 * (u ^ 4 * exp (- (a * u)) /
    (exp u - 1)) - 1 /
      2 * (u ^ 3 * exp (- (a * u)))
    if u: 0 < u for u
  proof -
    have exp u - 1 ≠ 0 using u by simp
    thus ?thesis using u by (simp add: field-simps eval-nat-numeral)
  qed
  have HI: set-integrable lborel {0<..} (λu. 1 / 4 * (u ^ 4 * exp (- (a * u))) +
  1 / 2 *
    (u ^ 4 * exp (- (a * u)) / (exp u - 1)) - 1 / 2 * (u ^ 3 * exp (- (a * u))))
  by (intro set-integral-diff(1) set-integral-add(1) i4 iG i3)
  have HV: set-lebesgue-integral lborel {0<..} (λu. 1 / 4 * (u ^ 4 * exp (- (a *
  u))) + 1 /
    2 * (u ^ 4 * exp (- (a * u)) / (exp u - 1)) - 1 / 2 * (u ^ 3 * exp (- (a *
  u))))
  = 6 / a ^ 5 - 3 / a ^ 4 + 12 * Z
  unfolding set-integral-diff(2)[OF set-integral-add(1)[OF i4 iG] i3] set-integral-add(2)[OF
  i4
    iG] set-integral-mult-right
    GV gam[OF a] by (simp add: fact-numeral)
  show ?thesis
  proof
    show set-integrable lborel {0<..} (λu. u ^ 4 * exp (- (a * u)) * (1 / 4 + 1 /
    (2 *
    (exp u - 1)) - 1 / (2 * u)))
    by (rule set-integrable-cong-Ioi[OF HI]) (rule H[symmetric])
    have set-lebesgue-integral lborel {0<..} (λu. u ^ 4 * exp (- (a * u)) * (1 / 4
    + 1 /
    (2 * (exp u - 1)) - 1 / (2 * u))) =
      set-lebesgue-integral lborel {0<..} (λu. 1 / 4 * (u ^ 4 * exp (- (a * u)))
    + 1 /
      2 * (u ^ 4 * exp (- (a * u)) / (exp u - 1)) - 1 / 2 * (u ^ 3 * exp (-
    (a * u))))
    by (rule set-integral-cong-Ioi) (rule H)
    thus set-lebesgue-integral lborel {0<..} (λu. u ^ 4 * exp (- (a * u)) * (1 / 4
    + 1 /
    (2 * (exp u - 1)) - 1 / (2 * u))) = 6 / a ^ 5 - 3 / a ^ 4 + 12 * Z
    using HV by simp
  qed

```

qed

end

theory *Interval-Integral-Substitutions*
imports *HOL-Analysis.Analysis*
begin

9 Substitutions in interval integrals with finite endpoints

Affine changes of variable and reflections for *interval-lebesgue-integral* with explicit endpoints (HOL-Analysis has the whole-line *lborel-integral-real-affine* and the monotone-substitution rules), restriction of integrability to a subinterval, joining of adjacent intervals, and the invariance $\int_0^{2\pi} g(x - \varphi) dx = \int_0^{2\pi} g$ of a 2π -periodic integrand.

Affine changes of variable in interval integrals with finite endpoints.

lemma *LBINT-affine-ind*:

fixes $a\ b\ c\ t\ x :: \text{real}$
assumes $c: 0 < c$
shows $\text{indicator } \{a <..<b\} (t + c * x) = (\text{indicator } \{(a - t) / c <..<(b - t) / c\} x :: \text{real})$
proof –
have $(t + c * x \in \{a <..<b\}) = (x \in \{(a - t) / c <..<(b - t) / c\})$ **using** c
by (*simp add: field-simps*)
thus *?thesis*
by (*simp add: indicator-def*)
qed

lemma *LBINT-reflect-ind*:

fixes $a\ b\ t\ x :: \text{real}$
shows $\text{indicator } \{a <..<b\} (t + (-1) * x) = (\text{indicator } \{t - b <..<t - a\} x :: \text{real})$
proof –
have $(t + (-1) * x \in \{a <..<b\}) = (x \in \{t - b <..<t - a\})$
by *auto*
thus *?thesis*
by (*simp add: indicator-def*)
qed

lemma *LBINT-reflect*:

fixes $f :: \text{real} \Rightarrow \text{real}$ **and** $a\ b\ t :: \text{real}$
assumes $ab: a \leq b$
shows $\text{interval-lebesgue-integral lborel } (\text{ereal } a) (\text{ereal } b) f$
 $= \text{interval-lebesgue-integral lborel } (\text{ereal } (t - b)) (\text{ereal } (t - a)) (\lambda x. f (t - x))$

proof –

```

have ab':  $t - b \leq t - a$  using ab by simp
have e:  $(\lambda x. \text{indicator } \{a <..< b\} (t + (-1) * x) * f (t + (-1) * x)) =$ 
 $(\lambda x. \text{indicator } \{t - b <..< t - a\} x * f (t - x))$ 
using LBINT-reflect-ind
by simp
have e1:  $\text{interval-lebesgue-integral lborel } (\text{ereal } a) (\text{ereal } b) f =$ 
 $(\int x. \text{indicator } \{a <..< b\} x * f x \partial \text{lborel})$ 
using ab
by (simp add: interval-lebesgue-integral-def set-lebesgue-integral-def)
have e2:  $(\int x. \text{indicator } \{a <..< b\} x * f x \partial \text{lborel}) =$ 
 $(\int x. \text{indicator } \{a <..< b\} (t + (-1) * x) * f (t + (-1) * x) \partial \text{lborel})$ 
using lborel-integral-real-affine[of - 1  $\lambda x. \text{indicator } \{a <..< b\} x * f x t]$ 
by simp
have e4:  $(\int x. \text{indicator } \{t - b <..< t - a\} x * f (t - x) \partial \text{lborel}) =$ 
 $\text{interval-lebesgue-integral lborel } (\text{ereal } (t - b)) (\text{ereal } (t - a)) (\lambda x. f (t - x))$ 
using ab'
by (simp add: interval-lebesgue-integral-def set-lebesgue-integral-def)
show ?thesis using e1 e2 e4 unfolding e
by presburger

```

qed

lemma LBINT-reflect-int:

```

fixes f ::  $\text{real} \Rightarrow \text{real}$  and a b t ::  $\text{real}$ 
assumes ab:  $a \leq b$ 
shows  $\text{interval-lebesgue-integrable lborel } (\text{ereal } a) (\text{ereal } b) f$ 
 $\longleftrightarrow \text{interval-lebesgue-integrable lborel } (\text{ereal } (t - b)) (\text{ereal } (t -$ 
 $a)) (\lambda x. f (t - x))$ 

```

proof –

```

have ab':  $t - b \leq t - a$  using ab by simp
have e:  $(\lambda x. \text{indicator } \{a <..< b\} (t + (-1) * x) * f (t + (-1) * x)) =$ 
 $(\lambda x. \text{indicator } \{t - b <..< t - a\} x * f (t - x))$ 
using LBINT-reflect-ind
by simp
have e1:  $\text{interval-lebesgue-integrable lborel } (\text{ereal } a) (\text{ereal } b) f \longleftrightarrow$ 
 $\text{integrable lborel } (\lambda x. \text{indicator } \{a <..< b\} x * f x)$ 
using ab
by (simp add: interval-lebesgue-integrable-def set-integrable-def)
have e2:  $\text{integrable lborel } (\lambda x. \text{indicator } \{a <..< b\} x * f x) \longleftrightarrow$ 
 $\text{integrable lborel } (\lambda x. \text{indicator } \{a <..< b\} (t + (-1) * x) * f (t + (-1) * x))$ 
using lborel-integrable-real-affine-iff[of - 1  $\lambda x. \text{indicator } \{a <..< b\} x * f x t]$ 
by argo
have e4:  $\text{integrable lborel } (\lambda x. \text{indicator } \{t - b <..< t - a\} x * f (t -$ 
 $x)) \longleftrightarrow \text{interval-lebesgue-integrable lborel } (\text{ereal } (t - b)) (\text{ereal } (t -$ 
 $a)) (\lambda x. f (t - x))$ 
using ab'
by (simp add: interval-lebesgue-integrable-def set-integrable-def)
show ?thesis using e1 e2 e4 unfolding e
by meson

```


qed

lemma *LBINT-affine-int:*

fixes $f :: \text{real} \Rightarrow \text{real}$ **and** $a\ b\ c\ t :: \text{real}$

assumes $c: 0 < c$ **and** $ab: a \leq b$

shows $\text{interval-lebesgue-integrable lborel (ereal a) (ereal b) f}$

$\longleftrightarrow \text{interval-lebesgue-integrable lborel (ereal ((a - t) / c)) (ereal ((b - t) / c)) (\lambda x. f (t + c * x))$

proof –

have $ab': (a - t) / c \leq (b - t) / c$ **using** $ab\ c$

by (*fastforce intro: divide-right-mono*)

have $\text{interval-lebesgue-integrable lborel (ereal a) (ereal b) f} \longleftrightarrow$

$\text{integrable lborel } (\lambda x. \text{indicator } \{a <..<b\} x * f x)$

using ab

by (*simp add: interval-lebesgue-integrable-def set-integrable-def*)

also have $\dots \longleftrightarrow$

$\text{integrable lborel } (\lambda x. \text{indicator } \{a <..<b\} (t + c * x) * f (t + c * x))$

using $\text{lborel-integrable-real-affine-iff[of } c\ \lambda x. \text{indicator } \{a <..<b\} x * f x\ t]\ c$

by *fastforce*

also have $\dots \longleftrightarrow$

$\text{integrable lborel } (\lambda x. \text{indicator } \{(a - t) / c <..<(b - t) / c\} x * f (t + c * x))$

using $\text{LBINT-affine-ind[OF } c]$

by *presburger*

also have $\dots \longleftrightarrow$

$\text{interval-lebesgue-integrable lborel (ereal ((a - t) / c)) (ereal ((b - t) / c)) (\lambda x. f (t + c * x))$

using ab'

by (*auto simp: interval-lebesgue-integrable-def set-integrable-def*)

finally show *?thesis* .

qed

lemma *LBINT-affine:*

fixes $f :: \text{real} \Rightarrow \text{real}$ **and** $a\ b\ c\ t :: \text{real}$

assumes $c: 0 < c$ **and** $ab: a \leq b$

shows $\text{interval-lebesgue-integral lborel (ereal a) (ereal b) f}$

$= c * \text{interval-lebesgue-integral lborel (ereal ((a - t) / c)) (ereal ((b - t) / c)) (\lambda x. f (t + c * x))$

proof –

have $ab': (a - t) / c \leq (b - t) / c$ **using** $ab\ c$

by (*auto intro: divide-right-mono*)

have $\text{interval-lebesgue-integral lborel (ereal a) (ereal b) f} =$

$(\int x. \text{indicator } \{a <..<b\} x * f x\ \partial \text{lborel})$

using ab

by (*simp add: interval-lebesgue-integral-def set-lebesgue-integral-def*)

also have $\dots = c * (\int x. \text{indicator } \{a <..<b\} (t + c * x) * f (t + c * x)\ \partial \text{lborel})$

using $\text{lborel-integral-real-affine[of } c\ \lambda x. \text{indicator } \{a <..<b\} x * f x\ t]\ c$

by *simp*

also have $(\int x. \text{indicator } \{a <..<b\} (t + c * x) * f (t + c * x)\ \partial \text{lborel}) =$

```

  (∫ x. indicator {(a - t) / c <..< (b - t) / c} x * f (t + c * x) ∂lborel)
  using LBINT-affine-ind[OF c]
  by simp
  also have ... = interval-lebesgue-integral lborel (ereal ((a - t) / c)) (ereal ((b
- t) /
  c)) (λx. f (t + c * x))
  using ab'
  by (simp add: interval-lebesgue-integral-def set-lebesgue-integral-def)
  finally show ?thesis .
qed

```

lemma *interval-integrable-sub*:

```

  fixes f :: real ⇒ real and a b c d :: real
  assumes a ≤ c c ≤ d d ≤
    b interval-lebesgue-integrable lborel (ereal a) (ereal b) f
  shows interval-lebesgue-integrable lborel (ereal c) (ereal d) f
proof -
  have s: set-integrable lborel {a<..b} f using assms unfolding interval-lebesgue-integrable-def
  by simp
  have set-integrable lborel {c<..d} f using set-integrable-subset[OF s, of {c<..d}]
  assms
  by fastforce
  thus ?thesis using assms unfolding interval-lebesgue-integrable-def
  by simp
qed

```

lemma *interval-integrable-join*:

```

  fixes f :: real ⇒ real and a b c :: real
  assumes ab: a ≤ b and bc: b ≤ c
    and i1: interval-lebesgue-integrable lborel (ereal a) (ereal b) f
    and i2: interval-lebesgue-integrable lborel (ereal b) (ereal c) f
  shows interval-lebesgue-integrable lborel (ereal a) (ereal c) f
proof -
  have s1: set-integrable lborel {a<..b} f using i1 ab
  unfolding interval-lebesgue-integrable-def by simp
  have s2: set-integrable lborel {b<..c} f using i2 bc
  unfolding interval-lebesgue-integrable-def by simp
  have u: set-integrable lborel ({a<..b} ∪ {b<..c}) f using set-integrable-Un[OF
s1 s2]
  by simp
  have set-integrable lborel {a<..c} f
    using set-integrable-discrete-difference[where X={b} and A={a<..b} ∪
{b<..c} and
    B={a<..c} and M=lborel and f=f] u ab bc
  by fastforce
  thus ?thesis using ab bc unfolding interval-lebesgue-integrable-def
  by simp
qed

```

lemma *periodic-shift-fun*:

fixes $g :: \text{real} \Rightarrow \text{real}$ **and** $\varphi :: \text{real}$
assumes $\text{per}: \bigwedge x. g (x + 2 * \pi) = g x$
shows $(\lambda x. g (x - \varphi)) = (\lambda x. g (2 * \pi - \varphi + 1 * x))$
proof
fix x
have $g (x - \varphi) = g (x - \varphi + 2 * \pi)$ **using** $\text{per}[of\ x - \varphi]$ **by** *simp*
thus $g (x - \varphi) = g (2 * \pi - \varphi + 1 * x)$
by (*simp add: algebra-simps*)
qed

lemma *periodic-shift-int*:

fixes $g :: \text{real} \Rightarrow \text{real}$ **and** $\varphi :: \text{real}$
assumes $\text{per}: \bigwedge x. g (x + 2 * \pi) = g x$
and $\text{ig}: \text{interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * \pi)) } g$
and $p0: 0 \leq \varphi \leq 2 * \pi$
shows $\text{interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * \pi)) } (\lambda x. g (x - \varphi))$
proof -
have $a1: \text{interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * \pi - \varphi)) } g$
using $\text{interval-integrable-sub}[OF - - - \text{ig}, of\ 0\ 2 * \pi - \varphi]\ p0$
by *simp*
have $i1: \text{interval-lebesgue-integrable lborel (ereal } \varphi) (ereal (2 * \pi)) (\lambda x. g (x - \varphi))$
using $a1\ \text{LBINT-affine-int}[of\ 1\ 0\ 2 * \pi - \varphi\ g - \varphi]\ p0$
by *simp*
have $a2: \text{interval-lebesgue-integrable lborel (ereal (2 * \pi - \varphi)) (ereal (2 * \pi)) } g$
using $\text{interval-integrable-sub}[OF - - - \text{ig}, of\ 2 * \pi - \varphi\ 2 * \pi]\ p0$
by *simp*
have $i2: \text{interval-lebesgue-integrable lborel (ereal 0) (ereal } \varphi) (\lambda x. g (x - \varphi))$
using $a2\ \text{LBINT-affine-int}[of\ 1\ 2 * \pi - \varphi\ 2 * \pi\ g\ 2 * \pi - \varphi]\ p0$
unfolding *periodic-shift-fun* $[of\ g\ \varphi,\ OF\ \text{per}]$
by *simp*
show *?thesis* **using** $\text{interval-integrable-join}[OF\ p0(1)\ p0(2)\ i2\ i1]$.
qed

lemma *periodic-shift-eq*:

fixes $g :: \text{real} \Rightarrow \text{real}$ **and** $\varphi :: \text{real}$
assumes $\text{per}: \bigwedge x. g (x + 2 * \pi) = g x$
and $\text{ig}: \text{interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * \pi)) } g$
and $p0: 0 \leq \varphi \leq 2 * \pi$
shows $\text{interval-lebesgue-integral lborel (ereal (0)) (ereal (2 * \pi)) } (\lambda x. g (x - \varphi)) = \text{interval-lebesgue-integral lborel (ereal (0)) (ereal (2 * \pi)) } g$
proof -
have $\text{interval-lebesgue-integral lborel (ereal (0)) (ereal } \varphi) (\lambda x. g (x - \varphi)) + \text{interval-lebesgue-integral lborel (ereal } \varphi) (ereal (2 * \pi - \varphi)) (\lambda x. g (x - \varphi)) = \text{interval-lebesgue-integral lborel (ereal 0) (ereal (2 * \pi)) } g$
using $\text{interval-integral-add}[of\ \text{interval-lebesgue-integral lborel (ereal 0) (ereal } \varphi) (\lambda x. g (x - \varphi))\ \text{interval-lebesgue-integral lborel (ereal } \varphi) (ereal (2 * \pi - \varphi)) (\lambda x. g (x - \varphi))\ \text{interval-lebesgue-integral lborel (ereal 0) (ereal (2 * \pi)) } g]\ p0$
by *simp*
show *?thesis* **using** $\text{interval-integral-add}[of\ \text{interval-lebesgue-integral lborel (ereal 0) (ereal } \varphi) (\lambda x. g (x - \varphi))\ \text{interval-lebesgue-integral lborel (ereal } \varphi) (ereal (2 * \pi - \varphi)) (\lambda x. g (x - \varphi))\ \text{interval-lebesgue-integral lborel (ereal 0) (ereal (2 * \pi)) } g]\ p0$
by *simp*
qed

```

    pi)) (λx. g (x - φ)) = interval-lebesgue-integral lborel (ereal (0)) (ereal (2 *
    pi)) (λx. g (x - φ))
  using interval-integral-sum[of ereal 0 ereal φ ereal (2 * pi) λx. g (x -
    φ)] periodic-shift-int[OF per ig p0] p0
  by simp
  moreover have interval-lebesgue-integral lborel (ereal (φ)) (ereal (2 *
    pi)) (λx. g (x - φ)) = interval-lebesgue-integral lborel (ereal (0)) (ereal (2 *
    pi - φ)) g
  using LBINT-affine[of 1 0 2 * pi - φ g - φ] p0
  by simp
  moreover have interval-lebesgue-integral lborel (ereal (0)) (ereal (φ)) (λx. g (x
    - φ)) = interval-lebesgue-integral lborel (ereal (2 * pi - φ)) (ereal (2 * pi)) g
  using LBINT-affine[of 1 2 * pi - φ 2 * pi g 2 * pi - φ] p0
  unfolding periodic-shift-fun[of g φ, OF per]
  by simp
  moreover have interval-lebesgue-integral lborel (ereal (0)) (ereal (2 * pi - φ))
    g +
    interval-lebesgue-integral lborel (ereal (2 * pi - φ)) (ereal (2 * pi)) g =
    interval-lebesgue-integral lborel (ereal (0)) (ereal (2 * pi)) g
  using interval-integral-sum[of ereal 0 ereal (2 * pi - φ) ereal (2 * pi) g] ig p0
  by simp
  ultimately show ?thesis by linarith
qed

end

```

```

theory Log-Sine-Integral
  imports Interval-Integral-Substitutions HOL-Complex-Analysis.Complex-Analysis
begin

```

10 The log-sine integral and circle averages of $\ln |z - a|$

Jordan's inequality $\sin \vartheta \geq 2\vartheta/\pi$ on $[0, \pi/2]$, integrability of $\ln$ near 0 , the classical $\int_0^\pi \ln(\sin \vartheta) d\vartheta = -\pi \ln 2$, and the circle average $\int_0^{2\pi} \ln |e^{i\vartheta} - a| d\vartheta = 2\pi \ln(\max 1 |a|)$ for every complex a , including the singular case $|a| = 1$; the same for the normalised arclength measure *circ* on a circle of radius ε about c . This is the layer of Jensen's formula that HOL-Complex_Analysis lacks.

Jordan's inequality $\sin \vartheta \geq 2\vartheta/\pi$ on $[0, \pi/2]$: $\sin$ is concave there ($\sin'' = -\sin \leq 0$) and the chord through $(0, 0)$ and $(\pi/2, 1)$ lies below it.

```

lemma sin-ge-two-over-pi:
  assumes 0 ≤ ϑ ≤ pi / 2
  shows 2 * ϑ / pi ≤ sin ϑ
proof -

```

```

have c1: convex {0..pi / 2 :: real}
  by blast
have c2:  $\bigwedge x. \text{DERIV } \sin x :> \cos x$ 
  by (blast intro: DERIV-sin)
have c3:  $\bigwedge x. \text{DERIV } \cos x :> - \sin x$ 
  by (fast intro: DERIV-cos)
have c4:  $\bigwedge x. x \in \{0..pi / 2\} \implies - \sin x \leq 0$ 
  using sin-ge-zero by simp
have conc: concave-on {0..pi / 2} sin
  using f''-le0-imp-concave[OF c1 c2 c3 c4] by fastforce
show ?thesis
  using concave-onD-Icc'[OF conc, of  $\vartheta$ ] assms
  by (simp add: field-simps)
qed

```

$\ln$ is integrable near 0: $\int -0^c (-\ln u) du = c - c \ln c < \infty$.

lemma *ln-integrable-near-zero*:

```

assumes 0 < c
shows set-integrable lborel {0<.. $c$ } ( $\lambda u :: \text{real}. \ln u$ )
proof -
  define F where F u = u * ln c - u * ln u + u for u :: real
  define f where f u = ln c - ln u for u :: real
  have D: DERIV F x :> f x if 0 < x for x
    unfolding F-def[abs-def] f-def using that
    by (intro derivative-eq-intros) auto
  have contf: isCont f x if 0 < x for x
    unfolding f-def using that
    by (fastforce intro: continuous-intros)
  have nonneg: AE x in lborel. ereal 0 < ereal x  $\longrightarrow$ 
    ereal x < ereal c  $\longrightarrow$  0  $\leq$  f x
    unfolding f-def
    by fastforce
  have lim0: ((F  $\circ$  real-of-ereal)  $\longrightarrow$  0) (at-right (ereal 0))
    unfolding ereal-tendsto-simps F-def
    by real-asymp
  have limc: ((F  $\circ$  real-of-ereal)  $\longrightarrow$  F c) (at-left (ereal c))
    unfolding ereal-tendsto-simps F-def using assms
    by (fast intro: tendsto-intros)
  note FTC = interval-integral-FTC-nonneg[of ereal 0 ereal c F f, OF - - - nonneg
    lim0 limc]
  have set-integrable lborel {0<.. $c$ } ( $\lambda u :: \text{real}. \ln u$ )
    using interval-integral-const(1)[of 0 c ln c] assms
    unfolding interval-lebesgue-integrable-def
    by auto
  moreover have set-integrable lborel {0<.. $c$ } f
    using FTC(1) assms D contf
    by auto
  ultimately have set-integrable lborel {0<.. $c$ } ( $\lambda u. \ln c - f u$ )
    by (rule set-integral-diff(1))

```

```

thus ?thesis unfolding f-def
  by simp
qed

```

The classical $\int_0^\pi \log \sin \vartheta \, d\vartheta = -\pi \log 2$; first the integrability.

```

lemma ln-sin-integrable: interval-lebesgue-integrable lborel (ereal 0) (ereal pi)
  (λϑ. ln (sin ϑ))

```

```

proof -

```

```

  have meas: (λϑ :: real. ln (sin ϑ)) ∈ borel-measurable borel

```

```

    by simp

```

```

  have lower: ln (2 * ϑ / pi) ≤ ln (sin ϑ) if ϑ ∈ {0 <..< pi / 2}

```

```

    for ϑ

```

```

    using sin-ge-two-over-pi[of ϑ] sin-gt-zero[of ϑ] pi-gt-zero ln-le-cancel-iff[of 2
      * ϑ / pi sin ϑ]

```

```

    using that

```

```

    by simp

```

```

  have upper: ln (sin ϑ) ≤ 0 if ϑ ∈ {0 <..< pi} for ϑ

```

```

    using sin-le-one[of ϑ] sin-gt-zero[of ϑ]

```

```

    using that

```

```

    by (simp add: ln-le-zero-iff)

```

```

  have bound: norm (ln (sin ϑ)) ≤ norm (ln ϑ) + norm (ln (2 / pi))

```

```

    if th: ϑ ∈ {0 <..< pi / 2} for ϑ

```

```

proof -

```

```

  have e0: 2 * ϑ / pi = ϑ * (2 / pi)

```

```

    by simp

```

```

  have ln (2 * ϑ / pi) = ln ϑ + ln (2 / pi)

```

```

    using th pi-gt-zero ln-mult[of ϑ 2 / pi] unfolding e0

```

```

    by simp

```

```

  moreover have ln (2 * ϑ / pi) ≤ ln (sin ϑ) using lower th by blast

```

```

  moreover have ln (sin ϑ) ≤ 0 using upper th pi-gt-zero by auto

```

```

  ultimately show ?thesis unfolding real-norm-def

```

```

    by linarith

```

```

qed

```

```

  have i1: set-integrable lborel {0 <..< pi / 2} (λϑ. norm (ln ϑ))

```

```

    using set-integrable-abs[OF ln-integrable-near-zero[OF pi-half-gt-zero]] unfold-

```

```

ing real-norm-def

```

```

    by simp

```

```

  have i2: set-integrable lborel {0 <..< pi / 2} (λϑ :: real. norm (ln (2 / pi)))

```

```

    using interval-integral-const(1)[of 0 pi / 2 norm (ln (2 / pi))] pi-gt-zero

```

```

    unfolding interval-lebesgue-integrable-def

```

```

    by simp

```

```

  have int-ln: set-integrable lborel {0 <..< pi / 2} (λϑ. norm (ln ϑ) +

```

```

    norm (ln (2 / pi)))

```

```

    using set-integral-add(1)[OF i1 i2] by simp

```

```

  have sb: set-borel-measurable lborel {0 <..< pi / 2} (λϑ. ln (sin ϑ))

```

```

    unfolding set-borel-measurable-def using meas

```

```

    by simp

```

```

  have ae: AE x in lborel. x ∈ {0 <..< pi / 2} ⟶

```

```

    norm (ln (sin x)) ≤ norm (norm (ln x) + norm (ln (2 / pi))) using bound

```

```

  by (auto intro: AE-I2)
have int-half: set-integrable lborel {0<.. $\pi$ /2} ( $\lambda \vartheta. \ln (\sin \vartheta)$ )
  using set-integrable-bound[OF int-ln sb ae] by simp
have interval-lebesgue-integrable lborel (ereal 0) (ereal ( $\pi$  /
  2)) ( $\lambda \vartheta. \ln (\sin \vartheta)$ ) using int-half
  unfolding interval-lebesgue-integrable-def
  by simp
hence interval-lebesgue-integrable lborel (ereal ( $\pi$  -  $\pi$  / 2)) (ereal ( $\pi$  -
  0)) ( $\lambda x. \ln (\sin (\pi - x))$ )
  using LBINT-reflect-int[of 0  $\pi$  / 2 ( $\lambda \vartheta. \ln (\sin \vartheta)$ )  $\pi$ ]
  by simp
hence int-upper: set-integrable lborel { $\pi$  / 2<.. $\pi$ } ( $\lambda \vartheta. \ln (\sin \vartheta)$ )
  unfolding interval-lebesgue-integrable-def
  by simp
have split: einterval (ereal 0) (ereal  $\pi$ ) =
  {0<.. $\pi$  / 2}  $\cup$  { $\pi$  / 2<.. $\pi$ }  $\cup$  { $\pi$  / 2}
  using pi-gt-zero
  by auto
have m1a: {0<.. $\pi$  / 2}  $\in$  sets lborel
  by simp
have m1b: { $\pi$  / 2<.. $\pi$ }  $\in$  sets lborel
  by simp
note m1 = m1a m1b
have u1: set-integrable lborel ({0<.. $\pi$  / 2}  $\cup$  { $\pi$  /
  2<.. $\pi$ }) ( $\lambda \vartheta. \ln (\sin \vartheta)$ ) using set-integrable-Un[OF int-half int-upper]
  using m1
  by blast
have set-integrable lborel ({0<.. $\pi$  / 2}  $\cup$  { $\pi$  / 2<.. $\pi$ }  $\cup$  { $\pi$  /
  2}) ( $\lambda \vartheta. \ln (\sin \vartheta)$ )
  using set-integrable-discrete-difference[where X={ $\pi$  / 2} and A={0<.. $\pi$ 
  / 2}  $\cup$  { $\pi$  / 2<.. $\pi$ } and B={0<.. $\pi$  / 2}  $\cup$  { $\pi$  / 2<.. $\pi$ }  $\cup$  { $\pi$  /
  2} and M=lborel and f=( $\lambda \vartheta. \ln (\sin \vartheta)$ )] u1
  by auto
moreover have ereal 0  $\leq$  ereal  $\pi$  using pi-gt-zero by simp
ultimately show ?thesis unfolding interval-lebesgue-integrable-def unfolding
split[symmetric]
  by simp
qed

```

lemma *ln-sin-integrable-half*: interval-lebesgue-integrable lborel (ereal 0) (ereal (π / 2)) ($\lambda \vartheta. \ln (\sin \vartheta)$)

proof –

```

have a: set-integrable lborel {0<.. $\pi$ } ( $\lambda \vartheta. \ln (\sin \vartheta)$ )
  using ln-sin-integrable pi-gt-zero unfolding interval-lebesgue-integrable-def
  by simp
have set-integrable lborel {0<.. $\pi$  / 2} ( $\lambda \vartheta. \ln (\sin \vartheta)$ )
  using set-integrable-subset[OF a, of {0<.. $\pi$  / 2}]

```

by fastforce
 thus ?thesis unfolding interval-lebesgue-integrable-def
 by simp
 qed

lemma *ln-cos-integrable-half*: interval-lebesgue-integrable lborel (ereal 0) (ereal (pi / 2)) (λu. ln (cos u))
proof –
 have e: (λx. ln (sin (pi / 2 - x))) = (λx. ln (cos x))
 by (simp add: cos-sin-eq)
 show ?thesis using LBINT-reflect-int[of 0 pi / 2 λϑ. ln (sin ϑ) pi / 2] *ln-sin-integrable-half* unfolding e
 by simp
 qed

lemma *ln-sin-integral*: interval-lebesgue-integral lborel (ereal 0) (ereal pi) (λϑ. ln (sin ϑ)) = - pi * ln 2
proof –
 define I where I = interval-lebesgue-integral lborel (ereal 0) (ereal pi) (λϑ. ln (sin ϑ))
 have interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2)) (λϑ. ln (sin ϑ)) + interval-lebesgue-integral lborel (ereal (pi / 2)) (ereal (pi)) (λϑ. ln (sin ϑ)) = I unfolding I-def
 using interval-integral-sum[of ereal 0 ereal (pi / 2) ereal pi (λϑ. ln (sin ϑ))] *ln-sin-integrable*
 by simp
 moreover have interval-lebesgue-integral lborel (ereal (pi / 2)) (ereal (pi)) (λϑ. ln (sin ϑ)) = interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2)) (λϑ. ln (sin ϑ))
 using LBINT-reflect[of pi / 2 pi (λϑ. ln (sin ϑ)) pi]
 by simp
 ultimately have half: I = 2 * interval-lebesgue-integral lborel (ereal 0) (ereal (pi / 2)) (λϑ. ln (sin ϑ))
 by linarith
 have p: 0 ≤ pi using pi-gt-zero by simp
 have pw: ln (sin (0 + 2 * x)) = ln 2 + ln (sin x) + ln (cos x) if x: 0 < x < pi / 2 for x
proof –
 have sx: 0 < sin x using x sin-gt-zero[of x]
 by simp
 have cx: 0 < cos x using x cos-gt-zero-pi[of x]
 by simp
 have e2: sin (0 + 2 * x) = 2 * (sin x * cos x) using sin-double[of x]
 by simp
 show ?thesis unfolding e2 using sx cx ln-mult[of 2 sin x * cos x] ln-mult[of sin x cos x]

by simp
 qed
 have $I = 2 * \text{interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2))} (\lambda x. \ln (\sin (0 + 2 * x)))$
 using LBINT-affine[of 2 0 pi $\lambda \vartheta. \ln (\sin \vartheta)$ 0, OF - p] unfolding I-def
 by simp
 moreover have $\text{interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2))} (\lambda x. \ln (\sin (0 + 2 * x))) =$
 $\text{interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2))} (\lambda x. \ln 2 + \ln (\sin x) +$
 $\ln (\cos x))$ using pw pi-gt-zero
 by (auto intro: interval-integral-cong)
 moreover have $\text{interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2))} (\lambda x. \ln 2 +$
 $\ln (\sin x) + \ln (\cos x)) = \text{pi} / 2 * \ln 2 +$
 $\text{interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2))} (\lambda u. \ln (\sin u)) +$
 $\text{interval-lebesgue-integral lborel (ereal (0)) (ereal (pi / 2))} (\lambda u. \ln (\cos u))$
 using ln-sin-integrable-half ln-cos-integrable-half
 by simp
 ultimately have doubling: $I = \text{pi} * \ln 2 + 2 * \text{interval-lebesgue-integral lborel (ereal 0) (ereal (pi / 2))} (\lambda u. \ln (\sin u))$
 $+ 2 * \text{interval-lebesgue-integral lborel (ereal 0) (ereal (pi / 2))} (\lambda u. \ln (\cos u))$
 by linarith
 have $e: (\lambda x. \ln (\cos (\text{pi} / 2 - x))) = (\lambda x. \ln (\sin x))$
 by (simp add: cos-diff)
 have cos-eq: $\text{interval-lebesgue-integral lborel (ereal 0) (ereal (pi / 2))} (\lambda u. \ln (\cos u)) = \text{interval-lebesgue-integral lborel (ereal 0) (ereal (pi / 2))} (\lambda u. \ln (\sin u))$
 using LBINT-reflect[of 0 pi / 2 $\lambda u. \ln (\cos u)$ pi / 2] pi-gt-zero unfolding e
 by simp
 show ?thesis using half doubling cos-eq unfolding I-def[symmetric] by linarith
 qed

 lemma norm-1-minus-cis: $\text{norm} (1 - \text{cis } \vartheta) ^ 2 = (2 * \sin (\vartheta / 2)) ^ 2$
 proof -
 have a: $\text{norm} (1 - \text{cis } \vartheta) ^ 2 = (1 - \cos \vartheta) ^ 2 + (\sin \vartheta) ^ 2$
 by (simp add: cmod-power2)
 have b: $\cos \vartheta = 1 - 2 * \sin (\vartheta / 2) ^ 2$ using cos-double-sin[of $\vartheta / 2$]
 by simp
 have c: $\sin \vartheta = 2 * \sin (\vartheta / 2) * \cos (\vartheta / 2)$
 using sin-double[of $\vartheta / 2$]
 by simp
 have d: $\sin (\vartheta / 2) ^ 2 + \cos (\vartheta / 2) ^ 2 = 1$ by (rule sin-cos-squared-add)
 have e: $(1 - (1 - 2 * \sin (\vartheta / 2) ^ 2)) ^ 2 + (2 * \sin (\vartheta / 2) * \cos (\vartheta / 2)) ^ 2 = (2 * \sin (\vartheta / 2)) ^ 2 * (\sin (\vartheta / 2) ^ 2 + \cos (\vartheta / 2) ^ 2)$
 using sin-cos-squared-add[of $\vartheta / 2$]
 unfolding power2-eq-square
 by argo

show ?thesis **unfolding** a b c e d
by simp
qed

lemma *ln-norm-one-minus-cis*:
assumes th: $0 < \vartheta \leq 2 * \pi$
shows $\ln (\text{norm } (1 - \text{cis } \vartheta)) = \ln 2 + \ln (\sin (\vartheta / 2))$
proof –
have s: $0 < \sin (\vartheta / 2)$ **using** th sin-gt-zero[of $\vartheta / 2$]
by simp
have n: $\text{norm } (1 - \text{cis } \vartheta) = 2 * \sin (\vartheta / 2)$ **using** norm-1-minus-cis[of ϑ] s
using power2-eq-iff-nonneg[of $\text{norm } (1 - \text{cis } \vartheta) 2 * \sin (\vartheta / 2)$]
by simp
show ?thesis **unfolding** n **using** s ln-mult[of $2 \sin (\vartheta / 2)$]
by simp
qed

$\int_{-2\pi}^{2\pi} \log|1 - e^{i\vartheta}| d\vartheta = 0$: the singular case of the circle average.

lemma *circle-log-one*: *interval-lebesgue-integral lborel (ereal 0) (ereal (2 * pi))* $(\lambda \vartheta. \ln (\text{norm } (1 - \text{cis } \vartheta))) = 0$
proof –
have c1: *interval-lebesgue-integral lborel (ereal 0) (ereal (2 * pi))* $(\lambda \vartheta. \ln (\text{norm } (1 - \text{cis } \vartheta))) =$
*interval-lebesgue-integral lborel (ereal 0) (ereal (2 * pi))* $(\lambda \vartheta. \ln 2 + \ln (\sin (\vartheta / 2)))$ **using** ln-norm-one-minus-cis pi-gt-zero
by (auto intro: interval-integral-cong)
have p: $0 \leq 2 * \pi$ **using** pi-gt-zero **by** simp
have gi: *interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * pi))* $(\lambda \vartheta. \ln (\sin (\vartheta / 2)))$
using LBINT-affine-int[of $2 0 2 * \pi \lambda \vartheta. \ln (\sin (\vartheta / 2)) 0$, OF - p] ln-sin-integrable
by simp
have gv: *interval-lebesgue-integral lborel (ereal 0) (ereal (2 * pi))* $(\lambda \vartheta. \ln (\sin (\vartheta / 2))) = 2 * (-\pi * \ln 2)$
using LBINT-affine[of $2 0 2 * \pi \lambda \vartheta. \ln (\sin (\vartheta / 2)) 0$, OF - p] ln-sin-integral
by simp
have c2: *interval-lebesgue-integral lborel (ereal 0) (ereal (2 * pi))* $(\lambda \vartheta. \ln 2 + \ln (\sin (\vartheta / 2))) =$
 $\ln 2 * (2 * \pi - 0) + \text{interval-lebesgue-integral lborel (ereal 0) (ereal (2 * pi)) } (\lambda \vartheta. \ln (\sin (\vartheta / 2)))$ **using** gi
by simp
show ?thesis **using** c1 c2 gv
by argo
qed

lemma *circle-log-one-integrable*: *interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * pi))* $(\lambda \vartheta. \ln (\text{norm } (1 - \text{cis } \vartheta)))$

proof –
have $p: 0 \leq 2 * \pi$ **using** *pi-gt-zero* **by** *simp*
have *interval-lebesgue-integrable lborel* (*ereal* 0) (*ereal* (2 *
 π)) ($\lambda \vartheta. \ln (\sin (\vartheta / 2))$)
using *LBINT-affine-int*[of 2 0 2 * π $\lambda \vartheta. \ln (\sin (\vartheta / 2))$] 0,
OF - *p*] *ln-sin-integrable*
by *simp*
hence *interval-lebesgue-integrable lborel* (*ereal* 0) (*ereal* (2 * π)) ($\lambda \vartheta. \ln 2 +$
 $\ln (\sin (\vartheta / 2))$)
by *simp*
hence *s2: set-integrable lborel* {0 <.. $< 2 * \pi$ } ($\lambda \vartheta. \ln 2 + \ln (\sin (\vartheta /$
 $2))$) **using** *pi-gt-zero* **unfolding** *interval-lebesgue-integrable-def*
by *simp*
have *set-integrable lborel* {0 <.. $< 2 * \pi$ } ($\lambda \vartheta. \ln (\text{norm } (1 - \text{cis } \vartheta))$)
using *set-integrable-cong*[of *lborel lborel* {0 <.. $< 2 * \pi$ } {0 <.. $< 2 * \pi$ }
 $\lambda \vartheta. \ln (\text{norm } (1 - \text{cis } \vartheta))$ $\lambda \vartheta. \ln 2 +$
 $\ln (\sin (\vartheta / 2))$] *ln-norm-one-minus-cis s2*
by *simp*
thus *?thesis* **using** *pi-gt-zero* **unfolding** *interval-lebesgue-integrable-def*
by *simp*
qed

Mean value of the holomorphic $\text{Ln } (1 - z/a)$ on the unit circle, $|a| > 1$.

lemma *circle-log-gt*:

assumes $1 < \text{norm } a$
shows *interval-lebesgue-integral lborel* (*ereal* 0) (*ereal* (2 *
 π)) ($\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - a))$) = 2 * π * $\ln (\text{norm } a)$
proof –
define *f* **where** $f = (\lambda z. \text{Ln } (1 - z / a))$
have *a0: a ≠ 0* **using** *assms* **by** *auto*
have *nr: 1 - z / a ∉ ℝ_{≤0}* **if** $\text{norm } z < \text{norm } a$ **for** *z*
proof –
have $\text{norm } (z / a) < 1$ **using** *that a0*
by (*simp add: norm-divide field-simps*)
hence $\text{Re } (z / a) < 1$ **using** *complex-Re-le-cmod*[of z / a] **by** *linarith*
thus *?thesis*
by (*simp add: complex-nonpos-Reals-iff*)
qed
have *hol: f holomorphic-on ball 0 (norm a)* **unfolding** *f-def* **using** *nr a0*
by (*fastforce intro: holomorphic-intros*)
have *sb: cball 0 1 ⊆ ball (0 :: complex) (norm a)* **using** *assms*
by *auto*
have *sb2: ball 0 1 ⊆ ball (0 :: complex) (norm a)* **using** *sb ball-subset-cball*
by *blast*
have *cont: continuous-on (cball 0 1) f* **using** *holomorphic-on-imp-continuous-on*[*OF*
hol] *sb*
by (*blast intro: continuous-on-subset*)
have *cauchy: ((λu. f u / (u - 0)) has-contour-integral (2 * of-real π * i * f 0)) (circlepath 0 1)*

```

using Cauchy-integral-circlepath [OF cont holomorphic-on-subset [OF hol sb2],
of 0]
by simp
have f0:  $f\ 0 = 0$  by (simp add: f-def)
have mean: interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
pi)) ( $\lambda\vartheta. \text{Re } (f\ (\text{cis } \vartheta))$ ) = 0
proof -
have cx: circlepath 0 1  $x = \text{cis } (2 * \text{pi} * x)$  for  $x$ 
by (simp add: circlepath cis-conv-exp mult-ac)
have m1: (( $\lambda x. f\ (\text{circlepath } 0\ 1\ x) / (\text{circlepath } 0\ 1\ x - 0) *$ 
vector-derivative (circlepath 0 1) (at  $x$  within {0..1})) has-integral 0) {0..1}
using cauchy f0 unfolding has-contour-integral-def
by simp
have m2:  $f\ (\text{circlepath } 0\ 1\ x) / (\text{circlepath } 0\ 1\ x - 0) *$ 
vector-derivative (circlepath 0 1) (at  $x$  within {0..1}) =
 $2 * \text{pi} * i * f\ (\text{cis } (2 * \text{pi} * x))$  if  $x \in \{0..1\}$  for  $x$ 
using vector-derivative-circlepath01[of x 0 1] that unfolding cx
by (simp add: cis-conv-exp field-simps)
have m3: (( $\lambda x. 2 * \text{pi} * i * f\ (\text{cis } (2 * \text{pi} * x))$ ) has-integral 0) {0..1}
using m1 has-integral-cong[of {0..1}, OF m2]
by fast
have cnz: complex-of-real (2 * pi) *  $i \neq 0$ 
by simp
have m4: (( $\lambda x. f\ (\text{cis } (2 * \text{pi} * x))$ ) has-integral 0) {0..1}
using m3 has-integral-mult-right-iff[of 2 * pi * i  $\lambda x. f\ (\text{cis } (2 * \text{pi} * x))$  0 {0..1}] cnz
by simp
have m5: (( $\lambda x. \text{Re } (f\ (\text{cis } (2 * \text{pi} * x)))$ ) has-integral 0) {0..1}
using has-integral-linear[OF m4 bounded-linear-Re]
by (simp add: o-def)
have g1: continuous-on UNIV ( $\lambda\vartheta. \text{Re } (f\ (\text{cis } \vartheta))$ )
using continuous-on-compose2[OF cont, of UNIV cis]
by (fastforce intro: continuous-intros)
have g2: continuous-on {0..1} ( $\lambda x. \text{Re } (f\ (\text{cis } (0 + 2 * \text{pi} * x)))$ )
using continuous-on-compose2[OF g1, of {0..1}  $\lambda x. 0 + 2 * \text{pi} * x$ ]
by (fast intro: continuous-intros)
have si: set-integrable lborel {0..1} ( $\lambda x. \text{Re } (f\ (\text{cis } (0 + 2 * \text{pi} * x)))$ )
using borel-integrable-compact[OF compact-Icc g2] unfolding set-integrable-def
by fastforce
have ie: interval-lebesgue-integral lborel (ereal 0) (ereal 1) ( $\lambda x. \text{Re } (f\ (\text{cis } (0$ 
+ 2 *
 $\text{pi} * x)))$ ) = integral {0..1} ( $\lambda x. \text{Re } (f\ (\text{cis } (0 + 2 * \text{pi} * x)))$ )
using interval-integral-eq-integral[OF - si]
by arg0
have iz: integral {0..1} ( $\lambda x. \text{Re } (f\ (\text{cis } (0 + 2 * \text{pi} * x)))$ ) = 0 using m5
by (auto intro: integral-unique)
have af: interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
pi)) ( $\lambda\vartheta. \text{Re } (f\ (\text{cis } \vartheta))$ ) =
 $2 * \text{pi} * \text{interval-lebesgue-integral lborel } (\text{ereal } 0) (\text{ereal } 1) (\lambda x. \text{Re } (f\ (\text{cis } (0$ 

```

```

+
  2 * pi * x))))
  using LBINT-affine[of 2 * pi 0 2 *
    pi λϑ. Re (f (cis ϑ)) 0] pi-gt-zero
  by simp
  show ?thesis using af ie iz
  by simp
qed
have pt: ln (norm (cis ϑ - a)) = ln (norm a) + Re (f (cis ϑ)) for ϑ
proof -
  have nz: 1 - cis ϑ / a ≠ 0
  proof
    assume 1 - cis ϑ / a = 0
    hence cis ϑ = a using a0
    by (simp add: field-simps)
    thus False using assms
    by auto
  qed
  have eq: norm (cis ϑ - a) = norm a * norm (1 - cis ϑ / a)
  proof -
    have cis ϑ - a = - (a * (1 - cis ϑ / a)) using a0
    by (simp add: field-simps)
    thus ?thesis
    by (simp add: norm-mult)
  qed
  show ?thesis unfolding eq f-def using nz a0 ln-mult[of norm a norm (1 - cis
    ϑ / a)]
  by simp
qed
have c1: continuous-on UNIV (λϑ. f (cis ϑ))
  using continuous-on-compose2[OF cont, of UNIV cis]
  by (fastforce intro: continuous-intros)
have c2: continuous-on {0..2 * pi} (λϑ. Re (f (cis ϑ)))
  using continuous-on-Re[OF c1]
  by (blast intro: continuous-on-subset)
have ii: interval-lebesgue-integrable lborel (ereal 0) (ereal (2 *
  pi)) (λϑ. Re (f (cis ϑ)))
  using interval-integrable-continuous-on[OF - c2] pi-gt-zero
  by simp
show ?thesis unfolding pt using ii mean
  by simp
qed

lemma circle-log-lt:
  assumes norm a < 1
  shows interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) (λϑ. ln (norm (cis ϑ - a))) = 0
proof (cases a = 0)
  case True

```

```

    thus ?thesis by simp
next
case False
define b where b = 1 / cnj a
have nb: norm b = 1 / norm a and b1: 1 < norm b using False assms
  by (auto simp: b-def norm-divide)
have pt: ln (norm (cis ϑ - a)) = ln (norm a) + ln (norm (cis ϑ - b)) for ϑ
proof -
  have c1: cnj (cnj a * cis ϑ - 1) = cis (- ϑ) * (a - cis ϑ)
    by (simp add: cis-cnj algebra-simps cis-mult)
  have norm (cnj a * cis ϑ - 1) = norm (a - cis ϑ)
  using arg-cong[where f = norm, OF c1] unfolding complex-mod-cnj norm-mult
  by auto
  hence e1: norm (cis ϑ - a) = norm (cnj a * cis ϑ - 1)
    using norm-minus-commute[of a cis ϑ]
    by simp
  have e2: norm (cnj a * cis ϑ - 1) = norm a * norm (cis ϑ - b)
  proof -
    have cnj a * cis ϑ - 1 = cnj a * (cis ϑ - b) using False unfolding b-def
      by (simp add: field-simps)
    thus ?thesis by (simp add: norm-mult)
  qed
  have nz: cis ϑ - b ≠ 0 using b1
    by auto
  show ?thesis unfolding e1 e2 using False nz ln-mult[of norm a norm (cis ϑ
- b)]
    by simp
qed
have nzb: norm (cis ϑ - b) ≠ 0 for ϑ using b1
  by auto
have cb: continuous-on {0..2 * pi} (λϑ. ln (norm (cis ϑ - b))) using nzb
  by (fast intro: continuous-intros)
have ib: interval-lebesgue-integrable lborel (ereal 0) (ereal (2 *
pi)) (λϑ. ln (norm (cis ϑ - b)))
  using interval-integrable-continuous-on[OF - cb] pi-gt-zero
  by simp
show ?thesis unfolding pt using ib circle-log-gt[OF b1] nb False
  by (simp add: ln-div)
qed

```

A point of the unit circle as $\text{cis } \varphi$ with $\varphi \in [0, 2\pi]$.

lemma *cis-of-norm-one*:

assumes $\text{norm } a = 1$

obtains φ **where** $a = \text{cis } \varphi$ $0 \leq \varphi < 2 * \pi$

proof -

have $c: a = \text{cis } (\text{Arg } a)$ **using** *rcis-cmod-Arg*[of a] *assms*

by (*simp add: rcis-def*)

show ?thesis

proof (*cases* $0 \leq \text{Arg } a$)

```

    case True
    thus ?thesis using that[of Arg a] c Arg-bounded[of a]
      by linarith
  next
  case False
  have cis (Arg a + 2 * pi) = cis (Arg a)
    by (simp add: cis-mult[symmetric])
  thus ?thesis using that[of Arg a + 2 * pi] c Arg-bounded[of a] False
    by simp
qed
qed

```

```

lemma norm-cis-minus-cis: norm (cis  $\vartheta$  - cis  $\varphi$ ) = norm (1 - cis ( $\vartheta$  -  $\varphi$ ))
proof -
  have q: cis  $\vartheta$  - cis  $\varphi$  = cis  $\varphi$  * (cis ( $\vartheta$  -  $\varphi$ ) - 1)
    by (simp add: cis-mult algebra-simps)
  show ?thesis unfolding q norm-mult
    using norm-minus-commute[of cis ( $\vartheta$  -  $\varphi$ ) 1]
    by simp
qed

```

```

lemma circle-log-eq:
  assumes norm a = 1
  shows interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln (\text{norm} (\text{cis } \vartheta - a))$ ) = 0
proof -
  obtain  $\varphi$  where phi: a = cis  $\varphi$  and p0:  $0 \leq \varphi \leq 2 * \pi$ 
    by (rule cis-of-norm-one[OF assms])
  define h where h u =  $\ln (\text{norm} (1 - \text{cis } u))$  for u
  have per: h (x + 2 * pi) = h x for x unfolding h-def
    by (simp add: cis-mult[symmetric])
  have ih: interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * pi)) h
    using circle-log-one-integrable unfolding h-def[abs-def] .
  have pt:  $\ln (\text{norm} (\text{cis } \vartheta - a)) = h (\vartheta - \varphi)$  for  $\vartheta$ 
    using norm-cis-minus-cis[of  $\vartheta \varphi$ ] unfolding phi h-def by simp
  show ?thesis unfolding pt using periodic-shift-eq[of h  $\varphi$ , OF per ih p0] cir-
    cle-log-one
    unfolding h-def[abs-def]
    by simp
qed

```

```

lemma circle-log-eq-integrable:
  assumes norm a = 1
  shows interval-lebesgue-integrable lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln (\text{norm} (\text{cis } \vartheta - a))$ )
proof -
  obtain  $\varphi$  where phi: a = cis  $\varphi$  and p0:  $0 \leq \varphi \leq 2 * \pi$ 
    by (rule cis-of-norm-one[OF assms])
  define h where h u =  $\ln (\text{norm} (1 - \text{cis } u))$  for u

```

```

have per:  $h(x + 2 * \pi) = h x$  for  $x$  unfolding h-def
by (simp add: cis-mult[symmetric])
have ih: interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * pi)) h
using circle-log-one-integrable unfolding h-def[abs-def] .
have pt:  $\ln(\text{norm}(\text{cis } \vartheta - a)) = h(\vartheta - \varphi)$  for  $\vartheta$ 
using norm-cis-minus-cis[of  $\vartheta \varphi$ ] unfolding phi h-def by simp
show ?thesis unfolding pt using periodic-shift-int[of  $h \varphi$ , OF per ih p0]
unfolding h-def[abs-def]
by simp
qed

```

```

theorem circle-avg-log:
  interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln(\text{norm}(\text{cis } \vartheta - a))$ ) =  $2 * \pi * \ln(\max 1 (\text{norm } a))$ 
proof (cases norm a 1 :: real rule: linorder-cases)
  case less thus ?thesis using circle-log-lt[OF less] by (simp add: max-def)
next
  case equal thus ?thesis using circle-log-eq[OF equal] by (simp add: max-def)
next
  case greater thus ?thesis using circle-log-gt[OF greater] by (simp add: max-def)
qed

```

```

lemma circle-log-integrable:
  interval-lebesgue-integrable lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln(\text{norm}(\text{cis } \vartheta - a))$ )
proof (cases norm a = 1)
  case True
  thus ?thesis by (rule circle-log-eq-integrable)
next
  case False
  have nz:  $\text{norm}(\text{cis } \vartheta - a) \neq 0$  for  $\vartheta$  using False
  by auto
  have cb: continuous-on  $\{0..2 * \pi\}$  ( $\lambda \vartheta. \ln(\text{norm}(\text{cis } \vartheta - a))$ ) using nz
  by (fast intro: continuous-intros)
  show ?thesis using interval-integrable-continuous-on[OF - cb] pi-gt-zero
  by simp
qed

```

Normalised arclength on the circle of radius ε about c .

```

definition circ :: complex  $\Rightarrow$  real  $\Rightarrow$  complex measure where
  circ c  $\varepsilon$  = distr (density lborel ( $\lambda \vartheta. \text{ennreal}(\text{indicator } \{0..2 *
    \pi\} \vartheta / (2 * \pi))$ )) borel
    ( $\lambda \vartheta. c + \text{of-real } \varepsilon * \text{cis } \vartheta$ )

```

The circle average of $\log|z - a|$ over the circle of radius ε about c : $\log \max(\varepsilon, |c - a|)$.

```

lemma circ-log:
  assumes  $0 < \varepsilon$ 
  shows  $\text{integral}^L(\text{circ } c \varepsilon) (\lambda z. \ln(\text{norm}(z - a))) =$ 

```



```

    ln (max ε (norm (c - a)))
proof -
  have integralL (circ c ε) (λz. ln (norm (z - a)))
    = (1 / (2 * pi)) * interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
      pi)) (λϑ. ln (norm (c + of-real ε * cis ϑ - a)))
proof -
  have mg0: (λϑ. c + of-real ε * cis ϑ) ∈
    borel-measurable borel
  by (fast intro: borel-measurable-continuous-onI continuous-intros)
  have mF: (λz. ln (norm (z - a))) ∈ borel-measurable borel
  by simp
  have i1: integralL (distr (density lborel (λϑ. ennreal (indicator {0..2 *
    pi} ϑ / (2 * pi)))) borel (λϑ. c + of-real ε *
    cis ϑ)) (λz. ln (norm (z - a))) =
    integralL (density lborel (λϑ. ennreal (indicator {0..2 * pi} ϑ /
    (2 * pi)))) (λϑ. ln (norm (c + of-real ε * cis ϑ - a)))
  using integral-distr[of λϑ. c + of-real ε *
    cis ϑ (density lborel (λϑ. ennreal (indicator {0..2 * pi} ϑ /
    (2 * pi)))) borel λz. ln (norm (z - a))] mg0 mF
  by fastforce
  have mG: (λϑ. ln (norm (c + of-real ε * cis ϑ - a))) ∈
    borel-measurable borel using measurable-compose[OF mg0 mF]
  by simp
  have i2: integralL (density lborel (λϑ. ennreal (indicator {0..2 *
    pi} ϑ / (2 * pi)))) (λϑ. ln (norm (c + of-real ε *
    cis ϑ - a))) = (∫ ϑ. (indicator {0..2 * pi} ϑ / (2 *
    pi)) *R (λϑ. ln (norm (c + of-real ε * cis ϑ -
    a))) ϑ ∂lborel)
  using integral-density[of (λϑ. ln (norm (c + of-real ε *
    cis ϑ - a))) lborel λϑ. indicator {0..2 * pi} ϑ / (2 *
    pi)] mG
  by fastforce
  have i3a: (λϑ. (indicator {0..2 * pi} ϑ / (2 *
    pi)) *R (λϑ. ln (norm (c + of-real ε * cis ϑ -
    a))) ϑ) = (λϑ. (1 / (2 * pi)) * (indicator {0..2 *
    pi} ϑ *R (λϑ. ln (norm (c + of-real ε * cis ϑ -
    a))) ϑ))
  by (simp add: fun-eq-iff)
  have i3b: (∫ ϑ. (1 / (2 * pi)) * (indicator {0..2 *
    pi} ϑ *R (λϑ. ln (norm (c + of-real ε * cis ϑ -
    a))) ϑ) ∂lborel) =
    (1 / (2 * pi)) * (∫ ϑ. indicator {0..2 *
    pi} ϑ *R (λϑ. ln (norm (c + of-real ε * cis ϑ -
    a))) ϑ ∂lborel)
  by (fast intro: integral-mult-right-zero)
  have i3c: (∫ ϑ. indicator {0..2 *
    pi} ϑ *R (λϑ. ln (norm (c + of-real ε * cis ϑ -
    a))) ϑ ∂lborel) = interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) (λϑ. ln (norm (c + of-real ε * cis ϑ - a)))

```

```

    using interval-integral-Icc[of 0 2 * pi (λϑ. ln (norm (c +
      of-real ε * cis ϑ - a)))] pi-gt-zero unfolding set-lebesgue-integral-def
    by linarith
  show ?thesis unfolding circ-def using i1 i2 i3a i3b i3c
    by presburger
qed
also have ... = (1 / (2 * pi)) * interval-lebesgue-integral lborel (ereal 0) (ereal
(2 *
  pi)) (λϑ. ln ε + ln (norm (cis ϑ - (a - c) /
    of-real ε)))
proof -
  define w where w = (a - c) / of-real ε
  have bad: {ϑ. cis ϑ = w} ⊆
    range (λn::int. Arg w + 2 * pi * of-int n)
proof
  fix ϑ assume th: ϑ ∈ {ϑ. cis ϑ = w}
  have w0: w ≠ 0 using th
    by auto
  have cw: cis (Arg w) = w using cis-Arg[OF w0] th
    by (force simp: sgn-div-norm)
  have sin ϑ = sin (Arg w) ∧ cos ϑ = cos (Arg w) using th cw
    by (simp add: complex-eq-iff)
  then obtain n :: int where ϑ = Arg w + 2 * pi * n using sin-cos-eq-iff by
blast
  thus ϑ ∈ range (λn::int. Arg w + 2 * pi * of-int n) by auto
qed
have null: {ϑ. cis ϑ = w} ∈ null-sets lborel
  using countable-imp-null-set-lborel countable-subset[OF bad]
  by fast
have pw: ln (norm (c + of-real ε * cis ϑ - a)) =
  ln ε + ln (norm (cis ϑ - w)) if nw: cis ϑ ≠ w for ϑ
proof -
  have e: c + of-real ε * cis ϑ - a = of-real ε * (cis ϑ - w)
    using assms unfolding w-def
    by (simp add: field-simps)
  show ?thesis unfolding e norm-mult
    using assms nw ln-mult[of ε norm (cis ϑ - w)]
    by simp
qed
have n1: (λϑ. norm (c + of-real ε * cis ϑ - a)) ∈
  borel-measurable borel
  by (fast intro: borel-measurable-continuous-onI continuous-intros)
have n2: (λϑ. norm (cis ϑ - w)) ∈ borel-measurable borel
  by (fast intro: borel-measurable-continuous-onI continuous-intros)
have m3: (λϑ. ln (norm (c + of-real ε * cis ϑ - a))) ∈
  borel-measurable borel using borel-measurable-ln[OF n1]
  by simp
have m4: (λϑ. ln ε + ln (norm (cis ϑ - w))) ∈
  borel-measurable borel using borel-measurable-ln[OF n2]

```

```

    by (auto intro: borel-measurable-add)
  have ae: AE  $\vartheta$  in lborel.  $\vartheta \in$ 
    einterval (min (ereal 0) (ereal (2 * pi))) (max (ereal 0) (ereal (2 * pi)))  $\longrightarrow$ 
    ( $\lambda \vartheta. \ln (\text{norm } (c + \text{of-real } \varepsilon * \text{cis } \vartheta - a))$ )  $\vartheta =$ 
    ( $\lambda \vartheta. \ln \varepsilon + \ln (\text{norm } (\text{cis } \vartheta - w))$ )  $\vartheta$ 
  using AE-I'[OF null, of  $\lambda \vartheta. \vartheta \in$ 
    einterval (min (ereal 0) (ereal (2 * pi))) (max (ereal 0) (ereal (2 *
    pi)))  $\longrightarrow$  ( $\lambda \vartheta. \ln (\text{norm } (c + \text{of-real } \varepsilon * \text{cis } \vartheta - a))$ )  $\vartheta =$  ( $\lambda \vartheta. \ln \varepsilon + \ln (\text{norm } (\text{cis } \vartheta -$ 
    w)))  $\vartheta]$  pw
  by blast
  have eqI: interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln (\text{norm } (c + \text{of-real } \varepsilon * \text{cis } \vartheta - a))$ ) =
    interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln \varepsilon + \ln (\text{norm } (\text{cis } \vartheta - w))$ )
  using interval-integral-cong-AE[OF m3 m4 ae]
  by fastforce
  show ?thesis unfolding w-def[symmetric] using eqI
  by simp
qed
also have ... = (1 / (2 * pi)) * (2 * pi * ln  $\varepsilon$  + 2 * pi * ln (max 1 (norm ((a
  -
    c) / of-real  $\varepsilon$ ))))
  using circle-log-integrable[of (a - c) / of-real  $\varepsilon$ ] circle-avg-log[of (a - c) /
    of-real  $\varepsilon]$ 
  by simp
also have ... = ln (max  $\varepsilon$  (norm (c - a)))
proof -
  have nd: norm ((a - c) / of-real  $\varepsilon$ ) = norm (c - a) /  $\varepsilon$  using assms
  by (simp add: norm-divide norm-minus-commute)
  have k: ln  $\varepsilon$  + ln (max 1 (norm (c - a) /  $\varepsilon$ )) =
    ln (max  $\varepsilon$  (norm (c - a)))
  proof (cases norm (c - a)  $\leq \varepsilon$ )
    case True
    thus ?thesis using assms
    by (simp add: max-def)
  next
    case False
    thus ?thesis using assms ln-div[of norm (c - a)  $\varepsilon]$ 
    by (auto simp: max-def)
  qed
  have e: (1 / (2 * pi)) * (2 * pi * ln  $\varepsilon$  + 2 * pi * ln (max 1 (norm (c - a) /
     $\varepsilon$ )))
    = ln  $\varepsilon$  + ln (max 1 (norm (c - a) /  $\varepsilon$ ))
  by (simp add: field-simps)
  show ?thesis unfolding nd e k ..
qed
finally show ?thesis .
qed

```

end

theory *Arcsine-Distribution*
imports *Log-Sine-Integral*
begin

11 The arcsine distribution and its logarithmic potential

The arcsine law on (a, b) with density $1 / (\pi \sqrt{(t - a)(b - t)})$, as the pushforward of the uniform distribution on $(0, \pi)$ under $\vartheta \mapsto (a+b)/2 - (b-a)/2 \cdot \cos \vartheta$; its logarithmic potential $U(t) = \int \ln |t - u| d\omega(u)$, which equals $\ln((b - a)/4)$ on $[a, b]$ and has an explicit (Joukowski) form outside; and the bound $\omega[x - r, x + r] \leq 8 \sqrt{r} / (\pi \sqrt{(b - a)})$ on the mass of short intervals.

The arcsine probability density on (a, b) , and the measure.

definition *arcsine-dens* :: *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* **where**
arcsine-dens $a\ b\ t = (\text{if } a < t \wedge t < b \text{ then } 1 / (\pi * \text{sqrt } ((t - a) * (b - t)))$
else $0)$

definition *arcsine* :: *real* $\Rightarrow$ *real* $\Rightarrow$ *real measure* **where**
arcsine $a\ b = \text{density lborel } (\lambda t. \text{ennreal } (\text{arcsine-dens } a\ b\ t))$

The logarithmic potential $U^\mu(t) = \int \log |t - u| d\mu(u)$ of a measure on the line.

definition *logPot* :: *real measure* $\Rightarrow$ *real* $\Rightarrow$ *real* **where**
logPot $\mu\ t = \text{integral}^L \mu (\lambda u. \ln |t - u|)$

11.1 The arcsine measure

lemma *arcsine-distr*:

assumes $a < b$

shows $\text{arcsine } a\ b = \text{distr } (\text{density lborel } (\lambda \vartheta. \text{ennreal } (\text{indicator } \{0 < \vartheta < \pi\} \vartheta / \pi))) \text{ borel } (\lambda \vartheta. (a + b) / 2 - (b - a) / 2 * \cos \vartheta)$

proof –

define m **where** $m = (a + b) / 2$

define r **where** $r = (b - a) / 2$

define g **where** $g = (\lambda \vartheta. m - r * \cos \vartheta)$

have $r0$: $0 < r$ **using** *assms* **unfolding** *r-def* **by** *simp*

have $g\text{-deriv}$: $\bigwedge \vartheta. (g \text{ has-real-derivative } (r * \sin \vartheta))$ (at ϑ)

unfolding *g-def* **by** (auto intro!: *derivative-eq-intros*)

have $g0$: $g\ 0 = a$ **unfolding** *g-def* *m-def* *r-def*

by (*simp add: field-simps*)

have $g\pi$: $g\ \pi = b$ **unfolding** *g-def* *m-def* *r-def*

by (*simp add: field-simps*)

```

note  $g\text{-ends} = g0 \text{ gpi}$ 
have  $g\text{-pos}: \bigwedge \vartheta. \vartheta \in \{0..pi\} \implies 0 \leq r * \sin \vartheta$ 
  using  $r0$  by (simp add: sin-ge-zero)
have  $dens: \bigwedge \vartheta. \vartheta \in \{0<..
   $\arcsine\text{-dens } a \ b \ (g \ \vartheta) * (r * \sin \vartheta) = 1 / \pi$ 
proof -
  fix  $\vartheta :: \text{real}$  assume  $th: \vartheta \in \{0<..
  have  $s0: 0 < \sin \vartheta$  using  $th$  by (simp add: sin-gt-zero)
  hence  $0 < \sin \vartheta ^ 2$  by simp
  hence  $\cos \vartheta ^ 2 < 1$  using sin-cos-squared-add[ $\text{of } \vartheta$ ] by linarith
  hence  $c1: |\cos \vartheta| < 1$  using abs-square-less-1[ $\text{of } \cos \vartheta$ ]
    by simp
  have  $ab: a = m - r \ b = m + r$  unfolding  $m\text{-def } r\text{-def}$  by (simp-all add:
field-simps)
  have  $r * \cos \vartheta < r$  using  $c1 \ r0$ 
    by (simp add: abs-less-iff)
  moreover have  $-r < r * \cos \vartheta$  using  $c1 \ r0$ 
    using mult-strict-left-mono[ $\text{of } -1 \ \cos \vartheta \ r$ ]  $c1 \ r0$  abs-less-iff[ $\text{of } \cos \vartheta \ 1$ ]
    by linarith
  ultimately have  $ga: a < g \ \vartheta \wedge g \ \vartheta < b$  unfolding  $g\text{-def } ab$ 
    by auto
  have  $pr0: (g \ \vartheta - a) * (b - g \ \vartheta) = r ^ 2 * (1 - \cos \vartheta ^ 2)$ 
    unfolding  $g\text{-def } ab$ 
    by (simp add: power2-eq-square algebra-simps)
  have  $sc: 1 - \cos \vartheta ^ 2 = \sin \vartheta ^ 2$  using sin-cos-squared-add[ $\text{of } \vartheta$ ]
    by linarith
  have  $pr: (g \ \vartheta - a) * (b - g \ \vartheta) = (r * \sin \vartheta) ^ 2$  unfolding  $pr0 \ sc$ 
    by (simp add: power-mult-distrib)
  have  $sq: \text{sqrt } ((g \ \vartheta - a) * (b - g \ \vartheta)) = r * \sin \vartheta$  unfolding  $pr$ 
    using  $r0 \ s0$  by simp
  show  $\arcsine\text{-dens } a \ b \ (g \ \vartheta) * (r * \sin \vartheta) = 1 / \pi$  unfolding  $\arcsine\text{-dens-def}$ 
    using  $ga \ sq \ r0 \ s0$ 
    by simp
qed
define  $U$  where  $U = \text{density lborel } (\lambda \vartheta. \text{ennreal } (\text{indicator } \{0<..
 $\pi))$ 
have  $mg: g \in \text{borel-measurable borel}$  unfolding  $g\text{-def}$ 
  by (fast intro: borel-measurable-continuous-onI continuous-intros)
have  $meq: \arcsine \ a \ b = \text{distr } U \ \text{borel } g$ 
proof (rule measure-eqI)
  show  $\text{sets } (\arcsine \ a \ b) = \text{sets } (\text{distr } U \ \text{borel } g)$  unfolding  $\arcsine\text{-def } U\text{-def}$ 
    by simp
  fix  $A$  assume  $A \in \text{sets } (\arcsine \ a \ b)$ 
  hence  $A: A \in \text{sets borel}$  unfolding  $\arcsine\text{-def}$  by simp
  define  $F$  where  $F \ t = \arcsine\text{-dens } a \ b \ t * \text{indicator } A \ t$  for  $t$ 
  have  $md: (\lambda t. \arcsine\text{-dens } a \ b \ t) \in \text{borel-measurable borel}$ 
    unfolding  $\arcsine\text{-dens-def}$  by measurable
  have  $mF: F \in \text{borel-measurable borel}$  unfolding  $F\text{-def}$  using  $A \ md$ 
    by (blast intro: borel-measurable-times borel-measurable-indicator)$$$ 
```

```

have l0: emeasure (arcsine a b) A =
  (∫+ t. ennreal (arcsine-dens a b t) * indicator A t ∂lborel)
unfolding arcsine-def using A
using md
by (auto simp: emeasure-density)
have pw1: ennreal (arcsine-dens a b t) * indicator A t =
  ennreal (F t * indicator {g 0..g pi} t) for t
unfolding F-def g-ends arcsine-dens-def
by (simp add: indicator-def)
have l1: emeasure (arcsine a b) A =
  (∫+ t. ennreal (F t * indicator {g 0..g pi} t) ∂lborel)
unfolding l0 pw1 ..
have sbF: set-borel-measurable borel {g 0..g pi} F using mF unfolding
set-borel-measurable-def
by simp
have cg: continuous-on {0..pi} (λx. r * sin x)
by (fast intro: continuous-intros)
have l2: (∫+ t. ennreal (F t * indicator {g 0..g pi} t) ∂lborel) =
  (∫+ ∅. ennreal (F (g ∅) * (r * sin ∅) *
    indicator {0..pi} ∅) ∂lborel)
using nn-integral-substitution[where f=F and g=g and a=0 and b=pi and
g'=λx. r *
  sin x, OF sbF - cg] g-deriv g-pos pi-gt-zero
by simp
have pw3: ennreal (F (g ∅) * (r * sin ∅) * indicator {0..pi} ∅) =
  ennreal (indicator {0<..+ ∅. ennreal (indicator {0<..

```

Integration against the arcsine measure: $\int f d\omega = \pi^{-1} \int_{-\pi}^{\pi} f(m)$

– $r \cos \vartheta$) $d\vartheta$ (substitution $t = m - r \cos \vartheta$: *has-integral-substitution*).

theorem *integral-arcsine*:

assumes $a < b$ $f \in \text{borel-measurable borel}$

shows $\text{integral}^L (\text{arcsine } a \ b) \ f =$

$$1 / \pi * \text{set-lebesgue-integral lborel } \{0 < .. < \pi\} (\lambda \vartheta. f ((a + b) / 2 - (b - a) / 2 * \cos \vartheta))$$

proof –

define g **where** $g = (\lambda \vartheta :: \text{real}. (a + b) / 2 - (b - a) / 2 * \cos \vartheta)$

define U **where** $U = \text{density lborel } (\lambda \vartheta. \text{ennreal } (\text{indicator } \{0 < .. < \pi\} \ \vartheta / \pi))$

have $\text{meq}: \text{arcsine } a \ b = \text{distr } U \ \text{borel } g$ **unfolding** $U\text{-def } g\text{-def}$

by $(\text{rule arcsine-distr}[OF \ \text{assms}(1)])$

have $\text{mg}: g \in \text{borel-measurable borel}$ **unfolding** $g\text{-def}$

by $(\text{fast intro: borel-measurable-continuous-onI continuous-intros})$

have $\text{mgU}: g \in \text{measurable } U \ \text{borel}$ **using** mg **unfolding** $U\text{-def}$ **by** simp

have $\text{mfg}: (\lambda \vartheta. f (g \ \vartheta)) \in \text{borel-measurable borel}$

using $\text{measurable-compose}[OF \ \text{mg} \ \text{assms}(2)]$

by simp

have $\text{integral}^L (\text{arcsine } a \ b) \ f = \text{integral}^L \ U \ (\lambda \vartheta. f (g \ \vartheta))$

unfolding meq

using $\text{integral-distr}[OF \ \text{mgU} \ \text{assms}(2)]$

by fast

moreover have $\text{integral}^L \ U \ (\lambda \vartheta. f (g \ \vartheta)) =$

$$(\int \vartheta. (\text{indicator } \{0 < .. < \pi\} \ \vartheta / \pi) *_{\mathbb{R}} f (g \ \vartheta) \ \partial \text{lborel})$$

unfolding $U\text{-def}$ **using** $\text{integral-density}[of \ \lambda \vartheta. f (g \ \vartheta) \ \text{lborel}]$

$$\lambda \vartheta. \text{indicator } \{0 < .. < \pi\} \ \vartheta / \pi] \ \text{mfg}$$

by simp

moreover have $(\int \vartheta. (\text{indicator } \{0 < .. < \pi\} \ \vartheta / \pi) *_{\mathbb{R}} f (g \ \vartheta) \ \partial \text{lborel}) =$

$$1 / \pi * \text{set-lebesgue-integral lborel } \{0 < .. < \pi\} (\lambda \vartheta. f (g \ \vartheta))$$

unfolding $\text{set-lebesgue-integral-def}$

by $(\text{simp add: integral-mult-right-zero[symmetric]})$

ultimately show $?thesis$ **unfolding** $g\text{-def}$

by linarith

qed

lemma *arcsine-sqrt-identity*:

assumes $a < t < b$

shows $\text{sqrt } (1 - ((2 * t - a - b) / (b - a))^2) = 2 * \text{sqrt } ((t - a) * (b - t)) / (b - a)$

proof –

define X **where** $X = (t - a) * (b - t)$

have $X0: 0 \leq X$ **using** assms **by** $(\text{simp add: } X\text{-def})$

have $\text{ba}: 0 < b - a$ **using** assms **by** simp

have $\text{num}: (b - a)^2 - (2 * t - a - b)^2 = 4 * X$

by $(\text{simp add: } X\text{-def algebra-simps power2-eq-square})$

have $1 - ((2 * t - a - b) / (b - a))^2 =$

$$((b - a)^2 - (2 * t - a - b)^2) / (b - a)^2$$

```

proof -
  have g:  $1 - (c / d)^2 = (d^2 - c^2) / d^2$  if  $d \neq 0$ 
  for  $c\ d :: \text{real}$ 
  using that by (simp add: power-divide field-simps)
  show ?thesis by (rule g) (use ba in simp)
qed
also have  $\dots = 4 * X / (b - a)^2$  unfolding num ..
also have  $\dots = (2 * \text{sqrt } X / (b - a))^2$ 
  using X0 by (simp add: power-divide power-mult-distrib)
finally have  $\text{sqrt } (1 - ((2 * t - a - b) / (b - a))^2) = |2 * \text{sqrt } X / (b - a)|$ 
  by simp
also have  $\dots = 2 * \text{sqrt } X / (b - a)$  using ba X0 by simp
finally show ?thesis by (simp add: X-def)
qed

lemma arcsine-dens-integral:
  assumes ab:  $a < b$ 
  shows  $(\int^+ t. \text{ennreal } (\text{arcsine-dens } a\ b\ t) \ \partial \text{lborel}) = 1$ 
proof -
  define u where  $u\ t = (2 * t - a - b) / (b - a)$  for t
  define F where  $F\ t = \arcsin (u\ t) / \pi$  for t
  define f where  $f\ t = 1 / (\pi * \text{sqrt } ((t - a) * (b - t)))$  for t
  have ba:  $0 < b - a$  using ab by simp
  have u-rng:  $-1 < u\ t \wedge u\ t < 1$  if  $a < t < b$  for t
    using that ba by (auto simp: u-def field-simps)
  have u-rng':  $-1 \leq u\ t \wedge u\ t \leq 1$  if  $a \leq t \leq b$  for t
    using that ba by (auto simp: u-def field-simps)
  have deriv:  $(F \text{ has-real-derivative } f\ t) \text{ (at } t) \text{ if } t: a < t < b$  for t
  proof -
    have ue:  $u = (\lambda t. (2 / (b - a)) * t - (a + b) / (b - a))$ 
    by (rule ext) (simp add: u-def diff-divide-distrib add-divide-distrib)
    have lin:  $((\lambda t. c * t - e) \text{ has-real-derivative } c) \text{ (at } t) \text{ for } c\ e :: \text{real}$ 
    by (auto intro!: derivative-eq-intros)
    have du:  $(u \text{ has-real-derivative } 2 / (b - a)) \text{ (at } t)$ 
    unfolding ue by (rule lin)
    have  $((\lambda t. \arcsin (u\ t)) \text{ has-real-derivative inverse } (\text{sqrt } (1 - (u\ t)^2))) * (2 / (b - a))) \text{ (at } t)$ 
    using DERIV-arcsin[of u t] u-rng[OF t] du by (intro DERIV-chain2) auto
    hence  $(F \text{ has-real-derivative inverse } (\text{sqrt } (1 - (u\ t)^2))) * (2 / (b - a)) / \pi$ 
    (at t)
    unfolding F-def[abs-def] by (rule DERIV-cdivide)
    moreover have  $\text{inverse } (\text{sqrt } (1 - (u\ t)^2)) * (2 / (b - a)) / \pi = f\ t$ 
  proof -
    have pos:  $0 < (t - a) * (b - t)$  using t by simp
    show ?thesis unfolding u-def arcsine-sqrt-identity[OF t] f-def
    using ba pos by (simp add: field-simps)
  qed
  ultimately show ?thesis by simp
qed

```



```

have conf: isCont f t if t: a < t < b for t
proof -
  have 0 < (t - a) * (b - t) using t by simp
  thus ?thesis unfolding f-def by (auto intro!: continuous-intros)
qed
have contF: continuous-on {a..b} F
  unfolding F-def[abs-def] u-def
  by (intro continuous-intros continuous-on-arcsin) (use u-rng' ab in ⟨auto simp:
    u-def⟩)
have lima: ((F ∘ real-of-ereal) ⟶ F a) (at-right (ereal a))
  unfolding ereal-tendsto-simps by (rule continuous-on-Icc-at-rightD[OF contF
    ab])
have limb: ((F ∘ real-of-ereal) ⟶ F b) (at-left (ereal b))
  unfolding ereal-tendsto-simps by (rule continuous-on-Icc-at-leftD[OF contF
    ab])
have nonneg: AE x in lborel. ereal a < ereal x ⟶
  ereal x < ereal b ⟶ 0 ≤ f x
  by (auto simp: f-def)
note FTC = interval-integral-FTC-nonneg[of ereal a ereal b F f, OF - - - nonneg
  lima limb]
have si: set-integrable lborel {a<..b} f
  using FTC(1) ab deriv conf by (simp add: einterval-eq-Icc)
have ua: u a = - 1 and ub: u b = 1 using ba by (simp-all add: u-def field-simps)
have Fab: F b - F a = 1 by (simp add: F-def ua ub arcsin-minus-1)
have int: interval-lebesgue-integral lborel (ereal a) (ereal b) f = 1
  using FTC(2) ab deriv conf Fab by simp
have eq: arcsine-dens a b t = indicator {a<..b} t *R f t for t
  by (simp add: arcsine-dens-def f-def indicator-def)
have (∫+ t. ennreal (arcsine-dens a b t) ∂lborel) =
  ennreal (∫ t. indicator {a<..b} t *R f t ∂lborel)
  unfolding eq
  by (rule nn-integral-eq-integral) (use si in ⟨auto simp: set-integrable-def f-def
    indicator-def⟩)
also have (∫ t. indicator {a<..b} t *R f t ∂lborel) = 1
  using int ab by (simp add: interval-lebesgue-integral-def einterval-eq-Icc
    set-lebesgue-integral-def)
finally show ?thesis by simp
qed

```

```

lemma arcsine-dens-measurable [measurable]: (λt. arcsine-dens a b t) ∈
  borel-measurable borel
  unfolding arcsine-dens-def by measurable

```

```

theorem arcsine-univ: a < b ⟹ emeasure (arcsine a b) UNIV = 1
  unfolding arcsine-def by (subst emeasure-density) (auto simp: arcsine-dens-integral)

```

(A.1): the potential of the unit arcsine measure (Joukowski parameter, the integral of $\log|x - \cos \vartheta|$ over a period; alternatively via *Cauchy-integral-circlepath*).

The Joukowski radius $|w|$ with $w^2 + 1 = 2xw$: 1 for $|x| \leq 1$, else $|x| +$

$\sqrt{(x^2 - 1)}$.

definition *jouk* :: *real* $\Rightarrow$ *real* **where**

jouk *x* = (if $|x| \leq 1$ then 1 else $|x| + \text{sqrt } (x^2 - 1)$)

lemma *jouk-ge-1*: $1 \leq \text{jouk } x$

unfolding *jouk-def* **using** *real-sqrt-ge-zero*[of $x^2 - 1$]

using *abs-square-le-1*[of *x*]

by *argo*

lemma *exists-w*: $\exists w :: \text{complex}. w^2 + 1 = 2 * \text{of-real } x * w \wedge \text{norm } w = \text{jouk } x$

proof (cases $|x| \leq 1$)

case *True*

define *s* **where** $s = \text{sqrt } (1 - x^2)$

have *x2*: $x^2 \leq 1$ **using** *True abs-square-le-1*

by *blast*

have *s2*: $s^2 = 1 - x^2$ **unfolding** *s-def* **using** *x2*

by *simp*

define *w* **where** $w = \text{Complex } x s$

have *e*: $w^2 + 1 = 2 * \text{of-real } x * w$ **unfolding** *w-def* **using** *s2*

by (*simp add: complex-eq-iff power2-eq-square algebra-simps*)

have *n*: $\text{norm } w = 1$ **unfolding** *w-def cmod-def* **using** *s2*

by *simp*

show *?thesis* **using** *e n True* **unfolding** *jouk-def*

by *meson*

next

case *False*

define *s* **where** $s = \text{sqrt } (x^2 - 1)$

have *x2*: $1 \leq x^2$ **using** *False abs-square-le-1*[of *x*]

by *linarith*

have *s2*: $s^2 = x^2 - 1$ **unfolding** *s-def* **using** *x2*

by *simp*

have *s0*: $0 \leq s$ **unfolding** *s-def* **using** *x2*

by *simp*

have *sg*: $\text{sgn } x * \text{sgn } x = 1$ **using** *False*

by (*auto simp: sgn-if*)

define *w* **where** $w = \text{complex-of-real } (x + \text{sgn } x * s)$

have *e0*: $(x + \text{sgn } x * s)^2 + 1 = 2 * x * (x + \text{sgn } x * s)$ **using** *s2 sg*

by (*simp add: power2-eq-square algebra-simps*)

have *e*: $w^2 + 1 = 2 * \text{of-real } x * w$ **unfolding** *w-def*

using *arg-cong*[**where** $f = \text{complex-of-real}, OF e0$]

by *simp*

have *n0*: $|x + \text{sgn } x * s| = |x| + s$ **using** *s0 False*

by (*force simp: sgn-if abs-if*)

have *n*: $\text{norm } w = |x| + s$ **unfolding** *w-def* **using** *n0*

unfolding *norm-of-real*

by *meson*

show *?thesis* **using** *e n False* **unfolding** *jouk-def s-def*

by *metis*

qed

lemma joukowski:

```

  assumes  $w^2 + 1 = 2 * \text{of-real } x * w$ 
  shows  $\text{norm } (\text{cis } \vartheta - w) * \text{norm } (\text{cis } \vartheta - \text{cnj } w) =$ 
     $2 * \text{norm } w * |x - \cos \vartheta|$ 
proof -
  define u where  $u = \text{Re } w$ 
  define v where  $v = \text{Im } w$ 
  define c where  $c = \cos \vartheta$ 
  define s where  $s = \sin \vartheta$ 
  have re:  $u^2 - v^2 + 1 = 2 * x * u$  using arg-cong[where  $f = \text{Re}$ , OF
  assms]
  unfolding u-def v-def
  by (simp add: power2-eq-square)
  have im:  $2 * u * v = 2 * x * v$  using arg-cong[where  $f = \text{Im}$ , OF assms]
  unfolding u-def v-def
  by (simp add: power2-eq-square)
  have cs:  $c^2 + s^2 = 1$  unfolding c-def s-def by simp
  have n1:  $\text{norm } (\text{cis } \vartheta - w)^2 = (c - u)^2 + (s - v)^2$ 
  unfolding c-def s-def u-def v-def
  by (simp add: cmod-power2)
  have n2:  $\text{norm } (\text{cis } \vartheta - \text{cnj } w)^2 = (c - u)^2 + (s + v)^2$ 
  unfolding c-def s-def u-def v-def
  by (simp add: cmod-power2)
  have nw:  $\text{norm } w^2 = u^2 + v^2$  unfolding u-def v-def by (simp add:
  cmod-power2)
  have vu:  $v = 0 \vee u = x$  using im by auto
  have sq0:  $((c - u)^2 + (s - v)^2) * ((c - u)^2 + (s + v)^2) =$ 
     $4 * (u^2 + v^2) * (x - c)^2$ 
  proof (cases  $v = 0$ )
  case True
  have h:  $(c - u)^2 + s^2 = 2 * u * (x - c)$  using cs re True
  by (simp add: power2-eq-square algebra-simps)
  show ?thesis using h True
  by (simp add: power2-eq-square algebra-simps)
  next
  case False
  hence ux:  $u = x$  using vu by simp
  have v2:  $v^2 = 1 - x^2$  using re ux by (simp add: power2-eq-square)
  have s2:  $s^2 = 1 - c^2$  using cs by simp
  have fac:  $((c - x)^2 + (s - v)^2) * ((c - x)^2 + (s + v)^2) =$ 
     $((c - x)^2 + s^2 + v^2)^2 - 4 * s^2 * v^2$ 
  by (simp add: power2-eq-square algebra-simps)
  have fin:  $((c - x)^2 + (1 - c^2) + (1 - x^2))^2 - 4 * (1 - c^2)$ 
     $* (1 - x^2) =$ 
     $4 * (x^2 + (1 - x^2)) * (x - c)^2$ 
  by (simp add: power2-eq-square algebra-simps)
  show ?thesis unfolding ux fac s2 v2 using fin

```

```

    by blast
  qed
  have sq: (norm (cis  $\vartheta$  - w) * norm (cis  $\vartheta$  - cnj w)) ^ 2 =
    (2 * norm w * |x - cos  $\vartheta$ |) ^ 2
    unfolding power-mult-distrib n1 n2 nw power2-abs c-def[symmetric] using sq0
    by simp
  show ?thesis using sq power2-eq-iff-nonneg[of norm (cis  $\vartheta$  - w) * norm (cis  $\vartheta$ 
    -
    cnj w) 2 * norm w * |x - cos  $\vartheta$ |]
    by simp
  qed

lemma cos-level-null:  $\{\vartheta :: \text{real}. \cos \vartheta = x\} \in \text{null-sets lborel}$ 
proof (cases  $|x| \leq 1$ )
  case True
  have sub:  $\{\vartheta :: \text{real}. \cos \vartheta = x\} \subseteq$ 
    range  $(\lambda n::\text{int}. \arccos x + 2 * \text{of-int } n * \pi) \cup \text{range } (\lambda n::\text{int}. -$ 
    arccos  $x + 2 * \text{of-int } n * \pi)$ 
  proof
    fix  $\vartheta :: \text{real}$  assume th:  $\vartheta \in \{\vartheta. \cos \vartheta = x\}$ 
    have  $\cos \vartheta = \cos (\arccos x)$  using th True
    by (simp add: cos-arccos-abs)
    then obtain n where  $n: n \in \mathbb{Z} \ \vartheta = \arccos x + 2 * n * \pi \vee$ 
       $\vartheta = - \arccos x + 2 * n * \pi$  using cos-eq by blast
    then obtain k :: int where  $n = \text{of-int } k$  using Ints-cases by blast
    thus  $\vartheta \in \text{range } (\lambda n::\text{int}. \arccos x + 2 * \text{of-int } n * \pi) \cup \text{range } (\lambda n::\text{int}. -$ 
      arccos  $x + 2 * \text{of-int } n * \pi)$  using n(2)
    by fast
  qed
  show ?thesis using countable-imp-null-set-lborel countable-subset[OF sub]
    by fast
next
  case False
  hence  $\{\vartheta :: \text{real}. \cos \vartheta = x\} = \{\}$  using abs-cos-le-one
    by blast
  thus ?thesis by simp
qed

lemma log-cos-pt:
  assumes  $w: w ^ 2 + 1 = 2 * \text{of-real } x * w$  and  $w1: 1 \leq \text{norm } w$  and th:  $\cos$ 
 $\vartheta \neq$ 
 $x$ 
  shows  $\ln |x - \cos \vartheta| =$ 
 $\ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) - \ln 2 - \ln (\text{norm } w)$ 
proof -
  have j:  $\text{norm } (\text{cis } \vartheta - w) * \text{norm } (\text{cis } \vartheta - \text{cnj } w) =$ 
 $2 * \text{norm } w * |x - \cos \vartheta|$  by (rule joukowski[OF w])
  have p:  $0 < 2 * \text{norm } w * |x - \cos \vartheta|$  using w1 th
    by (force intro: mult-pos-pos)

```

have $n1$: $\text{norm } (\text{cis } \vartheta - w) \neq 0$ **and** $n2$: $\text{norm } (\text{cis } \vartheta - \text{cnj } w) \neq 0$
using j p **by** *auto*
have $\ln (\text{norm } (\text{cis } \vartheta - w) * \text{norm } (\text{cis } \vartheta - \text{cnj } w)) =$
 $\ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w))$
using $n1$ $n2$ *ln-mult* **by** *simp*
moreover **have** $\ln (2 * \text{norm } w * |x - \cos \vartheta|) =$
 $\ln 2 + \ln (\text{norm } w) + \ln |x - \cos \vartheta|$
using $w1$ *th ln-mult[of 2 * norm w |x - cos v|]* *ln-mult[of 2 norm w]*
by *fastforce*
ultimately **show** *?thesis* **unfolding** j
by *linarith*
qed

lemma *log-cos-ae*:
assumes w : $w^2 + 1 = 2 * \text{of-real } x * w$ **and** $w1$: $1 \leq \text{norm } w$
shows $AE \vartheta$ *in lborel*. $P \vartheta \longrightarrow$
 $\ln |x - \cos \vartheta| = (\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) +$
 $\ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) - \ln 2 - \ln (\text{norm } w)) \vartheta$
using $AE-I'[OF \text{cos-level-null[of } x], \text{ of } \lambda \vartheta. P \vartheta \longrightarrow$
 $\ln |x - \cos \vartheta| = (\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) +$
 $\ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) - \ln 2 - \ln (\text{norm } w)) \vartheta]$ *log-cos-pt[OF w w1]*
by *blast*

lemma *log-cos-meas*: $(\lambda \vartheta :: \text{real}. \ln |x - \cos \vartheta|) \in$
borel-measurable borel
proof –
have h : $(\lambda \vartheta :: \text{real}. |x - \cos \vartheta|) \in \text{borel-measurable borel}$
by (*force intro: borel-measurable-continuous-onI continuous-intros*)
show *?thesis* **using** *borel-measurable-ln[OF h]*
by *simp*
qed

lemma *log-cos-rhs-meas*: $(\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta -$
 $\text{cnj } w)) - \ln 2 - \ln (\text{norm } w)) \in \text{borel-measurable borel}$
proof –
have $(\lambda \vartheta. \text{norm } (\text{cis } \vartheta - w)) \in$
borel-measurable borel $(\lambda \vartheta. \text{norm } (\text{cis } \vartheta - \text{cnj } w)) \in$
borel-measurable borel
by (*fast intro: borel-measurable-continuous-onI continuous-intros*) +
thus *?thesis*
by (*fastforce intro: borel-measurable-ln borel-measurable-add borel-measurable-diff*)
qed

lemma *log-cos-integrable*: *interval-lebesgue-integrable lborel (ereal 0) (ereal (2 * pi))*
 $(\lambda \vartheta. \ln |x - \cos \vartheta|)$
proof –
obtain $w :: \text{complex}$ **where** w : $w^2 + 1 = 2 * \text{of-real } x * w$ $\text{norm } w = \text{jouk } x$
using *exists-w*
by *blast*

```

have w1:  $1 \leq \text{norm } w$  using w(2) jouk-ge-1 by simp
have ig: interval-lebesgue-integrable lborel (ereal 0) (ereal (2 *
  pi)) ( $\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) -$ 
   $\ln 2 - \ln (\text{norm } w)$ )
using circle-log-integrable[of w] circle-log-integrable[of cnj w]
by simp
have ae:  $AE \vartheta \in \text{einterval } (\min (\text{ereal } 0) (\text{ereal } (2 * \text{pi}))) (\max (\text{ereal } 0) (\text{ereal } (2 * \text{pi})))$ 
   $\text{in lborel. } \ln |x - \cos \vartheta| =$ 
   $(\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) - \ln 2 -$ 
   $\ln (\text{norm } w)) \vartheta$ 
by (rule log-cos-ae[OF w(1) w1])
show ?thesis using interval-lebesgue-integrable-cong-AE[OF - ae] log-cos-meas[of
x]
  log-cos-rhs-meas[of w] ig
by simp
qed

 $\int -\vartheta \in \{2\pi\} \log|x - \cos \vartheta| d\vartheta = 2\pi(\log \text{jouk}(x) - \log 2).$ 

lemma log-cos-integral:
  interval-lebesgue-integral lborel (ereal 0) (ereal (2 * pi)) ( $\lambda \vartheta. \ln |x -$ 
   $\cos \vartheta|$ ) =  $2 * \text{pi} * (\ln (\text{jouk } x) - \ln 2)$ 
proof -
  obtain w :: complex where w:  $w^2 + 1 = 2 * \text{of-real } x * w$   $\text{norm } w = \text{jouk } x$ 
using exists-w
  by blast
  have w1:  $1 \leq \text{norm } w$  using w(2) jouk-ge-1 by simp
  have ae:  $AE \vartheta \in \text{einterval } (\min (\text{ereal } 0) (\text{ereal } (2 * \text{pi}))) (\max (\text{ereal } 0) (\text{ereal } (2 * \text{pi})))$ 
     $\text{in lborel. } \ln |x - \cos \vartheta| =$ 
     $(\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) - \ln 2 -$ 
     $\ln (\text{norm } w)) \vartheta$ 
    by (rule log-cos-ae[OF w(1) w1])
  have interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln |x - \cos \vartheta|$ ) =
    interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) -$ 
     $\ln 2 - \ln (\text{norm } w)$ )
    using interval-integral-cong-AE[OF log-cos-meas[of x] log-cos-rhs-meas[of w]
ae]
    by fast
  moreover have interval-lebesgue-integral lborel (ereal 0) (ereal (2 *
    pi)) ( $\lambda \vartheta. \ln (\text{norm } (\text{cis } \vartheta - w)) + \ln (\text{norm } (\text{cis } \vartheta - \text{cnj } w)) -$ 
     $\ln 2 - \ln (\text{norm } w)$ ) =  $2 * \text{pi} * \ln (\text{norm } w) + 2 * \text{pi} * \ln (\text{norm } w) - \ln 2 * (2 * \text{pi}) -$ 
     $\ln (\text{norm } w) * (2 * \text{pi})$ 
    using circle-log-integrable[of w] circle-log-integrable[of cnj w] circle-avg-log[of
w]
    circle-avg-log[of cnj w] w1

```

by (simp add: max-absorb2)
 ultimately show ?thesis unfolding w(2)[symmetric]
 by (simp add: algebra-simps)
 qed

lemma log-cos-half:

interval-lebesgue-integral lborel (ereal 0) (ereal pi) ($\lambda \vartheta. \ln |x - \cos \vartheta|$) = $\pi * (\ln (\text{jouk } x) - \ln 2)$

proof –

have s: interval-lebesgue-integral lborel (ereal (0)) (ereal (pi)) ($\lambda \vartheta. \ln |x - \cos \vartheta|$) + interval-lebesgue-integral lborel (ereal (pi)) (ereal (2 * pi)) ($\lambda \vartheta. \ln |x - \cos \vartheta|$) =
 interval-lebesgue-integral lborel (ereal (0)) (ereal (2 * pi)) ($\lambda \vartheta. \ln |x - \cos \vartheta|$)

using interval-integral-sum[of ereal 0 ereal pi ereal (2 * pi) ($\lambda \vartheta. \ln |x - \cos \vartheta|$)] log-cos-integrable[of x] pi-gt-zero

by simp

have e: ($\lambda t. \ln |x - \cos (2 * \pi - t)|$) =
 ($\lambda \vartheta. \ln |x - \cos \vartheta|$)

by (simp add: cos-diff)

have r: interval-lebesgue-integral lborel (ereal (pi)) (ereal (2 * pi)) ($\lambda \vartheta. \ln |x - \cos \vartheta|$) =
 interval-lebesgue-integral lborel (ereal (0)) (ereal (pi)) ($\lambda \vartheta. \ln |x - \cos \vartheta|$)

using LBINT-reflect[of pi 2 * pi ($\lambda \vartheta. \ln |x - \cos \vartheta|$) 2 * pi] pi-gt-zero unfolding e

by simp

show ?thesis using s r log-cos-integral[of x]

by linarith

qed

lemma arcsine-integrable-ln:

assumes ab: $a < b$

shows integrable (arcsine a b) ($\lambda u. \ln |t - u|$)

proof –

define m where $m = (a + b) / 2$

define r where $r = (b - a) / 2$

define y where $y = (m - t) / r$

define g where $g = (\lambda \vartheta. (a + b) / 2 - (b - a) / 2 * \cos \vartheta)$

define U where $U = \text{density lborel } (\lambda \vartheta. \text{ennreal } (\text{indicator } \{0 < \cdot < \pi\} \vartheta / \pi))$

have r0: $0 < r$ using ab unfolding r-def by simp

have mg: $g \in \text{borel-measurable borel}$ unfolding g-def

by (fast intro: borel-measurable-continuous-onI continuous-intros)

have mgU: $g \in \text{measurable } U \text{ borel}$ using mg unfolding U-def by simp

have h: ($\lambda u :: \text{real}. |t - u|$) $\in \text{borel-measurable borel}$

by (force intro: borel-measurable-continuous-onI continuous-intros)

have mF: ($\lambda u :: \text{real}. \ln |t - u|$) $\in \text{borel-measurable borel}$

```

    using borel-measurable-ln[OF h] by simp
  have mFg:  $(\lambda \vartheta. \ln |t - g \vartheta|) \in \text{borel-measurable borel}$ 
    using measurable-compose[OF mg mF] by (simp add: o-def)
  have e1:  $\text{integrable } (\arcsine a b) (\lambda u. \ln |t - u|) =$ 
     $\text{integrable } U (\lambda \vartheta. \ln |t - g \vartheta|)$ 
    unfolding arcsine-distr[OF ab] g-def[symmetric] U-def[symmetric]
    using integrable-distr-eq[OF mgU mF]
  by blast
  have e2:  $\text{integrable } U (\lambda \vartheta. \ln |t - g \vartheta|) =$ 
     $\text{integrable lborel } (\lambda \vartheta. (\text{indicator } \{0 < .. < \pi\} \vartheta / \pi) *_{\mathbb{R}} \ln |t -$ 
     $g \vartheta|)$ 
    unfolding U-def using integrable-density[of  $\lambda \vartheta. \ln |t -$ 
     $g \vartheta|$  lborel  $\lambda \vartheta. \text{indicator } \{0 < .. < \pi\} \vartheta / \pi$ ] mFg
  by simp
  have fe:  $(\lambda \vartheta. (\text{indicator } \{0 < .. < \pi\} \vartheta / \pi) *_{\mathbb{R}} \ln |t -$ 
     $g \vartheta|) = (\lambda \vartheta. 1 / \pi * (\text{indicator } \{0 < .. < \pi\} \vartheta *_{\mathbb{R}} \ln |t - g \vartheta|))$ 
    by (simp add: fun-eq-iff)
  have e3:  $\text{integrable lborel } (\lambda \vartheta. (\text{indicator } \{0 < .. < \pi\} \vartheta /$ 
     $\pi) *_{\mathbb{R}} \ln |t - g \vartheta|) =$ 
     $\text{set-integrable lborel } \{0 < .. < \pi\} (\lambda \vartheta. \ln |t - g \vartheta|)$ 
    unfolding set-integrable-def fe
    using integrable-mult-left-iff[of lborel 1 /
     $\pi \lambda \vartheta. \text{indicator } \{0 < .. < \pi\} \vartheta *_{\mathbb{R}} \ln |t -$ 
     $g \vartheta|]$  pi-gt-zero
  by (simp add: divide-inverse mult.commute)
  have pw:  $\ln |t - g \vartheta| = \ln r + \ln |y - \cos \vartheta|$ 
    if nc:  $\cos \vartheta \neq y$  for  $\vartheta$ 
  proof -
    have e0:  $t - g \vartheta = r * (\cos \vartheta - y)$  using r0 unfolding y-def g-def m-def
    r-def
    by (simp add: field-simps)
    have e:  $|t - g \vartheta| = r * |y - \cos \vartheta|$  unfolding e0 using r0
    by (simp add: abs-mult abs-minus-commute)
    show ?thesis unfolding e using r0 nc ln-mult[of r  $|y - \cos \vartheta|$ ] by simp
  qed
  have ae:  $AE \vartheta \in \{0 < .. < \pi\} \text{ in lborel. } (\lambda \vartheta. \ln |t -$ 
     $g \vartheta|) \vartheta = (\lambda \vartheta :: \text{real. } \ln r + \ln |y -$ 
     $\cos \vartheta|) \vartheta$ 
    using AE-I'[OF cos-level-null[of y], of  $\lambda \vartheta. \vartheta \in$ 
     $\{0 < .. < \pi\} \longrightarrow (\lambda \vartheta. \ln |t - g \vartheta|) \vartheta =$ 
     $(\lambda \vartheta :: \text{real. } \ln r + \ln |y - \cos \vartheta|) \vartheta]$  pw by fast
  have mG:  $(\lambda \vartheta :: \text{real. } \ln r + \ln |y - \cos \vartheta|) \in$ 
     $\text{borel-measurable borel}$  using log-cos-meas[of y] by (auto intro: borel-measurable-add)
  have interval-lebesgue-integrable lborel (ereal 0) (ereal  $\pi$ )  $(\lambda \vartheta. \ln |y -$ 
     $\cos \vartheta|)$ 
    using interval-integrable-sub[OF - - - log-cos-integrable[of y], of 0  $\pi$ ] pi-gt-zero
  by simp
  hence i1:  $\text{set-integrable lborel } \{0 < .. < \pi\} (\lambda \vartheta. \ln |y - \cos \vartheta|)$ 

```



```

    using pi-gt-zero unfolding interval-lebesgue-integrable-def by simp
  have c1: set-integrable lborel {0 <..<pi} (λϑ::real. ln r)
    using interval-integral-const(1)[of 0 pi ln r] pi-gt-zero
    unfolding interval-lebesgue-integrable-def by simp
  have sG: set-integrable lborel {0 <..<pi} (λϑ::real. ln r + ln |y -
    cos ϑ|) using set-integral-add(1)[OF c1 i1] .
  have si: set-integrable lborel {0 <..<pi} (λϑ. ln |t - g ϑ|)
    using set-integrable-cong-AE[of λϑ. ln |t -
      g ϑ| lborel λϑ. ln r + ln |y -
      cos ϑ| {0 <..<pi}] mFg mG ae sG
    by simp
  show ?thesis using e1 e2 e3 si by simp
qed

```

```

lemma dens-ln-meas: (λu :: real. ln |t - u|) ∈ borel-measurable borel
proof -
  have h: (λu :: real. |t - u|) ∈ borel-measurable borel
    by (force intro: borel-measurable-continuous-onI continuous-intros)
  show ?thesis using borel-measurable-ln[OF h] by simp
qed

```

```

lemma dens-ln-integrable:
  assumes ab: a < b
  shows integrable lborel (λu. arcsine-dens a b u * ln |t - u|)
proof -
  have md: (λu. arcsine-dens a b u) ∈ borel-measurable borel
    unfolding arcsine-dens-def by measurable
  have nn: AE u in lborel. 0 ≤ arcsine-dens a b u by (simp add: arcsine-dens-def)
  show ?thesis using arcsine-integrable-ln[OF ab,
    of t] integrable-density[of λu. ln |t -
      u| lborel arcsine-dens a b] dens-ln-meas[of t] md nn
    unfolding arcsine-def
    by simp
qed

```

```

lemma dens-ln-integral:
  shows (∫ u. arcsine-dens a b u * ln |t - u| ∂lborel) =
    logPot (arcsine a b) t
proof -
  have md: (λu. arcsine-dens a b u) ∈ borel-measurable borel
    unfolding arcsine-dens-def by measurable
  have nn: AE u in lborel. 0 ≤ arcsine-dens a b u by (simp add: arcsine-dens-def)
  show ?thesis unfolding logPot-def arcsine-def
    using integral-density[of λu. ln |t -
      u| lborel arcsine-dens a b] dens-ln-meas[of t] md nn
    by simp
qed

```

(A.1): the potential of the unit arcsine measure.

```

theorem arcsine-potential:
  assumes  $a < b$ 
  shows  $\logPot\ (arcsine\ a\ b)\ t =$ 
     $(if\ a \leq t \wedge t \leq b\ then\ \ln\ ((b - a) / 4)$ 
       $else\ \ln\ ((|t - (a + b) / 2| + \sqrt{(t - a) * (t - b)}) / 2))$ 
proof -
  define  $m$  where  $m = (a + b) / 2$ 
  define  $r$  where  $r = (b - a) / 2$ 
  define  $y$  where  $y = (m - t) / r$ 
  have  $r0$ :  $0 < r$  using assms unfolding  $r$ -def by simp
  have  $\logPot\ (arcsine\ a\ b)\ t = 1 / \pi *$ 
     $set\text{-}lebesgue\text{-}integral\ lborel\ \{0 < .. < \pi\}\ (\lambda \vartheta. \ln |t - (m - r * \cos \vartheta)|)$ 
  unfolding  $\logPot$ -def  $m$ -def  $r$ -def by (rule integral-arcsine[OF assms]) measurable
  also have  $\dots = 1 / \pi * set\text{-}lebesgue\text{-}integral\ lborel\ \{0 < .. < \pi\}\ (\lambda \vartheta. \ln r +$ 
     $\ln |y - \cos \vartheta|)$ 
  proof -
  have  $pw$ :  $\ln |t - (m - r * \cos \vartheta)| = \ln r + \ln |y - \cos \vartheta|$ 
  if  $nc$ :  $\cos \vartheta \neq y$  for  $\vartheta$ 
  proof -
  have  $e$ :  $|t - (m - r * \cos \vartheta)| = r * |y - \cos \vartheta|$  using  $r0$ 
  unfolding  $y$ -def
  by (simp add: field-simps abs-mult abs-minus-commute)
  show ?thesis unfolding  $e$  using  $r0\ nc\ \ln\text{-}mult[of\ r\ |y - \cos \vartheta|]$  by simp
qed
  have  $ae$ :  $AE\ \vartheta \in \{0 < .. < \pi\}\ in\ lborel. (\lambda \vartheta :: real. \ln |t - (m -$ 
     $r * \cos \vartheta)|) \vartheta =$ 
     $(\lambda \vartheta :: real. \ln r + \ln |y - \cos \vartheta|) \vartheta$ 
  using AE-I'[OF cos-level-null[of y], of  $\lambda \vartheta. \vartheta \in$ 
     $\{0 < .. < \pi\} \longrightarrow (\lambda \vartheta :: real. \ln |t - (m - r * \cos \vartheta)|) \vartheta = (\lambda \vartheta :: real. \ln r + \ln |y - \cos \vartheta|) \vartheta]$   $pw$ 
  by fast
  have  $mf$ :  $(\lambda \vartheta :: real. \ln |t - (m - r * \cos \vartheta)|) \in$ 
    borel-measurable lborel
  proof -
  have  $h$ :  $(\lambda \vartheta :: real. |t - (m - r * \cos \vartheta)|) \in$ 
    borel-measurable borel
  by (force intro: borel-measurable-continuous-onI continuous-intros)
  show ?thesis using borel-measurable-ln[OF h] by simp
qed
  have  $mg$ :  $(\lambda \vartheta :: real. \ln r + \ln |y - \cos \vartheta|) \in$ 
    borel-measurable lborel using log-cos-meas[of y]
  by (auto intro: borel-measurable-add)
  show ?thesis using set-lebesgue-integral-cong-AE[OF - mf mg ae]
  by simp
qed
  also have  $\dots = 1 / \pi * (\pi * \ln r + \pi * (\ln (jouk\ y) - \ln 2))$ 

```

```

proof -
  have interval-lebesgue-integrable lborel (ereal 0) (ereal pi) ( $\lambda \vartheta. \ln |y - \cos \vartheta|$ )
  using interval-integrable-sub[OF - - log-cos-integrable[of y], of 0 pi] pi-gt-zero
  by simp
  hence i1: set-integrable lborel {0 <..pi} ( $\lambda \vartheta. \ln |y - \cos \vartheta|$ ) using pi-gt-zero unfolding interval-lebesgue-integrable-def
  by simp
  have v1: set-lebesgue-integral lborel {0 <..pi} ( $\lambda \vartheta. \ln |y - \cos \vartheta|$ ) = pi * (ln (jouk y) - ln 2)
  using log-cos-half[of y] interval-lebesgue-integral-le-eq[of ereal 0 ereal pi lborel ( $\lambda \vartheta. \ln |y - \cos \vartheta|$ )] pi-gt-zero
  by simp
  have c1: set-integrable lborel {0 <..pi} ( $\lambda \vartheta::\text{real}. \ln r$ )
  using interval-integral-const(1)[of 0 pi ln r] pi-gt-zero
  unfolding interval-lebesgue-integrable-def
  by simp
  have c2: set-lebesgue-integral lborel {0 <..pi} ( $\lambda \vartheta::\text{real}. \ln r$ ) = pi * ln r
  using interval-integral-const(2)[of 0 pi ln r] pi-gt-zero
  unfolding interval-lebesgue-integral-def
  by auto
  show ?thesis using set-integral-add(2)[OF c1 i1] c2 v1
  by simp
qed
also have ... = (if a ≤ t ∧
  t ≤ b then ln ((b - a) / 4) else ln ((|t - (a + b) / 2| + sqrt ((t - a) * (t - b))) / 2))
proof -
  have iff: |y| ≤ 1  $\longleftrightarrow$  a ≤ t ∧ t ≤ b using r0
  unfolding y-def m-def r-def
  by (auto simp: abs-le-iff field-simps)
  have e0: 1 / pi * (pi * ln r + pi * (ln (jouk y) - ln 2)) = ln r + ln (jouk y)
  - ln 2
  by (simp add: field-simps)
  have bm: (b - a) / 4 = r / 2 unfolding r-def by simp
  have p0: 0 < jouk y using jouk-ge-1[of y] by simp
  show ?thesis
proof (cases a ≤ t ∧ t ≤ b)
  case True
  hence jouk y = 1 using iff unfolding jouk-def by simp
  thus ?thesis unfolding e0 bm using True r0 ln-div[of r 2]
  by simp
next
  case False
  have y1: 1 < |y| using iff False by linarith
  have j: jouk y = |y| + sqrt (y ^ 2 - 1) using y1 unfolding jouk-def by
simp
  have q1: r * |y| = |t - (a + b) / 2| using r0 unfolding y-def m-def
  by (simp add: abs-divide abs-minus-commute)

```

```

    have q2b:  $r^2 * (y^2 - 1) = (m - t)^2 - r^2$  using r0 unfolding
y-def
    by (simp add: field-simps power2-eq-square)
    have q2c:  $(m - t)^2 - r^2 = (t - a) * (t - b)$  unfolding m-def r-def
    by (simp add: power2-eq-square algebra-simps field-simps)
    have q2a:  $r^2 * (y^2 - 1) = (t - a) * (t - b)$  using q2b q2c
    by simp
    have q2:  $r * \sqrt{y^2 - 1} = \sqrt{(t - a) * (t - b)}$ 
    using q2a r0 real-sqrt-mult[of  $r^2 y^2 - 1$ ]
    by simp
    have q3:  $(|t - (a + b) / 2| + \sqrt{(t - a) * (t - b)}) / 2 = r * \text{jouk } y / 2$ 
    using q1 q2 j
    by (metis distrib-left)
    have fin:  $\ln r + \ln (\text{jouk } y) - \ln 2 =$ 
     $\ln ((|t - (a + b) / 2| + \sqrt{(t - a) * (t - b)}) / 2)$ 
    unfolding q3 using r0 p0 ln-div[of  $r * \text{jouk } y / 2$ ] ln-mult[of  $r \text{jouk } y$ ]
    by simp
    show ?thesis unfolding e0 using False fin
    by presburger
  qed
qed
finally show ?thesis .
qed

```

```

lemma arcsine-dens-nonneg:  $0 \leq \text{arcsine-dens } a \ b \ t$ 
  by (simp add: arcsine-dens-def)

```

```

lemma arcsine-finite:  $a < b \implies \text{finite-measure } (\text{arcsine } a \ b)$ 
  using arcsine-univ[of  $a \ b$ ] unfolding arcsine-def
  by (fastforce intro: finite-measureI)

```

```

lemma arcsine-AE:  $\text{AE } x \text{ in arcsine } a \ b. \ a < x \wedge x < b$ 
proof -
  have md:  $(\lambda t. \text{ennreal } (\text{arcsine-dens } a \ b \ t)) \in \text{borel-measurable lborel}$ 
    unfolding arcsine-dens-def by measurable
  show ?thesis unfolding arcsine-def using AE-density[OF md]
    by (simp add: arcsine-dens-def)
qed

```

```

lemma abs-ln-bound:
  fixes z :: real
  assumes  $0 \leq z < 2$ 
  shows  $|\ln z| \leq 2 * \ln 2 - \ln z$ 
proof (cases  $z = 0$ )
  case True
  thus ?thesis
    by simp
next

```

```

case False
hence z0: 0 < z using assms by simp
have ln z ≤ ln 2 using z0 assms
  by simp
thus ?thesis using ln-gt-zero[of 2::real]
  by linarith
qed

lemma arcsine-swap:
  fixes a b c d L :: real
  assumes ab: a < b 0 < a b < 2 and cd: c < d 0 < c d < 2
    and bnd:  $\bigwedge x. 0 < x \implies x < 2 \implies$ 
      L ≤ logPot (arcsine c d) x
  shows integralL (arcsine a b) (logPot (arcsine c d)) =
    integralL (arcsine c d) (logPot (arcsine a b))
proof -
  have fm1: finite-measure (arcsine a b) by (rule arcsine-finite[OF ab(1)])
  have fm2: finite-measure (arcsine c d) by (rule arcsine-finite[OF cd(1)])
  have ps: pair-sigma-finite (arcsine a b) (arcsine c d) using fm1 fm2
    by (simp add: pair-sigma-finite-def finite-measure-def)
  have sf2: sigma-finite-measure (arcsine c d) using fm2 unfolding finite-measure-def
  by simp
  have sets1: sets (arcsine a b) = sets borel and sets2: sets (arcsine c d) = sets
    borel
    unfolding arcsine-def by simp-all
  have meas: (λp. ln |fst p - snd p|) ∈
    borel-measurable (arcsine a b  $\otimes_M$  arcsine c d)
    unfolding measurable-cong-sets[OF sets-pair-measure-cong[OF sets1 sets2] refl]
  by measurable
  have inner: (∫+ y. ennreal (norm (ln |x -
    y|)) ∂arcsine c d) ≤ ennreal (2 * ln 2 - L) if x: 0 < x x < 2 for x
  proof -
    have i1: (∫+ y. ennreal (norm (ln |x -
      y|)) ∂arcsine c d) ≤
      (∫+ y. ennreal (2 * ln 2 - ln |x - y|) ∂arcsine c d)
    proof (rule nn-integral-mono-AE)
      have |ln |x - y|| ≤ 2 * ln 2 - ln |x - y|
        if c < y ∧ y < d for y
        using abs-ln-bound[of |x - y|] that x cd
        by linarith
      thus AE y in arcsine c d. ennreal (norm (ln |x - y|)) ≤
        ennreal (2 * ln 2 - ln |x - y|)
        using arcsine-AE[of c d]
        by (auto intro: ennreal-leI elim: AE-mp)
    qed
    have ig: integrable (arcsine c d) (λy. 2 * ln 2 - ln |x - y|)
      using arcsine-integrable-ln[OF cd(1), of x] fm2
    by (fast intro: Bochner-Integration.integrable-diff finite-measure.integrable-const)
    have nn0: 0 ≤ 2 * ln 2 - ln |x - y| if c < y ∧ y < d for y

```

```

    using abs-ln-bound[of |x - y|] that x cd
    by linarith
  have nn: AE y in arcsine c d.  $0 \leq 2 * \ln 2 - \ln |x - y|$ 
    using arcsine-AE[of c d] nn0
    by (fast elim: AE-mp)
  have i2:  $(\int^+ y. \text{ennreal } (2 * \ln 2 - \ln |x - y|) \partial \text{arcsine } c \ d) = \text{ennreal } (\int y. 2 * \ln 2 - \ln |x - y| \partial \text{arcsine } c \ d)$ 
    using nn-integral-eq-integral[OF ig nn] .
  have mu:  $\text{measure } (\text{arcsine } c \ d) (\text{space } (\text{arcsine } c \ d)) = 1$  using arcsine-univ[OF cd(1)]
    unfolding arcsine-def
    by (simp add: measure-def)
  have i3:  $(\int y. 2 * \ln 2 - \ln |x - y| \partial \text{arcsine } c \ d) = 2 * \ln 2 - \log \text{Pot } (\text{arcsine } c \ d) \ x$ 
    using Bochner-Integration.integral-diff[of arcsine c d  $\lambda y. 2 * \ln 2 - \ln |x - y|$ ] arcsine-integrable-ln[OF cd(1), of x] finite-measure.integrable-const[OF fm2] mu
    unfolding logPot-def
    by fastforce
  show ?thesis using i1 i2 i3 bnd[OF x]
    by (fastforce intro: order-trans ennreal-leI)
qed
  have outer:  $(\int^+ x. (\int^+ y. \text{ennreal } (\text{norm } (\ln |x - y|) \partial \text{arcsine } c \ d) \partial \text{arcsine } a \ b) \leq (\int^+ x. \text{ennreal } (2 * \ln 2 - L) \partial \text{arcsine } a \ b)$ 
    using arcsine-AE[of a b] inner ab
    by (fastforce intro: nn-integral-mono-AE elim: AE-mp)
  have fin:  $(\int^+ x. \text{ennreal } (2 * \ln 2 - L) \partial \text{arcsine } a \ b) < \infty$ 
    using arcsine-univ[OF ab(1)] unfolding arcsine-def
    by simp
  have pr:  $(\int^+ x. (\int^+ y. \text{ennreal } (\text{norm } (\ln |x - y|) \partial \text{arcsine } c \ d) \partial \text{arcsine } a \ b) = (\int^+ p. \text{ennreal } (\text{norm } (\ln |fst \ p - snd \ p|) \partial (\text{arcsine } a \ b \otimes_M \text{arcsine } c \ d)))$ 
    using sigma-finite-measure.nn-integral-fst[OF sf2, of  $\lambda p. \text{ennreal } (\text{norm } (\ln |fst \ p - snd \ p|)) \text{arcsine } a \ b$ ] meas
    by simp
  have intg:  $\text{integrable } (\text{arcsine } a \ b \otimes_M \text{arcsine } c \ d) (\lambda(x, y). \ln |x - y|)$ 
    using integrable-iff-bounded[of arcsine a b  $\otimes_M$  arcsine c d  $\lambda p. \ln |fst \ p - snd \ p|$ ] meas outer fin pr
    by (simp add: case-prod-beta)
  have F:  $(\int y. (\int x. \ln |x - y| \partial \text{arcsine } a \ b) \partial \text{arcsine } c \ d) = (\int x. (\int y. \ln |x - y| \partial \text{arcsine } c \ d) \partial \text{arcsine } a \ b)$ 
    using pair-sigma-finite.Fubini-integral[OF ps, of  $\lambda x \ y. \ln |x - y|$ ] intg
    by simp

```

```

show ?thesis unfolding logPot-def using F
  by (simp add: abs-minus-commute)
qed

```

$\omega\text{-}\{[a,b]\}([x-r, x+r]) \leq 8\sqrt{r}/(\pi\sqrt{(b-a)})$: the density is dominated by $(\pi\sqrt{((b-a)/2)})^{-1} ((u-a)^{-1/2} + (b-u)^{-1/2})$.

```

lemma cos-gap:
  fixes s t :: real
  assumes st: 0 ≤ s ≤ t ≤ pi
  shows 2 * ((t - s) / pi) ^ 2 ≤ cos s - cos t
proof -
  define d where d = (t - s) / 2
  define y where y = (s + t) / 2
  have cd: cos s - cos t = 2 * sin y * sin d unfolding y-def d-def by (rule
cos-diff-cos)
  have d0: 0 ≤ d ≤ pi / 2 using st unfolding d-def by auto
  have sd: 2 * d / pi ≤ sin d by (rule sin-ge-two-over-pi[OF d0])
  have p2: 0 ≤ 2 * d / pi using d0 by simp
  have sy: sin d ≤ sin y
  proof (cases y ≤ pi / 2)
    case True
    thus ?thesis using sin-monotone-2pi-le[of d y] d0 st unfolding y-def d-def
      by simp
  next
    case False
    have sin d ≤ sin (pi - y)
    proof (rule sin-monotone-2pi-le)
      show - (pi / 2) ≤ d using d0 pi-gt-zero by linarith
      show d ≤ pi - y using st unfolding y-def d-def by argo
      show pi - y ≤ pi / 2 using False unfolding y-def by linarith
    qed
    thus ?thesis by simp
  qed
  have (2 * d / pi) * (2 * d / pi) ≤ sin d * sin y using sd sy p2
    by (fastforce intro: mult-mono)
  thus ?thesis unfolding cd d-def
    by (simp add: power2-eq-square field-simps)
qed

```

```

lemma diam-measure:
  fixes S :: real set and D :: real
  assumes ne: S ≠ {} and bdd: ∀ x ∈ S. 0 ≤ x and dm: ⋀ x y. x ∈
    S ⇒ y ∈ S ⇒ y - x ≤ D
  shows emeasure lborel S ≤ ennreal D
proof -
  define c where c = Inf S
  have b: bdd-below S using bdd
    by (fast intro: bdd-belowI)
  have sub: S ⊆ {c..c + D}

```

```

proof
  fix  $x$  assume  $x: x \in S$ 
  have  $c \leq x$  unfolding  $c\text{-def}$  using  $c\text{Inf-lower}[OF\ x\ b]$  .
  moreover have  $lb: x - D \leq y$  if  $y \in S$  for  $y$  using  $dm[OF\ that\ x]$ 
    by  $force$ 
  moreover have  $x - D \leq c$  unfolding  $c\text{-def}$  using  $c\text{Inf-greatest}[OF\ ne\ lb]$ 
    by  $blast$ 
  ultimately show  $x \in \{c..c + D\}$  by  $simp$ 
qed
obtain  $x0$  where  $x0 \in S$  using  $ne$  by  $blast$ 
hence  $D0: 0 \leq D$  using  $dm[of\ x0\ x0]$  by  $simp$ 
have  $emeasure\ lborel\ S \leq emeasure\ lborel\ \{c..c + D\}$  by  $(rule\ emeasure-mono[OF\ sub])\ simp$ 
also have  $\dots = ennreal\ D$  using  $D0$ 
  by  $simp$ 
finally show  $?thesis$  .
qed

lemma  $arcsine\text{-}Icc\text{-}le$ :
  assumes  $a < b$   $0 \leq r$ 
  shows  $emeasure\ (arcsine\ a\ b)\ \{x - r..x + r\} \leq ennreal\ (8 * sqrt\ r / (pi * sqrt\ (b - a)))$ 
proof -
  define  $R$  where  $R = (b - a) / 2$ 
  define  $g$  where  $g = (\lambda\vartheta. (a + b) / 2 - (b - a) / 2 * cos\ \vartheta)$ 
  define  $U$  where  $U = density\ lborel\ (\lambda\vartheta. ennreal\ (indicator\ \{0 <..<pi\}\ \vartheta / pi))$ 
  define  $S$  where  $S = \{\vartheta \in \{0 <..<pi\}. g\ \vartheta \in \{x - r..x + r\}\}$ 
  define  $l$  where  $l = 2 * r / R$ 
  have  $R0: 0 < R$  using  $assms$  unfolding  $R\text{-def}$  by  $simp$ 
  have  $mg: g \in borel\text{-measurable}\ borel$  unfolding  $g\text{-def}$ 
    by  $(fast\ intro: borel\text{-measurable}\text{-continuous}\text{-onI}\ continuous\text{-intros})$ 
  have  $mgU: g \in measurable\ U\ borel$  using  $mg$  unfolding  $U\text{-def}$  by  $simp$ 
  have  $Se: S = \{0 <..<pi\} \cap (g - \{x - r..x + r\} \cap space\ borel)$  unfolding  $S\text{-def}$ 
    by  $auto$ 
  have  $Sm: S \in sets\ lborel$  unfolding  $Se$  using  $measurable\text{-sets}[OF\ mg,\ of\ \{x - r..x + r\}]$ 
    by  $(force\ intro: sets.Int)$ 
  have  $e1: emeasure\ (arcsine\ a\ b)\ \{x - r..x + r\} =$ 
     $(\int^+ \vartheta. ennreal\ (indicator\ \{0 <..<pi\}\ \vartheta / pi) * indicator\ (g - \{x - r..x + r\})\ \vartheta\ \partial lborel)$ 
    unfolding  $arcsine\text{-distr}[OF\ assms(1)]\ g\text{-def}[symmetric]\ U\text{-def}[symmetric]$ 
    using  $emeasure\text{-distr}[OF\ mgU,\ of\ \{x - r..x + r\}]\ measurable\text{-sets}[OF\ mg,\ of\ \{x - r..x + r\}]$ 
    unfolding  $U\text{-def}$ 
    by  $(force\ simp: emeasure\text{-density})$ 
  have  $e2: (\int^+ \vartheta. ennreal\ (indicator\ \{0 <..<pi\}\ \vartheta / pi) * indicator\ (g - \{x - r..x + r\})\ \vartheta\ \partial lborel) =$ 
     $ennreal\ (1 / pi) * emeasure\ lborel\ S$ 

```



```

proof -
  have (λϑ. ennreal (indicator {0 <..<pi} ϑ / pi) * indicator (g - ' {x -
    r..x + r}) ϑ) = (λϑ. ennreal (1 / pi) * indicator S ϑ)
  unfolding S-def
  by (simp add: indicator-def fun-eq-iff)
  thus ?thesis using nn-integral-cmult-indicator[OF Sm, of ennreal (1 / pi)] by
simp
qed
have e3: emeasure lborel S ≤ ennreal (pi * sqrt (l / 2))
proof (cases S = {})
  case True
  thus ?thesis by simp
next
  case False
  have bdd: ∀ ϑ ∈ S. 0 ≤ ϑ unfolding S-def by auto
  have lp: 0 ≤ pi * sqrt (l / 2) unfolding l-def using R0 assms by simp
  have dm: ϑ2 - ϑ1 ≤ pi * sqrt (l / 2)
    if th: ϑ1 ∈ S ϑ2 ∈ S for ϑ1 ϑ2
  proof (cases ϑ2 ≤ ϑ1)
    case True
    thus ?thesis using lp by linarith
  next
    case False
    have t: 0 ≤ ϑ1 ϑ1 ≤ ϑ2 ϑ2 ≤ pi using th False
    unfolding S-def by auto
    have cg: 2 * ((ϑ2 - ϑ1) / pi) ^ 2 ≤ cos ϑ1 - cos ϑ2
      by (rule cos-gap[OF t])
    have gi: g ϑ2 - g ϑ1 ≤ 2 * r using th unfolding S-def by auto
    have gc: g ϑ2 - g ϑ1 = R * (cos ϑ1 - cos ϑ2)
      unfolding g-def R-def
      by (simp add: field-simps)
    have cl: cos ϑ1 - cos ϑ2 ≤ l using gi gc R0 unfolding l-def
      by (auto simp: pos-le-divide-eq mult.commute)
    have sq: ((ϑ2 - ϑ1) / pi) ^ 2 ≤ l / 2 using cg cl by linarith
    have (ϑ2 - ϑ1) / pi ≤ sqrt (l / 2) using real-le-rsqrt[OF sq] .
    thus ?thesis using pi-gt-zero
      by (fastforce simp: field-simps)
  qed
show ?thesis by (rule diam-measure[OF False bdd dm])
qed
have e4: ennreal (1 / pi) * ennreal (pi * sqrt (l / 2)) ≤
  ennreal (8 * sqrt r / (pi * sqrt (b - a)))
proof -
  have l0: 0 ≤ sqrt (l / 2) unfolding l-def using R0 assms by simp
  have q0: ennreal (1 / pi) * ennreal (pi * sqrt (l / 2)) = ennreal (sqrt (l / 2))
    using ennreal-mult[of 1 / pi pi * sqrt (l / 2)] pi-gt-zero l0
    by force
  have q1: l / 2 = 2 * r / (b - a) unfolding l-def R-def using assms by (simp
    add: field-simps)

```

```

have q2:  $\sqrt{2 * r / (b - a)} = \sqrt{2} * \sqrt{r} / \sqrt{b - a}$ 
  by (simp add: real-sqrt-mult real-sqrt-divide)
have q3:  $\sqrt{2} < 3 / 2$  using real-less-lsqrt[of 3 / 2 2]
  by (simp add: power2-eq-square)
have q4a:  $\sqrt{2} * \pi \leq 3 / 2 * 4$  using q3 pi-less-4 pi-gt-zero
  by (fastforce intro: mult-mono)
have q4:  $\sqrt{2} * \pi \leq 8$  using q4a
  by linarith
have k5:  $\sqrt{2} * \pi * \sqrt{r} \leq 8 * \sqrt{r}$  using q4 real-sqrt-ge-zero[OF
assms(2)]
  by (fast intro: mult-right-mono)
have ba:  $0 < \sqrt{b - a}$  using assms by simp
have q5:  $\sqrt{2} * \sqrt{r} / \sqrt{b - a} \leq 8 * \sqrt{r} / (\pi * \sqrt{b - a})$ 
  using k5 ba pi-gt-zero
  by (simp add: field-simps)
show ?thesis unfolding q0 unfolding q1 q2 using q5
  by (blast intro: ennreal-leI)
qed
have e5:  $\text{ennreal } (1 / \pi) * \text{emeasure lborel } S \leq$ 
   $\text{ennreal } (1 / \pi) * \text{ennreal } (\pi * \sqrt{l / 2})$ 
  using e3
  by (force intro: mult-left-mono)
show ?thesis unfolding e1 e2 using e5 e4
  by (fast intro: order-trans)
qed

lemma arcsine-Icc-le40:
  assumes  $a < b$   $1 / 225 < b - a$   $0 \leq r$ 
  shows  $\text{emeasure } (\text{arcsine } a \ b) \{x - r..x + r\} \leq \text{ennreal } (40 * \sqrt{r})$ 
proof -
  have s1:  $1 / 15 < \sqrt{b - a}$  using real-less-rsqrt[of 1 / 15 b - a] assms(2)
    by (simp add: power2-eq-square)
  have p0:  $3 * (1 / 15) \leq \pi * \sqrt{b - a}$  using s1 pi-gt3
    by (fastforce intro: mult-mono)
  have p:  $1 / 5 \leq \pi * \sqrt{b - a}$  using p0
    by simp
  have q:  $8 \leq 40 * (\pi * \sqrt{b - a})$  using p by linarith
  have k:  $8 * \sqrt{r} \leq 40 * \sqrt{r} * (\pi * \sqrt{b - a})$ 
    using mult-left-mono[OF q real-sqrt-ge-zero[OF assms(3)]]
    by (auto simp: algebra-simps)
  have pp:  $0 < \pi * \sqrt{b - a}$  using p by linarith
  have 8:  $8 * \sqrt{r} / (\pi * \sqrt{b - a}) \leq 40 * \sqrt{r}$  using k pp
    using pos-divide-le-eq by blast
  thus ?thesis using arcsine-Icc-le[OF assms(1) assms(3), of x] by (meson ennreal-leI order-trans)
qed
end

theory Logarithmic-Energy

```

imports *Laplace-Sin-Cot-Series*
begin

12 Logarithmic energy of measures on $\mathbb{C}$ and the zero-mass energy inequality

The mutual logarithmic energy $I(\mu, \nu) = \int \int \ln |z - w| d\mu(z) d\nu(w)$ of finite measures on $\mathbb{C}$, the two-dimensional Gaussian integral $\int_{\mathbb{C}} \exp(-b|v|^2) dv = \pi/b$, the non-negativity of the energy of Gaussian kernels, the truncated kernels $K_n \rightarrow \ln |z - w|$ obtained by integrating the Gaussian against ds/s (Frullani), and the main result *log-energy-zero-mass-nonpos*: for measures $\nu_1, \nu_2 \leq \eta$ of equal mass with $\ln |z - w|$ integrable on $\eta \otimes \eta$ and a null diagonal, $I(\nu_1) - 2 I(\nu_1, \nu_2) + I(\nu_2) \leq 0$.

Mutual logarithmic energy *integral integral* $\log|z - w| d\mu(z) d\nu(w)$ on $\mathbb{C}$.

definition *logEnergyC* :: *complex measure* $\Rightarrow$ *complex measure* $\Rightarrow$ *real* **where**
logEnergyC μ ν = *integral*^L μ (λz . *integral*^L ν (λw .
 $\ln (\text{norm } (z - w))))$

Lemma 6.2 (zero-mass logarithmic energy), in the corrected form of the Lean proof: for finite measures $\nu_1, \nu_2 \leq \eta$ on $\mathbb{C}$ of equal mass, with $\log|z-w|$ integrable for $\eta \otimes \eta$ and the diagonal $\eta \otimes \eta$ -null, $I(\nu_1) - 2 I(\nu_1, \nu_2) + I(\nu_2) \leq 0$. (The literal statement with $\log 0 = 0$ is false for point masses; the diagonal hypothesis is what the paper uses.) Proof: Gaussian kernels $e^{-\frac{1}{2}|z-w|^2}$ have nonnegative energy (Fubini), integrate in s against ds/s to reach $\log$ (dominated convergence).

The Gaussian integral on $\mathbb{C}$: $\int e^{-\frac{1}{2}|v|^2} dv = \pi/b$.

lemma *gauss-nn-C*:

fixes $b :: \text{real}$

assumes $b: 0 < b$

shows $(\int^+ v. \text{ennreal } (\exp (- b * (\text{norm } (v :: \text{complex})) ^ 2)) \partial \text{lborel}) =$
 $\text{ennreal } (\pi / b)$

proof –

define $h :: \text{real} \Rightarrow \text{real}$ **where** $h\ r = 2 * b * r * \exp (- b * r ^ 2)$ **for** r

have $hm: h \in \text{borel-measurable borel}$ **unfolding** *h-def*

by *simp*

have $p2: \text{filterlim } (\lambda r :: \text{real}. r ^ 2) \text{ at-top at-top}$

using *filterlim-pow-at-top*[*OF* - *filterlim-ident*, *of* 2]

by *simp*

have $l0: ((\lambda r :: \text{real}. \exp (- (b * r ^ 2))) \longrightarrow 0) \text{ at-top}$

using *filterlim-compose*[*OF* *exp-neg-lim*[*OF* b] $p2$]

by *simp*

have $l1: ((\lambda r :: \text{real}. \exp (- (b * r ^ 2)) * (\sum k \leq$
 $1. (b * r ^ 2) ^ k / \text{fact } k)) \longrightarrow 0) \text{ at-top}$

using *filterlim-compose*[*OF* *gam-lim*[*OF* b , *of* 1] $p2$]

```

  by simp
have lim1: ((λr :: real. - exp (- b * r ^ 2)) → 0) at-top
  using tendsto-minus[OF l0]
  by simp
have step1: ennreal (exp (- b * (norm v) ^ 2)) =
  (∫+ r. ennreal (h r) * indicator {norm v..} r ∂lborel)
  for v :: complex
proof -
  have D: DERIV (λr :: real. - exp (- b * r ^ 2)) r :> h r for r unfolding
h-def
  by (intro derivative-eq-intros, (rule refl)+) (simp add: algebra-simps power2-eq-square)
  have nn: 0 ≤ h r if norm v ≤ r for r
  proof -
    have r0: 0 ≤ r using that norm-ge-zero[of v]
    by linarith
    show ?thesis unfolding h-def using r0 b
    using mult-nonneg-nonneg by simp
  qed
  have (∫+ r. ennreal (h r) * indicator {norm v..} r ∂lborel) =
    ennreal (0 - (- exp (- b * (norm v) ^ 2)))
    using nn-integral-FTC-atLeast[OF hm D nn lim1] .
  thus ?thesis
  by simp
qed
have ps: pair-sigma-finite (lborel :: complex measure) (lborel :: real measure)
  by (simp add: pair-sigma-finite-def sigma-finite-lborel)
have jm: (λ(v :: complex, r :: real). ennreal (h r) * indicator {norm v..} r) ∈
  borel-measurable (lborel ⊗M lborel)
proof -
  have e: (λ(v :: complex, r :: real). ennreal (h r) * indicator {norm v..} r) =
    (λp. ennreal (h (snd p)) * indicator {p. norm (fst p) ≤ snd p} p)
    by (simp add: fun-eq-iff indicator-def)
  have mf2: (λp :: complex × real. snd p - norm (fst p)) ∈
    borel-measurable (lborel ⊗M lborel)
    by simp
  have eq: {p :: complex × real. norm (fst p) ≤ snd p} =
    (λp. snd p - norm (fst p)) - ' {0..} ∩ space (lborel ⊗M lborel)
    by (auto simp: space-pair-measure)
  have S: {p :: complex × real. norm (fst p) ≤ snd p} ∈
    sets (lborel ⊗M lborel) unfolding eq using measurable-sets[OF mf2, of {0..}]
    by simp
  show ?thesis unfolding e using S hm
  by simp
qed
have inner: (∫+ v. ennreal (h r) *
  indicator {norm v..} r ∂(lborel :: complex measure)) =
  ennreal (pi * h r * r ^ 2) * indicator {0..} r for r :: real
proof (cases 0 ≤ r)
  case True

```

```

have hn:  $0 \leq h \ r$  unfolding h-def using True b
  by simp
have ind:  $(\lambda v :: \text{complex}. \text{ennreal } (h \ r) * \text{indicator } \{\text{norm } v..\} \ r) =$ 
   $(\lambda v. \text{ennreal } (h \ r) * \text{indicator } (\text{cball } 0 \ r) \ v)$ 
  by (simp add: fun-eq-iff indicator-def)
have i1:  $(\int^+ v. \text{ennreal } (h \ r) * \text{indicator } (\text{cball } 0 \ r) \ v \ \partial(\text{lborel} :: \text{complex measure})) =$ 
   $\text{ennreal } (h \ r) * \text{emeasure lborel } (\text{cball } (0 :: \text{complex}) \ r)$ 
  using nn-integral-cmult-indicator[of cball (0 :: complex) r lborel ennreal (h
r)] by simp
have i2:  $\text{emeasure lborel } (\text{cball } (0 :: \text{complex}) \ r) = \text{ennreal } (\pi * r^2)$ 
  using emeasure-cball[where c = 0 :: complex and r = r] True
  by (simp add: unit-ball-vol-2)
show ?thesis unfolding ind i1 i2 using True hn
  by (simp add: ennreal-mult[symmetric] mult-ac)
next
case False
have  $(\lambda v :: \text{complex}. \text{ennreal } (h \ r) * \text{indicator } \{\text{norm } v..\} \ r) = (\lambda v. 0)$ 
proof -
  have  $\neg \text{norm } (v :: \text{complex}) \leq r$  for v using False norm-ge-zero[of v]
    by linarith
  thus ?thesis
    by (simp add: fun-eq-iff indicator-def)
qed
thus ?thesis using False
  by simp
qed
have  $(\int^+ v. \text{ennreal } (\exp (- b * (\text{norm } (v :: \text{complex}))^2)) \ \partial \text{lborel}) =$ 
   $(\int^+ v. (\int^+ r. \text{ennreal } (h \ r) * \text{indicator } \{\text{norm } (v :: \text{complex})..\} \ r \ \partial \text{lborel}) \ \partial \text{lborel})$ 
  unfolding step1 ..
also have ... =  $(\int^+ r. (\int^+ v. \text{ennreal } (h \ r) * \text{indicator } \{\text{norm } (v :: \text{complex})..\} \ r \ \partial \text{lborel}) \ \partial \text{lborel})$ 
  using pair-sigma-finite.Fubini'[OF ps jm]
  by (simp add: split-beta)
also have ... =  $(\int^+ r. \text{ennreal } (\pi * h \ r * r^2) * \text{indicator } \{0..\} \ r \ \partial \text{lborel})$  unfolding inner ..
also have ... =  $\text{ennreal } (\pi / b)$ 
proof -
  define F :: real  $\Rightarrow$  real where F r =  $-\pi * (r^2 + 1 / b) * \exp (- b * r^2)$ 
  for r
  have gm:  $(\lambda r. \pi * h \ r * r^2) \in \text{borel-measurable borel}$  using hm
    by simp
  have D2: DERIV F r :>  $\pi * h \ r * r^2$  for r unfolding F-def h-def
    by (intro derivative-eq-intros, (rule refl)+) (use b in <simp add: algebra-simps
power2-eq-square>)
  have nn2:  $0 \leq \pi * h \ r * r^2$  if  $0 \leq r$  for r unfolding h-def using that b
    by simp

```

```

have e: (λr. - pi * (r ^ 2 + 1 / b) * exp (- b * r ^ 2)) =
  (λr. (- pi / b) * (exp (- (b * r ^ 2)) * (∑ k ≤
    1. (b * r ^ 2) ^ k / fact k)))
using b by (simp add: fun-eq-iff algebra-simps)
have lim2: (F ⟶ 0) at-top unfolding F-def e
  using tendsto-mult-left[OF l1, of - pi / b]
  by simp
have (∫+ r. ennreal (pi * h r * r ^ 2) * indicator {0..} r ∂lborel) =
  ennreal (0 - F 0)
  using nn-integral-FTC-atLeast[OF gm D2 nn2 lim2] .
also have 0 - F 0 = pi / b unfolding F-def
  by simp
finally show ?thesis .
qed
finally show ?thesis .
qed

```

```

lemma gauss-integral-C:
  assumes 0 < b
  shows integralL lborel (λv :: complex. exp (- b * (norm v) ^ 2)) = pi / b
proof -
  have m: (λv :: complex. exp (- b * (norm v) ^ 2)) ∈ borel-measurable lborel
  by simp
  have integralL lborel (λv :: complex. exp (- b * (norm v) ^ 2)) =
    enn2real (∫+ v. ennreal (exp (- b *
      (norm (v :: complex)) ^ 2)) ∂lborel)
  using integral-eq-nn-integral[OF m]
  by simp
  also have ... = pi / b unfolding gauss-nn-C[OF assms] using assms
  by (simp add: enn2real-ennreal)
  finally show ?thesis .
qed

```

Convolution of two Gaussians: $\int e^{\{-2s|z-u|^2\}} e^{\{-2s|w-u|^2\}} du = \pi/(4s) e^{\{-s|z-w|^2\}}$.

```

lemma gauss-conv:
  assumes 0 < s
  shows integralL lborel (λu :: complex. exp (- 2 * s * (norm (z - u)) ^ 2) *
    exp (- 2 * s * (norm (w - u)) ^ 2)) = pi / (4 * s) * exp (- s * (norm (z -
    w)) ^ 2)
proof -
  define m where m = (z + w) / 2
  have par: 2 * (norm (z - u)) ^ 2 + 2 * (norm (w - u)) ^ 2 =
    4 * (norm (u - m)) ^ 2 + (norm (z - w)) ^ 2 for u
  unfolding m-def cmod-power2
  by (simp add: power2-eq-square field-simps)
  have pt: exp (- 2 * s * (norm (z - u)) ^ 2) * exp (- 2 * s * (norm (w - u))
    ^ 2) =
    exp (- s * (norm (z - w)) ^ 2) * exp (- (4 * s) * (norm (u - m)) ^ 2) for u

```

```

proof -
  have - 2 * s * (norm (z - u)) ^ 2 + - 2 * s * (norm (w - u)) ^ 2 =
    - s * (norm (z - w)) ^ 2 + - (4 * s) * (norm (u - m)) ^ 2
  using arg-cong[OF par[of u], of λx. - s * x]
  by (simp add: algebra-simps)
  thus ?thesis
  by (simp add: mult-exp-exp)
qed
have tr: integralL lborel (λu :: complex. exp (- (4 * s) * (norm (u - m)) ^ 2))
=
  integralL lborel (λu :: complex. exp (- (4 * s) * (norm u) ^ 2))
proof -
  have mf: (λu :: complex. exp (- (4 * s) * (norm (u - m)) ^ 2)) ∈
    borel-measurable borel
  by simp
  have integralL lborel (λu :: complex. exp (- (4 * s) * (norm (u - m)) ^ 2)) =
    integralL (distr lborel borel ((+) m)) (λu :: complex. exp (- (4 * s) *
      (norm (u - m)) ^ 2))
  using lborel-distr-plus[of m]
  by simp
  also have ... = integralL lborel (λu :: complex. exp (- (4 * s) * (norm (m +
    u - m)) ^ 2))
  using integral-distr[OF - mf, of (+) m lborel]
  by simp
  finally show ?thesis
  by simp
qed
have s4: 0 < 4 * s using assms
  by simp
  have integralL lborel (λu :: complex. exp (- 2 * s * (norm (z - u)) ^ 2) *
    exp (- 2 * s * (norm (w - u)) ^ 2))
    = integralL lborel (λu :: complex. exp (- s * (norm (z - w)) ^ 2) * exp (-
      (4 *
        s) * (norm (u - m)) ^ 2))
  unfolding pt ..
  also have ... = exp (- s * (norm (z - w)) ^ 2) *
    integralL lborel (λu :: complex. exp (- (4 * s) * (norm (u - m)) ^ 2))
  by simp
  also have ... = exp (- s * (norm (z - w)) ^ 2) * (pi / (4 * s))
  unfolding tr gauss-integral-C[OF s4] ..
  finally show ?thesis
  by (simp add: mult-ac)
qed

definition gPhi :: real ⇒ complex measure ⇒ complex ⇒ real
where
  gPhi s μ u = integralL μ (λz. exp (- 2 * s * (norm (z - u)) ^ 2))

definition Eform :: complex measure ⇒ complex measure ⇒ (complex ×

```

$\text{complex} \Rightarrow \text{real} \Rightarrow \text{real}$ **where**
 $E\text{form } \nu 1 \ \nu 2 \ k = \text{integral}^L (\nu 1 \otimes_M \nu 1) \ k - 2 * \text{integral}^L (\nu 1 \otimes_M \nu 2) \ k + \text{integral}^L (\nu 2 \otimes_M \nu 2) \ k$

lemma *gPhi-nn*:
fixes $\mu :: \text{complex measure}$
assumes *fm*: *finite-measure* μ **and** *sm*: *sets* $\mu = \text{sets borel}$ **and** *s*: $0 \leq s$
shows $\text{ennreal } (g\text{Phi } s \ \mu \ u) = (\int^+ z. \text{ennreal } (\exp (- 2 * s * (\text{norm } (z - u)) ^ 2)) \ \partial \mu)$
proof –
have *m*: $(\lambda z :: \text{complex}. \exp (- 2 * s * (\text{norm } (z - u)) ^ 2)) \in \text{borel-measurable } \mu$ **unfolding** *measurable-cong-sets*[*OF sm refl*]
by *simp*
have $\text{norm } (\exp (- 2 * s * (\text{norm } (z - u)) ^ 2)) \leq 1$ **for** $z :: \text{complex}$ **using** *s*
by *simp*
hence *b*: $\text{AE } z \text{ in } \mu. \text{norm } (\exp (- 2 * s * (\text{norm } (z - u)) ^ 2)) \leq 1$
by *simp*
have *i*: $\text{integrable } \mu (\lambda z :: \text{complex}. \exp (- 2 * s * (\text{norm } (z - u)) ^ 2))$
using *finite-measure.integrable-const-bound*[*OF fm b m*] .
show *?thesis* **unfolding** *gPhi-def* **using** *nn-integral-eq-integral*[*OF i*]
by *simp*
qed

lemma *gPhi-nonneg*: $0 \leq g\text{Phi } s \ \mu \ u$
unfolding *gPhi-def*
by *simp*

lemma *gPhi-meas*:
fixes $\mu :: \text{complex measure}$
assumes *fm*: *finite-measure* μ **and** *sm*: *sets* $\mu = \text{sets borel}$
shows $g\text{Phi } s \ \mu \in \text{borel-measurable borel}$
proof –
have *sf*: *sigma-finite-measure* μ **using** *fm*
by (*simp add: finite-measure-def*)
have *j*: $(\lambda(u, z). \exp (- 2 * s * (\text{norm } (z - u)) ^ 2)) \in \text{borel-measurable } (\text{borel} \otimes_M \mu)$
unfolding *measurable-cong-sets*[*OF sets-pair-measure-cong*[*OF refl sm*] *refl*]
by *simp*
show *?thesis* **unfolding** *gPhi-def*[*abs-def*]
using *sigma-finite-measure.borel-measurable-lebesgue-integral*[*OF sf*,
of $\lambda u \ z. \exp (- 2 * s * (\text{norm } (z - u)) ^ 2) \text{ borel}$] *j*
by *simp*
qed

lemma *par-exp*:
fixes $z \ w \ u :: \text{complex}$
shows $\exp (- 2 * s * (\text{norm } (z - u)) ^ 2) * \exp (- 2 * s * (\text{norm } (w - u)) ^ 2) =$

$\exp(-s * (\text{norm } (z - w)) \wedge 2) * \exp(-(4 * s) * (\text{norm } (u - (z + w) / 2)) \wedge 2)$

proof –

have $\text{par}: 2 * (\text{norm } (z - u)) \wedge 2 + 2 * (\text{norm } (w - u)) \wedge 2 =$
 $4 * (\text{norm } (u - (z + w) / 2)) \wedge 2 + (\text{norm } (z - w)) \wedge 2$
unfolding *cmod-power2*
by (*simp add: power2-eq-square field-simps*)
have $- 2 * s * (\text{norm } (z - u)) \wedge 2 + - 2 * s * (\text{norm } (w - u)) \wedge 2 =$
 $- s * (\text{norm } (z - w)) \wedge 2 + - (4 * s) * (\text{norm } (u - (z + w) / 2)) \wedge 2$
using *arg-cong[OF par, of $\lambda x. - s * x$]*
by (*simp add: algebra-simps*)
thus *?thesis*
by (*simp add: mult-exp-exp*)

qed

lemma *gauss-shift-nn*:

fixes $b :: \text{real}$ **and** $m :: \text{complex}$

assumes $b: 0 < b$

shows $(\int^+ u. \text{ennreal } (\exp(-b * (\text{norm } (u - m)) \wedge 2)) \partial \text{lborel}) =$
 $\text{ennreal } (\pi / b)$

proof –

have $\text{mf}: (\lambda u :: \text{complex}. \text{ennreal } (\exp(-b * (\text{norm } (u - m)) \wedge 2))) \in$
 $\text{borel-measurable borel}$
by *simp*
have $(\int^+ u. \text{ennreal } (\exp(-b * (\text{norm } (u - m)) \wedge 2)) \partial \text{lborel}) =$
 $(\int^+ u. \text{ennreal } (\exp(-b * (\text{norm } (u - m)) \wedge 2)) \partial (\text{distr lborel borel } ((+) m)))$
using *lborel-distr-plus[of m]*
by *simp*
also have $\dots = (\int^+ u. \text{ennreal } (\exp(-b * (\text{norm } (m + u - m)) \wedge 2)) \partial \text{lborel})$
using *nn-integral-distr[of (+) m lborel borel $\lambda u. \text{ennreal } (\exp(-b * (\text{norm } (u - m)) \wedge 2))]$ mf*
by *simp*
also have $\dots = \text{ennreal } (\pi / b)$ **using** *gauss-nn-C[OF b]*
by *simp*
finally show *?thesis* .

qed

lemma *conv-nn*:

fixes $z w :: \text{complex}$

assumes $s: 0 < s$

shows $(\int^+ u. \text{ennreal } (\exp(-2 * s * (\text{norm } (z - u)) \wedge 2) * \exp(-2 * s * (\text{norm } (w - u)) \wedge 2)) \partial \text{lborel}) =$
 $\text{ennreal } (\pi / (4 * s) * \exp(-s * (\text{norm } (z - w)) \wedge 2))$

proof –

have $s_4: 0 < 4 * s$ **using** s
by *simp*

```

have (∫+ u. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2) * exp (- 2 * s *
  (norm (w - u)) ^ 2)) ∂lborel)
  = (∫+ u. ennreal (exp (- s * (norm (z - w)) ^ 2)) * ennreal (exp (- (4 *
s) *
  (norm (u - (z + w) / 2)) ^ 2)) ∂lborel)
unfolding par-exp
by (simp add: ennreal-mult)
also have ... = ennreal (exp (- s * (norm (z - w)) ^ 2)) *
  (∫+ u. ennreal (exp (- (4 * s) * (norm (u - (z + w) /
  2)) ^ 2)) ∂lborel)
by (simp add: nn-integral-cmult)
also have ... = ennreal (exp (- s * (norm (z - w)) ^ 2)) * ennreal (pi / (4 *
s))
unfolding gauss-shift-nn[OF s4] ..
also have ... = ennreal (pi / (4 * s) * exp (- s * (norm (z - w)) ^ 2)) using
s
by (simp add: ennreal-mult[symmetric] mult-ac)
finally show ?thesis .
qed

```

Gaussian kernels have nonnegative energy: $E(e^{\{-s|p-1 - p-2|^2\}}) = (4s/\pi) \int (\Phi-1 - \Phi-2)^2 \geq 0$.

lemma *gauss-pair-nn*:

```

fixes μ ν :: complex measure
assumes fm: finite-measure μ and fn: finite-measure ν and sm: sets μ =
  sets borel and sn: sets ν = sets borel and s: 0 < s
shows (∫+ u. ennreal (gPhi s μ u * gPhi s ν u) ∂lborel) =
  ennreal (pi / (4 * s)) * (∫+ z. ∫+ w. ennreal (exp (- s *
  (norm (z - w)) ^ 2)) ∂ν ∂μ)
proof -
have s0: 0 ≤ s using s
by simp
have sfm: sigma-finite-measure μ using fm
by (simp add: finite-measure-def)
have sfn: sigma-finite-measure ν using fn
by (simp add: finite-measure-def)
have p1: pair-sigma-finite μ (lborel :: complex measure) using sfm
by (simp add: pair-sigma-finite-def sigma-finite-lborel)
have p2: pair-sigma-finite ν (lborel :: complex measure) using sfn
by (simp add: pair-sigma-finite-def sigma-finite-lborel)
have S1: ennreal (gPhi s μ u * gPhi s ν u) =
  (∫+ z. ∫+ w. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) *
  ennreal (exp (- 2 * s * (norm (w - u)) ^ 2)) ∂ν ∂μ) for u :: complex
proof -
have mf: (λz. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2))) ∈
  borel-measurable μ unfolding measurable-cong-sets[OF sm refl]
by simp
have mg: (λw. ennreal (exp (- 2 * s * (norm (w - u)) ^ 2))) ∈
  borel-measurable ν unfolding measurable-cong-sets[OF sn refl]

```

```

    by simp
  have ennreal (gPhi s μ u * gPhi s ν u) =
    ennreal (gPhi s μ u) * ennreal (gPhi s ν u) using gPhi-nonneg
  by (simp add: ennreal-mult)
  also have ... = (∫+ z. ennreal (exp (- 2 * s * (norm (z -
    u)) ^ 2)) ∂μ) * (∫+ w. ennreal (exp (- 2 * s * (norm (w -
    u)) ^ 2)) ∂ν)
  unfolding gPhi-nn[OF fm sm s0] gPhi-nn[OF fn sn s0] ..
  also have ... = (∫+ z. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) *
    (∫+ w. ennreal (exp (- 2 * s * (norm (w -
    u)) ^ 2)) ∂ν) ∂μ)
  using nn-integral-multc[OF mf, of ∫+ w. ennreal (exp (- 2 * s * (norm (w
    -
    u)) ^ 2)) ∂ν]
  by simp
  also have ... = (∫+ z. ∫+ w. ennreal (exp (- 2 * s *
    (norm (z - u)) ^ 2)) * ennreal (exp (- 2 * s * (norm (w -
    u)) ^ 2)) ∂ν ∂μ)
  using nn-integral-cmult[OF mg]
  by simp
  finally show ?thesis .
qed
have m1: (λ(z, u). ∫+ w. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) *
  ennreal (exp (- 2 * s * (norm (w - u)) ^ 2)) ∂ν) ∈
  borel-measurable (μ ⊗M lborel)
proof -
  have jm0: (λp :: (complex × complex) × complex. ennreal (exp (- 2 * s *
    (norm (fst (fst p) - snd (fst p))) ^ 2)) * ennreal (exp (- 2 * s * (norm (snd
    p -
    snd (fst p))) ^ 2))) ∈ borel-measurable ((borel ⊗M lborel)
    ⊗M borel)
  by simp
  have e: (λ((z :: complex, u :: complex),
    w :: complex). ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) * ennreal (exp
    (- 2 * s *
    (norm (w - u)) ^ 2))) = (λp. ennreal (exp (- 2 * s * (norm (fst (fst p) -
    snd (fst p))) ^ 2)) * ennreal (exp (- 2 * s * (norm (snd p - snd (fst p))) ^
    2)))
  by (simp add: fun-eq-iff)
  have jm: (λ((z, u), w). ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) *
    ennreal (exp (- 2 * s * (norm (w - u)) ^ 2))) ∈
    borel-measurable ((μ ⊗M lborel) ⊗M ν)
  unfolding measurable-cong-sets[OF sets-pair-measure-cong[OF sets-pair-measure-cong[OF
    sm refl]
    sn] refl]
  unfolding e using jm0 .
  show ?thesis using sigma-finite-measure.borel-measurable-nn-integral[OF sfn,
    of λ(z,
    u) w. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) * ennreal (exp (- 2 * s

```

```

* (norm (w -
  u)) ^ 2)) μ ⊗M lborel] jm
  by (simp add: split-beta case-prod-beta)
qed
have m2: (λ(w, u). ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) * ennreal (exp
(- 2 * s *
  (norm (w - u)) ^ 2))) ∈ borel-measurable (ν ⊗M lborel) for z :: complex
  unfolding measurable-cong-sets[OF sets-pair-measure-cong[OF sn refl] refl]
  by simp
have (∫+ u. ennreal (gPhi s μ u * gPhi s ν u) ∂lborel) =
  (∫+ u. ∫+ z. ∫+ w. ennreal (exp (- 2 * s *
  (norm (z - u)) ^ 2)) * ennreal (exp (- 2 * s * (norm (w -
  u)) ^ 2)) ∂ν ∂μ ∂lborel)
  unfolding S1 ..
also have ... = (∫+ z. ∫+ u. ∫+ w. ennreal
  (exp (- 2 * s * (norm (z - u)) ^ 2)) * ennreal (exp (- 2 * s * (norm (w -
  u)) ^ 2)) ∂ν ∂lborel ∂μ)
  using pair-sigma-finite.Fubini'[OF p1 m1]
  by simp
also have ... = (∫+ z. ∫+ w. ∫+ u. ennreal
  (exp (- 2 * s * (norm (z - u)) ^ 2)) * ennreal (exp (- 2 * s * (norm (w -
  u)) ^ 2)) ∂lborel ∂ν ∂μ)
proof (rule nn-integral-cong)
  fix z assume z ∈ space μ
  show (∫+ u. ∫+ w. ennreal (exp (- 2 * s * (norm (z -
  u)) ^ 2)) * ennreal (exp (- 2 * s * (norm (w - u)) ^ 2)) ∂ν ∂lborel) =
    (∫+ w. ∫+ u. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) *
    ennreal (exp (- 2 * s * (norm (w - u)) ^ 2)) ∂lborel ∂ν)
    using pair-sigma-finite.Fubini'[OF p2 m2[of z]]
    by simp
qed
also have ... = (∫+ z. ∫+ w. ennreal (pi / (4 * s)) *
  ennreal (exp (- s * (norm (z - w)) ^ 2)) ∂ν ∂μ)
proof -
  have c: (∫+ u. ennreal (exp (- 2 * s * (norm (z - u)) ^ 2)) *
    ennreal (exp (- 2 * s * (norm (w - u)) ^ 2)) ∂lborel) =
    ennreal (pi / (4 * s)) * ennreal (exp (- s * (norm (z - w)) ^ 2)) for z w ::
complex
    using conv-nn[OF s, of z w] s
    by (simp add: ennreal-mult[symmetric])
  show ?thesis unfolding c ..
qed
also have ... = ennreal (pi / (4 * s)) *
  (∫+ z. ∫+ w. ennreal (exp (- s * (norm (z -
  w)) ^ 2)) ∂ν ∂μ)
proof -
  have in1: (∫+ w. ennreal (pi / (4 * s)) * ennreal (exp (- s * (norm (z -
  w)) ^ 2)) ∂ν) = ennreal (pi / (4 * s)) *
    (∫+ w. ennreal (exp (- s * (norm (z - w)) ^ 2)) ∂ν)

```

```

for  $z :: \text{complex}$ 
using  $\text{nn-integral-cmult}$ [ $\text{of } \lambda w. \text{ennreal } (\exp (-s * (\text{norm } (z - w))) ^ 2)) \nu \text{ennreal } (\pi / (4 * s))$ ]] unfolding  $\text{measurable-cong-sets}$ [ $\text{OF } \text{sn}$ 
 $\text{refl}$ ]
by  $\text{simp}$ 
have  $\text{jm}: (\lambda(z, w). \text{ennreal } (\exp (-s * (\text{norm } (z - w))) ^ 2)) \in$ 
 $\text{borel-measurable } (\mu \otimes_M \nu)$ 
unfolding  $\text{measurable-cong-sets}$ [ $\text{OF } \text{sets-pair-measure-cong}$ [ $\text{OF } \text{sm sn}$ ]  $\text{refl}$ ]
by  $\text{simp}$ 
have  $\text{mz}: (\lambda z. \int^+ w. \text{ennreal } (\exp (-s * (\text{norm } (z - w))) ^ 2)) \partial \nu \in \text{borel-measurable } \mu$ 
using  $\text{sigma-finite-measure.borel-measurable-nn-integral}$ [ $\text{OF } \text{sfn jm}$ ] .
show  $?thesis$  unfolding  $\text{in1}$  using  $\text{nn-integral-cmult}$ [ $\text{OF } \text{mz}, \text{of } \text{ennreal } (\pi /$ 
 $(4 * s))$ ]] .
qed
finally show  $?thesis$  .
qed

```

lemma gauss-pair :

```

assumes  $\text{finite-measure } \mu \text{ finite-measure } \nu \text{ sets } \mu = \text{sets borel sets } \nu =$ 
 $\text{sets borel } 0 < s$ 
shows  $\text{integral}^L (\mu \otimes_M \nu) (\lambda p. \exp (-s * (\text{norm } (\text{fst } p - \text{snd } p))) ^ 2)) = 4 * s / \pi * \text{integral}^L \text{lborel } (\lambda u. gPhi s \mu u * gPhi s \nu u)$ 
proof -
have  $\text{sfn}: \text{sigma-finite-measure } \nu$  using  $\text{assms}(2)$ 
by  $(\text{simp add: finite-measure-def})$ 
have  $\text{gm}: (\lambda p :: \text{complex} \times \text{complex}. \exp (-s * (\text{norm } (\text{fst } p - \text{snd } p))) ^ 2)) \in$ 
 $\text{borel-measurable } (\mu \otimes_M \nu)$ 
unfolding  $\text{measurable-cong-sets}$ [ $\text{OF } \text{sets-pair-measure-cong}$ [ $\text{OF } \text{assms}(3,4)$ ]
 $\text{refl}$ ]
by  $\text{simp}$ 
have  $X: (\int^+ p. \text{ennreal } (\exp (-s * (\text{norm } (\text{fst } p - \text{snd } p))) ^ 2)) \partial(\mu \otimes_M \nu) =$ 
 $(\int^+ z. \int^+ w. \text{ennreal } (\exp (-s * (\text{norm } (z - w))) ^ 2)) \partial \nu \partial \mu$ 
using  $\text{sigma-finite-measure.nn-integral-fst}$ [ $\text{OF } \text{sfn},$ 
 $\text{of } \lambda p. \text{ennreal } (\exp (-s * (\text{norm } (\text{fst } p - \text{snd } p))) ^ 2)) \mu$ ]  $\text{gm}$ 
by  $\text{simp}$ 
have  $L: \text{integral}^L (\mu \otimes_M \nu) (\lambda p. \exp (-s * (\text{norm } (\text{fst } p - \text{snd } p))) ^ 2)) = \text{enn2real } (\int^+ z. \int^+ w. \text{ennreal } (\exp (-s * (\text{norm } (z - w))) ^ 2)) \partial \nu \partial \mu$ 
unfolding  $X[\text{symmetric}]$  using  $\text{integral-eq-nn-integral}$ [ $\text{OF } \text{gm}$ ]
by  $\text{simp}$ 
have  $\text{pm}: (\lambda u. gPhi s \mu u * gPhi s \nu u) \in \text{borel-measurable lborel}$ 
using  $gPhi\text{-meas}$ [ $\text{OF } \text{assms}(1,3)$ ]  $gPhi\text{-meas}$ [ $\text{OF } \text{assms}(2,4)$ ]
by  $\text{simp}$ 
have  $R: \text{integral}^L \text{lborel } (\lambda u. gPhi s \mu u * gPhi s \nu u) =$ 
 $\text{enn2real } (\int^+ u. \text{ennreal } (gPhi s \mu u * gPhi s \nu u) \partial \text{lborel})$ 

```

```

    using integral-eq-nn-integral[OF pm] gPhi-nonneg
    by simp
  show ?thesis unfolding L R gauss-pair-nn[OF assms] using assms(5)
    by (simp add: enn2real-mult)
qed

lemma gPhi-prod-int:
  fixes  $\mu \ \nu :: \text{complex measure}$ 
  assumes fm: finite-measure  $\mu$  and fn: finite-measure  $\nu$  and sm: sets  $\mu =$ 
    sets borel and sn: sets  $\nu = \text{sets borel}$  and s:  $0 < s$ 
  shows integrable lborel ( $\lambda u. gPhi \ s \ \mu \ u * gPhi \ s \ \nu \ u$ )
proof -
  have m: ( $\lambda u. gPhi \ s \ \mu \ u * gPhi \ s \ \nu \ u$ )  $\in$  borel-measurable lborel
    using gPhi-meas[OF fm sm] gPhi-meas[OF fn sn]
    by simp
  have le1: ( $\int^+ w. \text{ennreal} (\exp (- s * (\text{norm} (z - w)) ^ 2)) \partial \nu$ )  $\leq$ 
    emeasure  $\nu$  (space  $\nu$ ) for  $z :: \text{complex}$ 
  proof -
    have ( $\int^+ w. \text{ennreal} (\exp (- s * (\text{norm} (z - w)) ^ 2)) \partial \nu$ )  $\leq$ 
      ( $\int^+ w. 1 \ \partial \nu$ ) using s
    by (fastforce intro: nn-integral-mono)
  thus ?thesis
    by simp
qed
  have le2: ( $\int^+ z. \int^+ w. \text{ennreal} (\exp (- s * (\text{norm} (z - w)) ^ 2)) \partial \nu \ \partial \mu$ )  $\leq$ 
    ( $\int^+ z. \text{emeasure} \ \nu \ (\text{space} \ \nu) \ \partial \mu$ )
    using le1
    by (blast intro: nn-integral-mono)
  have emeasure  $\nu$  (space  $\nu$ )  $\neq \infty$  using fn
    by (simp add: finite-measure.emeasure-finite)
  moreover have emeasure  $\mu$  (space  $\mu$ )  $\neq \infty$  using fm
    by (simp add: finite-measure.emeasure-finite)
  ultimately have fin: ( $\int^+ z. \text{emeasure} \ \nu \ (\text{space} \ \nu) \ \partial \mu$ )  $<$ 
     $\infty$ 
    by (simp add: ennreal-mult-less-top less-top)
  have fin2: ( $\int^+ u. \text{ennreal} (gPhi \ s \ \mu \ u * gPhi \ s \ \nu \ u) \ \partial \text{lborel}$ )  $<$ 
     $\infty$ 
  proof -
    have fin3: ( $\int^+ z. \int^+ w. \text{ennreal} (\exp (- s * (\text{norm} (z - w)) ^ 2)) \partial \nu \ \partial \mu$ )  $< \infty$  using le2 fin
    by simp
    show ?thesis unfolding gauss-pair-nn[OF fm fn sm sn s] using fin3
      by (simp add: ennreal-mult-less-top less-top)
  qed
  have e: ( $\lambda u. \text{ennreal} (\text{norm} (gPhi \ s \ \mu \ u * gPhi \ s \ \nu \ u))$ ) =
    ( $\lambda u. \text{ennreal} (gPhi \ s \ \mu \ u * gPhi \ s \ \nu \ u)$ ) using gPhi-nonneg
    by (simp add: fun-eq-iff)
  show ?thesis unfolding integrable-iff-bounded e using m fin2

```

by simp
qed

lemma *Eform-gauss-nonneg*:

assumes *finite-measure* $\nu 1$ *finite-measure* $\nu 2$ *sets* $\nu 1 = \text{sets borel sets } \nu 2 = \text{sets borel}$ $0 < s$

shows $0 \leq \text{Eform } \nu 1 \ \nu 2 \ (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^\wedge 2))$

proof –

have *i11*: *integrable lborel* $(\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 1 \ u)$

by (rule *gPhi-prod-int*[*OF assms*(1,1,3,3,5)])

have *i12*: *integrable lborel* $(\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 2 \ u)$

by (rule *gPhi-prod-int*[*OF assms*(1,2,3,4,5)])

have *i22*: *integrable lborel* $(\lambda u. gPhi \ s \ \nu 2 \ u * gPhi \ s \ \nu 2 \ u)$

by (rule *gPhi-prod-int*[*OF assms*(2,2,4,4,5)])

have *Eform* $\nu 1 \ \nu 2 \ (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^\wedge 2)) =$
 $4 * s / \pi * (\text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 1 \ u) - 2 * \text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 2 \ u) + \text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 2 \ u * gPhi \ s \ \nu 2 \ u))$

unfolding *Eform-def gauss-pair*[*OF assms*(1,1,3,3,5)] *gauss-pair*[*OF assms*(1,2,3,4,5)]
gauss-pair[*OF assms*(2,2,4,4,5)]

by (simp add: *algebra-simps*)

also have *integral*^L *lborel* $(\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 1 \ u) - 2 * \text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 2 \ u) + \text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 2 \ u * gPhi \ s \ \nu 2 \ u) = \text{integral}^L \text{lborel } (\lambda u. (gPhi \ s \ \nu 1 \ u - gPhi \ s \ \nu 2 \ u)^\wedge 2)$

proof –

have *integral*^L *lborel* $(\lambda u. (gPhi \ s \ \nu 1 \ u - gPhi \ s \ \nu 2 \ u)^\wedge 2) = \text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 1 \ u - 2 * (gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 2 \ u) + gPhi \ s \ \nu 2 \ u * gPhi \ s \ \nu 2 \ u)$
by (simp add: *power2-eq-square algebra-simps*)

also have ... = *integral*^L *lborel* $(\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 1 \ u) - 2 * \text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 1 \ u * gPhi \ s \ \nu 2 \ u) + \text{integral}^L \text{lborel } (\lambda u. gPhi \ s \ \nu 2 \ u * gPhi \ s \ \nu 2 \ u)$

using *i11 i12 i22*

by (simp add: *Bochner-Integration.integral-diff Bochner-Integration.integral-add*)

finally show *?thesis* ..

qed

finally have *E*: *Eform* $\nu 1 \ \nu 2 \ (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^\wedge 2)) = 4 * s / \pi * \text{integral}^L \text{lborel } (\lambda u. (gPhi \ s \ \nu 1 \ u - gPhi \ s \ \nu 2 \ u)^\wedge 2)$.

have $0 \leq \text{integral}^L \text{lborel } (\lambda u. (gPhi \ s \ \nu 1 \ u - gPhi \ s \ \nu 2 \ u)^\wedge 2)$

by simp

thus *?thesis* **unfolding** *E* **using** *assms*(5)

by (simp add: *zero-le-mult-iff*)

qed

Frullani-type kernel: $K\text{-}n(p) = (1/2)(\int \{-1/(n+1)\}^\wedge \{n+1\} e^\wedge \{-s\}/s \ ds - \int \{-1/(n+1)\}^\wedge \{n+1\} e^\wedge \{-s|p-1 - p-2|^\wedge 2\}/s \ ds) \longrightarrow \log|p-1 - p-2|$.

definition *Kn* :: *nat* $\Rightarrow$ *complex* $\times$ *complex* $\Rightarrow$ *real* **where**

Kn *n* *p* = $1 / 2 * (\text{interval-lebesgue-integral lborel } (\text{ereal } (1 / (\text{real } n + 1))))$

```

(ereal (real n +
  1)) (λs. exp (- s) / s)
  - interval-lebesgue-integral lborel (ereal (1 / (real n + 1))) (ereal
(real n +
  1)) (λs. exp (- s * (norm (fst p - snd p)) ^ 2) / s))

```

Helpers on $\int e^{-u}/u \, du$ for the kernels K - n .

lemma *ed-int*:

```

fixes x y :: real
assumes 0 < x 0 < y
shows interval-lebesgue-integrable lborel (ereal x) (ereal y) (λu. exp (- u) / u)
  and interval-lebesgue-integrable lborel (ereal x) (ereal y) (λu. 1 / u)
proof -
  have p: 0 < u if min x y ≤ u for u using that assms
  by simp
  show interval-lebesgue-integrable lborel (ereal x) (ereal y) (λu. exp (- u) / u)
    by (rule interval-integrable-isCont) (use p in ⟨auto intro!: continuous-intros⟩)
  show interval-lebesgue-integrable lborel (ereal x) (ereal y) (λu. 1 / u)
    by (rule interval-integrable-isCont) (use p in ⟨auto intro!: continuous-intros⟩)
qed

```

lemma *inv-ffc*:

```

fixes x y :: real
assumes x: 0 < x and y: 0 < y
shows interval-lebesgue-integral lborel (ereal (x)) (ereal (y)) (λu. 1 / u) =
  ln y - ln x
proof (rule interval-integral-FTC-finite)
  show continuous-on {min x y..max x y} (λu. 1 / u) using x y
    by (force intro: continuous-intros)
  fix u assume min x y ≤ u u ≤ max x y
  hence u: 0 < u using x y
  by simp
  show (ln has-vector-derivative 1 / u) (at u within {min x y..max x y})
    using has-vector-derivative-at-within[OF DERIV-ln-divide[OF u,
      unfolded has-real-derivative-iff-has-vector-derivative]]
  by simp
qed

```

lemma *ed-le*:

```

fixes x y :: real
assumes x: 0 < x and xy: x ≤ y
shows 0 ≤ interval-lebesgue-integral lborel (ereal (x)) (ereal (y)) (λu. exp (- u) /
/
  u) and interval-lebesgue-integral lborel (ereal (x)) (ereal (y)) (λu. exp (- u) /
  u) ≤ ln y - ln x
proof -
  have y: 0 < y using x xy
  by simp
  have e1: interval-lebesgue-integral lborel (ereal (x)) (ereal (y)) (λu. exp (- u) /

```



```

u) =
  set-lebesgue-integral lborel {x<..} (λu. exp (- u) / u)
  using xy
  by (simp add: interval-lebesgue-integral-le-eq)
  have e2: interval-lebesgue-integral lborel (ereal (x)) (ereal (y)) (λu. 1 / u) =
    set-lebesgue-integral lborel {x<..} (λu. 1 / u)
    using xy
    by (simp add: interval-lebesgue-integral-le-eq)
  have i1: set-integrable lborel {x<..} (λu. exp (- u) / u) using ed-int(1)[OF
x y] xy
    by (simp add: interval-lebesgue-integrable-def)
  have i2: set-integrable lborel {x<..} (λu. 1 / u) using ed-int(2)[OF x y] xy
    by (simp add: interval-lebesgue-integrable-def)
  have pt: exp (- u) / u ≤ 1 / u if u ∈ {x<..} for u
  proof -
    have u: 0 < u using that x
    by simp
    show ?thesis using u
    by (simp add: divide-right-mono)
  qed
  show 0 ≤ interval-lebesgue-integral lborel (ereal (x)) (ereal (y)) (λu. exp (- u)
/
  u) unfolding e1 set-lebesgue-integral-def using x
    using Bochner-Integration.integral-nonneg by (simp add: indicator-def)
  show interval-lebesgue-integral lborel (ereal (x)) (ereal (y)) (λu. exp (- u) / u)
≤
  ln y - ln x unfolding e1 inv-fts[OF x y, symmetric] e2
    using set-integral-mono[OF i1 i2 pt] .
  qed

lemma ed-abs:
  fixes x r :: real
  assumes x: 0 < x and r: 0 < r
  shows |interval-lebesgue-integral lborel (ereal (x)) (ereal (x *
r)) (λu. exp (- u) / u)| ≤ |ln r|
  proof -
    have xr: 0 < x * r using x r
    by simp
    have l: ln (x * r) = ln x + ln r using x r
    by (simp add: ln-mult)
    show ?thesis
    proof (cases 1 ≤ r)
      case True
      have x ≤ x * r using True x
      by simp
      thus ?thesis using ed-le[OF x] l True
      by force
    next
      case False

```

```

have  $x * r \leq x$  using False x
  by simp
hence  $0 \leq \text{interval-lebesgue-integral lborel } (\text{ereal } (x * r)) (\text{ereal } (x)) (\lambda u. \exp(-u) / u)$  and  $\text{interval-lebesgue-integral lborel } (\text{ereal } (x * r)) (\text{ereal } (x)) (\lambda u. \exp(-u) / u) \leq \ln x - \ln(x * r)$ 
  using ed-le[OF xr] by auto
thus ?thesis using interval-integral-endpoints-reverse[of ereal x ereal (x * r)  $\lambda u. \exp(-u) / u$ ] l False r
  by simp
qed
qed

lemma lbint-bound:
  fixes  $x y C :: \text{real}$  and  $g :: \text{real} \Rightarrow \text{real}$ 
  assumes gi: interval-lebesgue-integrable lborel (ereal x) (ereal y) g and b:  $\bigwedge u. \min x y \leq u \implies u \leq \max x y \implies |g u| \leq C$ 
  shows  $|\text{interval-lebesgue-integral lborel } (\text{ereal } (x)) (\text{ereal } (y)) (g)| \leq C * |y - x|$ 
  proof -
    have aux:  $|\text{interval-lebesgue-integral lborel } (\text{ereal } (a)) (\text{ereal } (c)) (g)| \leq C * (c - a)$ 
    if ac:  $a \leq c$  and ga: interval-lebesgue-integrable lborel (ereal a) (ereal c) g and
    ba:  $\bigwedge u. a \leq u \implies u \leq c \implies |g u| \leq C$ 
    for  $a c$ 
    proof -
      have e:  $\text{interval-lebesgue-integral lborel } (\text{ereal } (a)) (\text{ereal } (c)) (h) = \text{set-lebesgue-integral lborel } \{a <..< c\} h$  for  $h :: \text{real} \Rightarrow \text{real}$  using ac
      by (simp add: interval-lebesgue-integral-le-eq)
      have ig: set-integrable lborel {a <..< c} g using ga ac
      by (simp add: interval-lebesgue-integrable-def)
      have ic: set-integrable lborel {a <..< c} ( $\lambda-. k$ ) for  $k :: \text{real}$ 
      using interval-integral-const(1)[of a c k] ac
      by (simp add: interval-lebesgue-integrable-def)
      have vc:  $\text{set-lebesgue-integral lborel } \{a <..< c\} (\lambda-. k) = k * (c - a)$  for  $k :: \text{real}$ 
      using interval-integral-const(2)[of a c k] e[of  $\lambda-. k$ ]
      by simp
      have up:  $\text{set-lebesgue-integral lborel } \{a <..< c\} g \leq \text{set-lebesgue-integral lborel } \{a <..< c\} (\lambda-. C)$ 
      proof (rule set-integral-mono[OF ig ic])
        fix  $u$  assume  $u \in \{a <..< c\}$ 
        thus  $g u \leq C$  using ba[of  $u$ ]
        by simp
      qed
      have lo:  $\text{set-lebesgue-integral lborel } \{a <..< c\} (\lambda-. - C) \leq \text{set-lebesgue-integral lborel } \{a <..< c\} g$ 
      proof (rule set-integral-mono[OF ic ig])

```

```

    fix u assume u ∈ {a<..}
    thus - C ≤ g u using ba[of u]
      by simp
  qed
  show ?thesis unfolding e using up lo vc[of C] vc[of - C]
    by (simp add: abs-le-iff)
  qed
  show ?thesis
  proof (cases x ≤ y)
    case True
    show ?thesis using aux[OF True gi] b True
      by simp
  next
    case False
    have gi': interval-lebesgue-integrable lborel (ereal y) (ereal x) g
      using gi interval-integrable-endpoints-reverse
      by auto
    have |interval-lebesgue-integral lborel (ereal (y)) (ereal (x)) (g)| ≤
      C * (x - y) using aux[OF - gi'] b False
      by simp
    thus ?thesis using interval-integral-endpoints-reverse[of ereal x ereal y g] False
      by simp
  qed
  qed

lemma Kn-eq:
  assumes fst p ≠ snd p
  shows 2 * Kn n p = interval-lebesgue-integral lborel (ereal (1 / (real n + 1)))
    (ereal (1 /
      (real n + 1) * (norm (fst p - snd p)) ^ 2)) (λu. exp (- u) / u) -
    interval-lebesgue-integral lborel (ereal (real n + 1)) (ereal ((real n + 1) * (norm
      (fst p -
        snd p)) ^ 2)) (λu. exp (- u) / u)
  proof -
    define a :: real where a = 1 / (real n + 1)
    define b :: real where b = real n + 1
    define r :: real where r = (norm (fst p - snd p)) ^ 2
    have r0: 0 < r unfolding r-def using assms
      by simp
    have a0: 0 < a unfolding a-def
      by simp
    have b0: 0 < b unfolding b-def
      by simp
    have ab: a ≤ b
    proof -
      have one: 1 ≤ real n + 1
        by simp
      have 1 / (real n + 1) ≤ 1 using one
        by simp

```

```

    thus ?thesis unfolding a-def b-def using one
      by linarith
  qed
  have ar:  $0 < a * r$  using a0 r0
    by simp
  have br:  $0 < b * r$  using b0 r0
    by simp
  have sub0: interval-lebesgue-integral lborel (ereal (a)) (ereal (b)) ( $\lambda x. r *_R \exp$ 
    ( $-(x * r)$ ) / ( $x * r$ )) = interval-lebesgue-integral lborel (ereal (a * r)) (ereal
    (b * r)) ( $\lambda u. \exp (-u) / u$ )
  proof -
    have d: (( $\lambda s. s * r$ ) has-real-derivative r) (at x within {a..b}) for x
      by (auto intro: derivative-eq-intros)
    have img: ( $\lambda s. s * r$ ) ' {a..b}  $\subseteq$  {0<..} using a0 r0
      by auto
    have c0: continuous-on {0<..} ( $\lambda u :: \text{real}. \exp (-u) / u$ )
      by (fast intro: continuous-intros)
    have ci: continuous-on (( $\lambda s. s * r$ ) ' {a..b}) ( $\lambda u. \exp (-u) / u$ )
      using continuous-on-subset[OF c0 img] .
    show ?thesis using interval-integral-substitution-finite[OF ab,
      of  $\lambda s. s * r$   $\lambda \cdot. r \lambda u. \exp (-u) / u$ , OF d ci]
      by simp
  qed
  have cg: interval-lebesgue-integral lborel (ereal (a)) (ereal (b)) ( $\lambda s. \exp (-s * r)$ 
    /
     $s$ ) = interval-lebesgue-integral lborel (ereal (a)) (ereal (b)) ( $\lambda x. r *_R \exp (-$ 
    ( $x * r$ )) / ( $x * r$ ))
  proof (rule interval-integral-cong)
    fix x assume  $x \in \text{einterval} (\min (\text{ereal } a) (\text{ereal } b)) (\max (\text{ereal } a) (\text{ereal } b))$ 
    hence  $0 < x$  using a0 ab
      by simp
    thus  $\exp (-x * r) / x = r *_R \exp (- (x * r)) / (x * r)$  using r0
      by (simp add: field-simps)
  qed
  have sum1: interval-lebesgue-integral lborel (ereal (a)) (ereal (a * r)) ( $\lambda u. \exp$ 
    ( $-u$ ) /
     $u$ ) + interval-lebesgue-integral lborel (ereal (a * r)) (ereal (b)) ( $\lambda u. \exp (-u)$ 
    /  $u$ ) =
    interval-lebesgue-integral lborel (ereal (a)) (ereal (b)) ( $\lambda u. \exp (-u) / u$ )
    using interval-integral-sum[of  $\text{ereal } a$   $\text{ereal } (a * r)$   $\text{ereal } b$   $\lambda u. \exp (-u) /$ 
     $u$ ] ed-int(1)[of  $\min a (\min (a * r) b) \max a (\max (a * r) b)$ ] a0 ar b0
    by simp
  have sum2: interval-lebesgue-integral lborel (ereal (a * r)) (ereal (b)) ( $\lambda u. \exp$ 
    ( $-u$ ) /
     $u$ ) + interval-lebesgue-integral lborel (ereal (b)) (ereal (b * r)) ( $\lambda u. \exp (-u)$ 
    /  $u$ ) =
    interval-lebesgue-integral lborel (ereal (a * r)) (ereal (b * r)) ( $\lambda u. \exp (-u) /$ 
     $u$ )

```

```

using interval-integral-sum[of ereal (a * r) ereal b ereal (b *
  r)  $\lambda u. \exp(-u) / u$ ] ed-int(1)[of min (a * r) (min b (b * r)) max (a *
  r) (max b (b * r))] ar b0 br
by simp
have k:  $2 * Kn \ n \ p = \text{interval-lebesgue-integral lborel (ereal (a)) (ereal (b)) (\lambda u. \exp(-u) / u) - \text{interval-lebesgue-integral lborel (ereal (a)) (ereal (b)) (\lambda s. \exp(-s * r) / s)}$ 
unfolding Kn-def a-def b-def r-def
by simp
have  $2 * Kn \ n \ p = \text{interval-lebesgue-integral lborel (ereal (a)) (ereal (a * r)) (\lambda u. \exp(-u) / u) - \text{interval-lebesgue-integral lborel (ereal (b)) (ereal (b * r)) (\lambda u. \exp(-u) / u)}$ 
unfolding k cg sub0 using sum1 sum2
by simp
thus ?thesis unfolding a-def b-def r-def .
qed

```

lemma ed-near:

```

fixes x r :: real
assumes x:  $0 < x$  and r:  $0 < r$ 
shows  $|\text{interval-lebesgue-integral lborel (ereal (x)) (ereal (x * r)) (\lambda u. \exp(-u) / u) - \ln r| \leq x * |r - 1|$ 
proof -
have xr:  $0 < x * r$  using x r
by simp
have l:  $\text{interval-lebesgue-integral lborel (ereal (x)) (ereal (x * r)) (\lambda u. 1 / u) = \ln r}$  using inv-ffc[OF x xr] x r
by (simp add: ln-mult)
have d:  $\text{interval-lebesgue-integral lborel (ereal (x)) (ereal (x * r)) (\lambda u. \exp(-u) / u) - \text{interval-lebesgue-integral lborel (ereal (x)) (ereal (x * r)) (\lambda u. 1 / u) = \text{interval-lebesgue-integral lborel (ereal (x)) (ereal (x * r)) (\lambda u. \exp(-u) / u - 1 / u)}$ 
using interval-lebesgue-integral-diff(2)[OF ed-int(1)[OF x xr] ed-int(2)[OF x xr]] ..
have di:  $\text{interval-lebesgue-integrable lborel (ereal x) (ereal (x * r)) (\lambda u. \exp(-u) / u - 1 / u)}$ 
using interval-lebesgue-integral-diff(1)[OF ed-int(1)[OF x xr] ed-int(2)[OF x xr]] .
have b:  $|\exp(-u) / u - 1 / u| \leq 1$ 
if min x (x * r)  $\leq u \leq$  max x (x * r) for u
proof -
have u:  $0 < u$  using that x xr
by simp
have e1:  $1 - u \leq \exp(-u)$  using exp-ge-add-one-self[of - u]

```

```

    by simp
    have e2:  $\exp(-u) \leq 1$  using u
    by simp
    have  $|\exp(-u) / u - 1 / u| = (1 - \exp(-u)) / u$  using u e2
    by (simp add: field-simps)
    also have  $\dots \leq 1$  using u e1
    by (simp add: divide-le-eq)
    finally show ?thesis .
qed
have  $|\text{interval-lebesgue-integral lborel}(\text{ereal}(x))(\text{ereal}(x * r))(\lambda u. \exp(-u) / u - 1 / u)| \leq 1 * |x * r - x|$ 
  by (rule lbint-bound[OF di b])
moreover have  $1 * |x * r - x| = x * |r - 1|$ 
proof -
  have  $x * r - x = x * (r - 1)$ 
  by (simp add: algebra-simps)
  thus ?thesis using x
  by (simp add: abs-mult)
qed
ultimately show ?thesis unfolding d[symmetric] l
  by simp
qed

lemma ed-far:
  fixes b r :: real
  assumes b:  $0 < b$  and r:  $0 < r$ 
  shows  $|\text{interval-lebesgue-integral lborel}(\text{ereal}(b))(\text{ereal}(b * r))(\lambda u. \exp(-u) / u)| \leq |r - 1| / (b * (\min 1 r) ^ 2)$ 
proof -
  have br:  $0 < b * r$  using b r
  by simp
  have m0:  $0 < b * \min 1 r$  using b r
  by simp
  have bd:  $|\exp(-u) / u| \leq 1 / (b * \min 1 r) ^ 2$ 
  if  $\min b (b * r) \leq u \leq \max b (b * r)$  for u
  proof -
    have mu:  $b * \min 1 r \leq u$  using that b r
    by (auto simp: min-def)
    have u:  $0 < u$  using mu m0
    by simp
    have e1:  $\exp(-u) \leq 1 / u$ 
    proof -
      have  $u \leq \exp u$  using exp-ge-add-one-self[of u]
      by linarith
      thus ?thesis using u
      by (simp add: exp-minus field-simps)
    qed
    have  $|\exp(-u) / u| = \exp(-u) / u$  using u
    by simp

```

```

also have ... ≤ 1 / u / u using e1 u
  using divide-right-mono[OF e1, of u] by simp
also have ... = 1 / u ^ 2
  by (simp add: power2-eq-square)
also have ... ≤ 1 / (b * min 1 r) ^ 2
  using power-mono[OF mu less-imp-le[OF m0], of 2] m0
  using u mu m0 by (metis ‹1 / u / u =
    1 / u^2› pos-divide-le-eq pos-le-divide-eq power2-eq-square
    times-divide-eq-left mult-1 zero-compare-simps(6))
finally show ?thesis .
qed
have |interval-lebesgue-integral lborel (ereal (b)) (ereal (b * r)) (λu. exp (- u) /
  u)| ≤ 1 / (b * min 1 r) ^ 2 * |b * r - b|
  by (rule lbint-bound[OF ed-int(1)[OF b br] bd])
also have ... = |r - 1| / (b * (min 1 r) ^ 2)
proof -
  have b * r - b = b * (r - 1)
  by (simp add: algebra-simps)
  hence ab: |b * r - b| = b * |r - 1| using b
  by (simp add: abs-mult)
  show ?thesis unfolding ab using b
  by (simp add: power2-eq-square field-simps)
qed
finally show ?thesis .
qed

lemma Kn-bound: fst p ≠ snd p ⟹
  |Kn n p| ≤ 2 * |ln (norm (fst p - snd p))|
proof -
  assume ne: fst p ≠ snd p
  have r0: 0 < (norm (fst p - snd p)) ^ 2 using ne
  by simp
  have a0: 0 < 1 / (real n + 1)
  by simp
  have b0: 0 < real n + 1
  by simp
  have e: 2 * Kn n p = interval-lebesgue-integral lborel (ereal (1 / (real n + 1)))
    (ereal (1 /
      (real n + 1) * (norm (fst p - snd p)) ^ 2)) (λu. exp (- u) / u) -
    interval-lebesgue-integral lborel (ereal (real n + 1)) (ereal ((real n + 1) * (norm
      (fst p -
        snd p)) ^ 2)) (λu. exp (- u) / u) by (rule Kn-eq[OF ne])
  have hA: |interval-lebesgue-integral lborel (ereal (1 / (real n + 1))) (ereal (1 /
    (real n +
      1) * (norm (fst p - snd p)) ^ 2)) (λu. exp (- u) / u)| ≤
    |ln ((norm (fst p - snd p)) ^ 2)| by (rule ed-abs[OF a0 r0])
  have hB: |interval-lebesgue-integral lborel (ereal (real n + 1)) (ereal ((real n +
    1) *
      (norm (fst p - snd p)) ^ 2)) (λu. exp (- u) / u)| ≤

```

```

|ln ((norm (fst p - snd p)) ^ 2)| by (rule ed-abs[OF b0 r0])
have lr: ln ((norm (fst p - snd p)) ^ 2) = 2 * ln (norm (fst p - snd p))
  by (simp add: ln-realpow)
show |Kn n p| ≤ 2 * |ln (norm (fst p - snd p))| using e hA hB
  unfolding lr
  by simp
qed

lemma Kn-tendsto: fst p ≠ snd p ⇒
  ((λn. Kn n p) ⟶ ln (norm (fst p - snd p))) at-top
proof -
  assume ne: fst p ≠ snd p
  define r where r = (norm (fst p - snd p)) ^ 2
  define K where K = (|r - 1| + |r - 1| / (min 1 r) ^ 2) / 2
  have r0: 0 < r unfolding r-def using ne
    by simp
  have lr: ln r = 2 * ln (norm (fst p - snd p)) unfolding r-def
    by (simp add: ln-realpow)
  have bd: norm (Kn n p - ln (norm (fst p - snd p))) ≤ K * inverse (real (Suc n)) for n
  proof -
    have a0: 0 < 1 / (real n + 1)
      by simp
    have b0: 0 < real n + 1
      by simp
    have e: 2 * Kn n p = interval-lebesgue-integral lborel (ereal (1 / (real n + 1)))
      (ereal (1 /
        (real n + 1) * (norm (fst p - snd p)) ^ 2)) (λu. exp (- u) / u) -
        interval-lebesgue-integral lborel (ereal (real n + 1)) (ereal ((real n + 1) *
        (norm (fst p -
        snd p)) ^ 2)) (λu. exp (- u) / u) by (rule Kn-eq[OF ne])
    have hA: |interval-lebesgue-integral lborel (ereal (1 / (real n + 1))) (ereal (1 /
      (real n + 1) * (norm (fst p - snd p)) ^ 2)) (λu. exp (- u) / u) - ln r| ≤
      1 / (real n + 1) * |r - 1| using ed-near[OF a0 r0] unfolding r-def .
    have hB: |interval-lebesgue-integral lborel (ereal (real n + 1)) (ereal ((real n +
      1) *
      (norm (fst p - snd p)) ^ 2)) (λu. exp (- u) / u)| ≤
      |r - 1| / ((real n + 1) * (min 1 r) ^ 2) using ed-far[OF b0 r0] unfolding
      r-def .
    have |Kn n p - ln (norm (fst p - snd p))| ≤
      (1 / (real n + 1) * |r - 1| + |r - 1| / ((real n + 1) * (min 1 r) ^ 2)) /
      2
      using e hA hB lr
      by argo
    also have ... = K * inverse (real (Suc n))
  proof -
    have m: 0 < (min 1 r) ^ 2 using r0
      by simp
    have N: inverse (real (Suc n)) = 1 / (real n + 1)

```



```

    by (simp add: inverse-eq-divide)
  show ?thesis unfolding K-def N using m
    using field-simps by argo
qed
finally show ?thesis
  by simp
qed
have z: ((λn. K * inverse (real (Suc n))) ⟶ 0) sequentially
  using tendsto-mult-right-zero[OF LIMSEQ-inverse-real-of-nat, of K]
  by simp
have ev: eventually (λn. norm (Kn n p - ln (norm (fst p - snd p))) ≤
  K * inverse (real (Suc n))) sequentially
  using bd by (simp add: always-eventually)
have ((λn. Kn n p - ln (norm (fst p - snd p))) ⟶ 0) sequentially
  using Lim-null-comparison[OF ev z] .
thus ((λn. Kn n p) ⟶ ln (norm (fst p - snd p))) sequentially
  by (simp add: LIM-zero-iff)
qed

lemma Kn-int-P:
  fixes P :: (complex × complex) measure
  assumes fP: finite-measure P and sP: sets P = sets (borel  $\otimes_M$  borel)
  shows integralL P (Kn n) = 1 / 2 * (interval-lebesgue-integral lborel (ereal (1 /
    (real n + 1))) (ereal (real n + 1)) (λs. exp (- s) / s) * measure P (space P) -
    integralL lborel (λs. indicator {1 / (real n + 1) <..L P (λp. exp (- s * (norm (fst p - snd p)) ^ 2))))
    and integrable lborel (λs. indicator {1 / (real n + 1) <..L P (λp. exp (- s * (norm (fst p - snd p)) ^ 2)))
  proof -
    define a :: real where a = 1 / (real n + 1)
    define b :: real where b = real n + 1
    define g where g p s = indicator {a <..L lborel (g p) for p
    unfolding g-def using ab
    by (simp add: interval-lebesgue-integral-le-eq set-lebesgue-integral-def)

```

```

have Kg:  $\text{Kn } n \text{ } p = 1 / 2 * (\text{interval-lebesgue-integral } \text{lborel } (\text{ereal } (1 / (\text{real } n + 1)))) (\text{ereal } (\text{real } n + 1)) (\lambda s. \text{exp } (- s) / s) - \text{integral}^L \text{lborel } (g \text{ } p))$  for  $p$ 
+
  unfolding Kn-def H[symmetric] a-def b-def ..
have sigma-finite-measure P using fP
  by (simp add: finite-measure-def)
hence ps: pair-sigma-finite P lborel
  by (simp add: pair-sigma-finite-def sigma-finite-lborel)
have gm:  $(\lambda x. g \text{ } (\text{fst } x) (\text{snd } x)) \in \text{borel-measurable } (P \otimes_M \text{lborel})$ 
  unfolding g-def measurable-cong-sets[OF sets-pair-measure-cong[OF sP refl] refl]
  by simp
have gb:  $|g \text{ } p \text{ } s| \leq \text{indicator } \{a < .. < b\} \text{ } s * (1 / a)$  for  $p \text{ } s$ 
proof (cases  $s \in \{a < .. < b\}$ )
  case True
    have s0:  $0 < s$  using True a0
    by simp
    have exp:  $-\text{exp } (- s * (\text{norm } (\text{fst } p - \text{snd } p)) ^ 2) \leq 1$  using s0
    by simp
    hence:  $|\text{exp } (- s * (\text{norm } (\text{fst } p - \text{snd } p)) ^ 2) / s| \leq 1 / s$  using s0
    by (simp add: divide-right-mono)
    also have  $1 / s \leq 1 / a$  using True a0
    by (simp add: divide-left-mono)
    finally show ?thesis unfolding g-def using True
    by simp
  next
  case False
    thus ?thesis unfolding g-def
    by simp
qed
have wint:  $\text{integrable } (P \otimes_M \text{lborel}) (\lambda x. \text{indicator } \{a < .. < b\} (\text{snd } x) * (1 / a))$ 
proof -
  have m:  $(\lambda x. \text{indicator } \{a < .. < b\} (\text{snd } x) * (1 / a)) \in \text{borel-measurable } (P \otimes_M \text{lborel})$ 
    unfolding measurable-cong-sets[OF sets-pair-measure-cong[OF sP refl] refl]
    by simp
  have e:  $(\int^+ x. \text{ennreal } (\text{norm } (\text{indicator } \{a < .. < b\} (\text{snd } x) * (1 / a)))) \partial(P \otimes_M \text{lborel}) =$ 
     $(\int^+ p. \int^+ s. \text{ennreal } (\text{indicator } \{a < .. < b\} s * (1 / a))) \partial \text{lborel } \partial P$ 
    using lborel.nn-integral-fst[of  $\lambda x. \text{ennreal } (\text{norm } (\text{indicator } \{a < .. < b\} (\text{snd } x) * (1 / a))) P$ ] m a0
    by (simp add: split-beta)
  have i:  $(\int^+ s. \text{ennreal } (\text{indicator } \{a < .. < b\} s * (1 / a)) \partial \text{lborel}) = \text{ennreal } (1 / a) * \text{emeasure } \text{lborel } \{a < .. < b\}$ 
  proof -
    have eqf:  $(\lambda s. \text{ennreal } (\text{indicator } \{a < .. < b\} s * (1 / a))) =$ 

```

```

    (λs. ennreal (1 / a) * indicator {a<..b} s)
  by (simp add: fun-eq-iff indicator-def)
show ?thesis unfolding eqf
  by (simp add: nn-integral-cmult-indicator)
qed
have f1: emeasure lborel {a<..b} ≠ ∞
  using ab by simp
have f2: emeasure P (space P) ≠ ∞ using fP
  by (simp add: finite-measure.emeasure-finite)
have (∫+ x. ennreal (norm (indicator {a<..b} (snd x) * (1 /
a))) ∂(P ⊗M lborel)) < ∞ unfolding e i using f1 f2
  by (simp add: ennreal-mult-less-top less-top)
thus ?thesis unfolding integrable-iff-bounded using m
  by simp
qed
have gint: integrable (P ⊗M lborel) (case-prod g)
proof -
  have gm': (λ(x, y). g x y) ∈ borel-measurable (P ⊗M lborel)
    using gm
  by (simp add: split-beta)
  have ae: AE x in P ⊗M lborel. norm (case x of (y, z) ⇒ g y z) ≤
    norm (indicator {a<..b} (snd x) * (1 / a) :: real)
    using gb a0 by (simp add: split-beta abs-mult)
  show ?thesis using Bochner-Integration.integrable-bound[OF wint gm' ae] .
qed
have Fub: integralL lborel (λs. integralL P (λp. g p s)) =
  integralL P (λp. integralL lborel (g p))
  using pair-sigma-finite.Fubini-integral[OF ps gint]
  by simp
have Hint: integrable P (λp. integralL lborel (g p))
  using pair-sigma-finite.integrable-fst'[OF ps gint]
  by simp
have Sint: integrable lborel (λs. integralL P (λp. g p s))
  using pair-sigma-finite.integrable-snd[OF ps gint]
  by simp
have inner: integralL P (λp. g p s) =
  indicator {a<..b} s / s * integralL P (λp. exp (- s * (norm (fst p -
snd p)) ^ 2)) for s
  unfolding g-def
  by simp
have ci: integrable P (λp. interval-lebesgue-integral lborel (ereal (1 / (real n +
1))) (ereal (real n + 1)) (λs. exp (- s) / s)) using fP
  by (simp add: finite-measure.integrable-const)
show integrable lborel (λs. indicator {1 / (real n + 1)<..b} s / s *
integralL P (λp. exp (- s * (norm (fst p - snd p)) ^ 2))) using Sint
  unfolding inner a-def b-def .
have integralL P (Kn n) = integralL P (λp. 1 / 2 *
(interval-lebesgue-integral lborel (ereal (1 / (real n + 1))) (ereal (real n +
1))) (λs. exp (- s) / s) - integralL lborel (g p)))

```

```

    unfolding Kg ..
  also have ... = 1 / 2 * (interval-lebesgue-integral lborel (ereal (1 / (real n +
    1))) (ereal (real n + 1)) (λs. exp (- s) / s) * measure P (space P) -
    integralL P (λp. integralL lborel (g p)))
    using ci Hint
    by (simp add: Bochner-Integration.integral-diff)
  finally show integralL P (Kn n) =
    1 / 2 * (interval-lebesgue-integral lborel (ereal (1 / (real n + 1))) (ereal (real
n +
    1)) (λs. exp (- s) / s) * measure P (space P) -
    integralL lborel (λs. indicator {1 / (real n + 1) <..< real n + 1} s / s *
    integralL P (λp. exp (- s * (norm (fst p - snd p)) ^ 2))))
    unfolding Fub[symmetric] inner a-def b-def .
qed

```

lemma *Eform-Kn-nonpos*:

```

  assumes finite-measure ν1 finite-measure ν2 sets ν1 = sets borel sets ν2 =
    sets borel emeasure ν1 UNIV = emeasure ν2 UNIV
  shows Eform ν1 ν2 (Kn n) ≤ 0
proof -
  have f11: finite-measure (ν1 ⊗M ν1)
    using finite-measure-pair-measure[OF assms(1,1)] .
  have f12: finite-measure (ν1 ⊗M ν2)
    using finite-measure-pair-measure[OF assms(2,1)] .
  have f22: finite-measure (ν2 ⊗M ν2)
    using finite-measure-pair-measure[OF assms(2,2)] .
  have s11: sets (ν1 ⊗M ν1) = sets (borel ⊗M borel)
    using sets-pair-measure-cong[OF assms(3,3)] .
  have s12: sets (ν1 ⊗M ν2) = sets (borel ⊗M borel)
    using sets-pair-measure-cong[OF assms(3,4)] .
  have s22: sets (ν2 ⊗M ν2) = sets (borel ⊗M borel)
    using sets-pair-measure-cong[OF assms(4,4)] .
  have mP: measure (μ ⊗M ν) (space (μ ⊗M ν)) =
    measure μ (space μ) * measure ν (space ν)
  if finite-measure μ finite-measure ν for μ ν :: complex measure
proof -
  have sf: sigma-finite-measure ν using that(2)
    by (simp add: finite-measure-def)
  have emeasure (μ ⊗M ν) (space μ × space ν) =
    emeasure μ (space μ) * emeasure ν (space ν)
    using sigma-finite-measure.emeasure-pair-measure-Times[OF sf,
    of space μ μ space ν]
    by simp
  thus ?thesis unfolding measure-def space-pair-measure
    by (simp add: enn2real-mult)
qed
  have space ν1 = UNIV using sets-eq-imp-space-eq[OF assms(3)]
    by simp
  moreover have space ν2 = UNIV using sets-eq-imp-space-eq[OF assms(4)]

```

```

    by simp
  ultimately have m12: measure  $\nu 1$  (space  $\nu 1$ ) = measure  $\nu 2$  (space  $\nu 2$ )
    using assms(5)
    by (simp add: measure-def)
  note I11 = Kn-int-P[OF f11 s11, of n] and I12 = Kn-int-P[OF f12 s12, of n]
and I22 =
  Kn-int-P[OF f22 s22, of n]
have nn:  $0 \leq \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) <..< \text{real } n + 1\} s / s * (\text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)) - 2 * \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)) + \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2))))$ 
proof (rule Bochner-Integration.integral-nonneg)
  fix s :: real
  show  $0 \leq \text{indicator} \{1 / (\text{real } n + 1) <..< \text{real } n + 1\} s / s * (\text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)) - 2 * \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)) + \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)))$ 
  proof (cases  $s \in \{1 / (\text{real } n + 1) <..< \text{real } n + 1\}$ )
    case True
    have p0:  $0 < 1 / (\text{real } n + 1)$ 
    by simp
    have  $1 / (\text{real } n + 1) < s$  using True
    by (simp add: greaterThanLessThan-iff)
    hence s0:  $0 < s$  using p0
    by linarith
    have  $0 \leq \text{Eform } \nu 1 \nu 2 (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2))$ 
    by (rule Eform-gauss-nonneg[OF assms(1-4) s0])
    thus ?thesis unfolding Eform-def using s0 True
    by simp
  next
    case False
    thus ?thesis
    by simp
  qed
qed
have split:  $\text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) <..< \text{real } n + 1\} s / s * \text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2))) - 2 * \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) <..< \text{real } n + 1\} s / s * \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2))) + \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) <..< \text{real } n + 1\} s / s * \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)))$ 
=  $\text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) <..< \text{real } n + 1\} s / s * (\text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)) - 2 * \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)) + \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp(-s * (\text{norm}(\text{fst } p - \text{snd } p))^2)))$ 

```

$$\text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))))$$

proof –

have *a*: $\text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s$

*

$$\begin{aligned} & \text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2)) - 2 * (\text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \\ & \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) + \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \\ & \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) = \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \\ & \text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) - 2 * \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \\ & \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) + \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \\ & \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) \end{aligned}$$

using *Bochner-Integration.integral-add*[*OF Bochner-Integration.integrable-diff*][*OF I11*(2)]

$$\text{integrable-mult-right}[\textbf{where } c = 2, \text{ OF } I12(2)] I22(2)]$$

Bochner-Integration.integral-diff[*OF I11*(2)] *integrable-mult-right*[**where** *c* = 2, *OF I12*(2)]]

Bochner-Integration.integral-mult-right[**where** *c* = 2, *OF I12*(2)] **by** *simp*

have *b*: $(\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2)) - 2 * (\text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) + \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) = (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * (\text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2)) - 2 * \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2)) + \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))))$

by (*simp add: fun-eq-iff algebra-simps*)

show *?thesis* **using** *a* **unfolding** *b* **by** *simp*

qed

have *Eform* $\nu 1 \nu 2 (Kn \ n) = - 1 / 2 * (\text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \text{integral}^L (\nu 1 \otimes_M \nu 1) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) - 2 * \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \text{integral}^L (\nu 1 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))) + \text{integral}^L \text{lborel} (\lambda s. \text{indicator} \{1 / (\text{real } n + 1) < .. < \text{real } n + 1\} s / s * \text{integral}^L (\nu 2 \otimes_M \nu 2) (\lambda p. \exp (- s * (\text{norm} (\text{fst } p - \text{snd } p)) ^ 2))))$

unfolding *Eform-def* *I11*(1) *I12*(1) *I22*(1) *mP*[*OF assms*(1,1)] *mP*[*OF assms*(1,2)] *mP*[*OF assms*(2,2)]

m12

```

    by (simp add: algebra-simps)
  thus ?thesis unfolding split using nn
    by simp
qed

```

Integrability passes to a smaller measure with the same sets.

```

lemma integrable-le-measure:
  fixes f :: 'a  $\Rightarrow$  real
  assumes PQ:  $P \leq Q$  and s: sets P = sets Q and fi: integrable Q f
  shows integrable P f
proof -
  have m:  $f \in \text{borel-measurable } P$  using borel-measurable-integrable[OF fi]
    measurable-cong-sets[OF s refl]
  by auto
  have le:  $(\int^+ x. \text{ennreal } (\text{norm } (f x)) \partial P) \leq$ 
     $(\int^+ x. \text{ennreal } (\text{norm } (f x)) \partial Q)$ 
    using nn-integral-mono-measure[OF s PQ]
  by simp
  have  $(\int^+ x. \text{ennreal } (\text{norm } (f x)) \partial Q) < \infty$  using fi
    unfolding integrable-iff-bounded
  by simp
  thus ?thesis using m le unfolding integrable-iff-bounded
  by simp
qed

```

```

lemma Kn-meas:  $Kn \ n \in \text{borel-measurable } (\text{borel} \otimes_M \text{borel} :: (\text{complex} \times \text{complex}) \text{ measure})$ 
  unfolding Kn-def interval-lebesgue-integral-def set-lebesgue-integral-def
  by simp

```

Dominated convergence: $\int K \cdot n \ dP \longrightarrow \int \log|p-1 - p-2| \ dP$ for $P \leq Q$ with $\log Q$ -integrable and the diagonal Q -null.

```

lemma Kn-integral-tendsto:
  assumes P  $\leq$  Q sets P = sets  $(\text{borel} \otimes_M \text{borel})$  sets Q =
    sets  $(\text{borel} \otimes_M \text{borel})$ 
  and integrable Q  $(\lambda p :: \text{complex} \times \text{complex}. \ln (\text{norm } (\text{fst } p - \text{snd } p)))$ 
    emeasure Q  $\{p. \text{fst } p = \text{snd } p\} = 0$ 
  shows  $((\lambda n. \text{integral}^L P (Kn \ n)) \longrightarrow \text{integral}^L P$ 
     $(\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))))$  at-top
proof -
  define D where  $D = \{p :: \text{complex} \times \text{complex}. \text{fst } p = \text{snd } p\}$ 
  have sPQ: sets P = sets Q using assms(2,3) by simp
  have mf:  $(\lambda p :: \text{complex} \times \text{complex}. \text{fst } p - \text{snd } p) \in$ 
    borel-measurable  $(\text{borel} \otimes_M \text{borel})$ 
  by simp
  have eq:  $\{p :: \text{complex} \times \text{complex}. \text{fst } p = \text{snd } p\} =$ 
     $(\lambda p. \text{fst } p - \text{snd } p) - \{0\} \cap \text{space } (\text{borel} \otimes_M \text{borel})$ 
  by (auto simp: space-pair-measure)

```

have Dm : $D \in \text{sets } P$ **unfolding** $D\text{-def } \text{assms}(2)$ **eq using** $\text{measurable-sets}[OF$
 $mf, \text{ of } \{0\}]$
by simp
have $\text{emeasure } P D = 0$ **using** $\text{le-measureD3}[OF \text{ assms}(1) \text{ sPQ}, \text{ of } D] \text{ assms}(5)$
unfolding $D\text{-def}$
by simp
hence ns : $D \in \text{null-sets } P$ **using** Dm
by $(\text{simp add: null-sets-def})$
have $AE\text{-off}$: $AE \text{ } p \text{ in } P. \text{fst } p \neq \text{snd } p$
using $AE\text{-I}'[OF \text{ ns}, \text{ of } \lambda p. \text{fst } p \neq \text{snd } p]$ **unfolding** $D\text{-def}$
by auto
have intQ : $\text{integrable } Q (\lambda p. 2 * |\ln (\text{norm } (\text{fst } p - \text{snd } p))|)$ **using** $\text{assms}(4)$
by simp
have intP : $\text{integrable } P (\lambda p. 2 * |\ln (\text{norm } (\text{fst } p - \text{snd } p))|)$
by $(\text{rule integrable-le-measure}[OF \text{ assms}(1) \text{ sPQ } \text{intQ}])$
have fm : $(\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))) \in \text{borel-measurable } P$
using $\text{borel-measurable-integrable}[OF \text{ integrable-le-measure}[OF \text{ assms}(1) \text{ sPQ}$
 $\text{assms}(4)]]$.
have sm : $Kn \text{ } n \in \text{borel-measurable } P$ **for** n
using $Kn\text{-meas}[\text{of } n] \text{ measurable-cong-sets}[OF \text{ assms}(2) \text{ refl}]$
by auto
have lim : $AE \text{ } p \text{ in } P. (\lambda n. Kn \text{ } n \text{ } p) \longrightarrow \ln (\text{norm } (\text{fst } p - \text{snd } p))$
using $AE\text{-off } Kn\text{-tendsto}$
by $(\text{auto elim: AE-mp})$
have bd : $AE \text{ } p \text{ in } P. \text{norm } (Kn \text{ } n \text{ } p) \leq 2 * |\ln (\text{norm } (\text{fst } p - \text{snd } p))|$ **for** n
using $AE\text{-off } Kn\text{-bound}$
by $(\text{auto elim: AE-mp})$
show $?thesis$ **by** $(\text{rule integral-dominated-convergence}[OF \text{ fm } \text{sm } \text{intP } \text{lim } \text{bd}])$
qed

lemma finite-le-measure :

assumes $\nu \leq \eta$ $\text{sets } \nu = \text{sets } \eta$ $\text{finite-measure } \eta$
shows $\text{finite-measure } \nu$
proof $(\text{rule finite-measureI})$
have fe : $\text{emeasure } \eta (\text{space } \eta) \neq \infty$ **using** $\text{assms}(3)$
by $(\text{simp add: finite-measure.emeasure-finite})$
have le : $\text{emeasure } \nu (\text{space } \nu) \leq \text{emeasure } \eta (\text{space } \nu)$
using $\text{le-measureD3}[OF \text{ assms}(1,2)]$.
have sp : $\text{space } \nu = \text{space } \eta$ **using** $\text{sets-eq-imp-space-eq}[OF \text{ assms}(2)]$.
show $\text{emeasure } \nu (\text{space } \nu) \neq \infty$ **using** $\text{le } \text{fe}$ **unfolding** sp
by $(\text{auto simp: top.extremum-unique})$
qed

lemma pair-measure-mono :

assumes $\nu \leq \eta$ $\nu' \leq \eta$ $\text{sets } \nu = \text{sets borel sets } \nu' =$
 $\text{sets borel sets } \eta = \text{sets borel finite-measure } \eta$
shows $(\nu \otimes_M \nu') \leq (\eta \otimes_M \eta)$
proof –
have s1 : $\text{sets } (\nu \otimes_M \nu') = \text{sets } (\eta \otimes_M \eta)$


```

    using assms(3,4,5)
    using sets-pair-measure-cong by simp
    have sp: space ( $\nu \otimes_M \nu'$ ) = space ( $\eta \otimes_M \eta$ )
    using sets-eq-imp-space-eq[OF s1] .
    have finite-measure  $\nu'$  using finite-le-measure[OF assms(2) - assms(6)] assms(4,5)
    by simp
    hence sfn': sigma-finite-measure  $\nu'$ 
    by (simp add: finite-measure-def)
    have sfe: sigma-finite-measure  $\eta$  using assms(6)
    by (simp add: finite-measure-def)
    have em: emeasure ( $\nu \otimes_M \nu'$ )  $X \leq$ 
    emeasure ( $\eta \otimes_M \eta$ )  $X$  for  $X$ 
    proof (cases  $X \in \text{sets } (\nu \otimes_M \nu')$ )
    case True
    have X2:  $X \in \text{sets } (\eta \otimes_M \eta)$  using True s1
    by simp
    have emeasure ( $\nu \otimes_M \nu'$ )  $X =$ 
    ( $\int^+ x. \int^+ y. \text{indicator } X (x,$ 
     $y) \partial\nu' \partial\nu$ )
    using sigma-finite-measure.emeasure-pair-measure[OF sfn' True] .
    also have ...  $\leq (\int^+ x. \int^+ y. \text{indicator } X (x,$ 
     $y) \partial\eta \partial\eta)$ 
    using nn-integral-mono-measure[of  $\nu' \eta$ ] assms(2,4,5)
    by (auto intro: nn-integral-mono)
    also have ...  $\leq (\int^+ x. \int^+ y. \text{indicator } X (x,$ 
     $y) \partial\eta \partial\eta)$ 
    using nn-integral-mono-measure[of  $\nu \eta$ ] assms(1,3,5)
    using nn-integral-mono by simp
    also have ... = emeasure ( $\eta \otimes_M \eta$ )  $X$ 
    using sigma-finite-measure.emeasure-pair-measure[OF sfe X2]
    by simp
    finally show ?thesis .
  next
  case False
  have emeasure ( $\nu \otimes_M \nu'$ )  $X = 0$  using False
  by (simp add: emeasure-notin-sets)
  thus ?thesis
  by simp
qed
show ?thesis unfolding le-measure-iff using s1 sp em
by (simp add: le-fun-def)
qed

```

```

lemma logEnergyC-prod:
  fixes  $\mu \nu :: \text{complex measure}$ 
  assumes fm: finite-measure  $\mu$  and fn: finite-measure  $\nu$  and int: integrable ( $\mu$ 
     $\otimes_M \nu$ ) ( $\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))$ )
  shows logEnergyC  $\mu \nu = \text{integral}^L (\mu \otimes_M \nu) (\lambda p. \ln$ 
    ( $\text{norm } (\text{fst } p - \text{snd } p)$ ))

```

```

proof –
  have ps: pair-sigma-finite  $\mu$   $\nu$  using fm fn
    by (simp add: pair-sigma-finite-def finite-measure-def)
  show ?thesis unfolding logEnergyC-def using pair-sigma-finite.integral-fst'[OF
ps int]
    by simp
qed

theorem log-energy-zero-mass-nonpos:
  fixes  $\nu1$   $\nu2$   $\eta$  :: complex measure
  assumes finite-measure  $\eta$  sets  $\nu1 = \text{sets borel}$  sets  $\nu2 =$ 
    sets borel sets  $\eta = \text{sets borel}$ 
  and  $\nu1 \leq \eta$   $\nu2 \leq \eta$  emeasure  $\nu1$  UNIV = emeasure  $\nu2$  UNIV
  and integrable ( $\eta \otimes_M \eta$ ) ( $\lambda zw. \ln (\text{norm} (\text{fst } zw - \text{snd } zw))$ )
  and emeasure ( $\eta \otimes_M \eta$ ) { $zw. \text{fst } zw = \text{snd } zw$ } = 0
  shows logEnergyC  $\nu1$   $\nu1 - 2 * \text{logEnergyC } \nu1 \nu2 + \text{logEnergyC } \nu2 \nu2 \leq 0$ 
proof –
  have f1: finite-measure  $\nu1$  and f2: finite-measure  $\nu2$ 
  proof –
    have fe: emeasure  $\eta$  (space  $\eta$ )  $\neq \infty$  using assms(1)
      by (simp add: finite-measure.emeasure-finite)
    show finite-measure  $\nu1$ 
  proof (rule finite-measureI)
    have le: emeasure  $\nu1$  (space  $\nu1$ )  $\leq$  emeasure  $\eta$  (space  $\nu1$ )
      using le-measureD3[OF assms(5)] assms(2,4)
      by simp
    have sp: space  $\nu1 = \text{space } \eta$  using sets-eq-imp-space-eq assms(2,4)
      by auto
    show emeasure  $\nu1$  (space  $\nu1$ )  $\neq \infty$  using le fe unfolding sp
      by (auto simp: top.extremum-unique)
  qed
  show finite-measure  $\nu2$ 
  proof (rule finite-measureI)
    have le: emeasure  $\nu2$  (space  $\nu2$ )  $\leq$  emeasure  $\eta$  (space  $\nu2$ )
      using le-measureD3[OF assms(6)] assms(3,4)
      by simp
    have sp: space  $\nu2 = \text{space } \eta$  using sets-eq-imp-space-eq assms(3,4)
      by auto
    show emeasure  $\nu2$  (space  $\nu2$ )  $\neq \infty$  using le fe unfolding sp
      by (auto simp: top.extremum-unique)
  qed
qed
have E: logEnergyC  $\nu1$   $\nu1 - 2 * \text{logEnergyC } \nu1 \nu2 + \text{logEnergyC } \nu2 \nu2 =$ 
  Eform  $\nu1 \nu2 (\lambda p. \ln (\text{norm} (\text{fst } p - \text{snd } p)))$ 
proof –
  have i: integrable ( $\mu \otimes_M \nu$ ) ( $\lambda p. \ln (\text{norm} (\text{fst } p - \text{snd } p))$ )
    if  $\mu \leq \eta$   $\nu \leq \eta$  sets  $\mu = \text{sets borel}$  sets  $\nu =$ 
      sets borel for  $\mu$   $\nu$  :: complex measure
  proof (rule integrable-le-measure[OF pair-measure-mono[OF that assms(4,1)])

```

```

- assms(8)])
  show sets ( $\mu \otimes_M \nu$ ) = sets ( $\eta \otimes_M \eta$ )
    using that assms(4)
    using sets-pair-measure-cong by simp
qed
have e11: logEnergyC  $\nu 1 \nu 1$  =
  integralL ( $\nu 1 \otimes_M \nu 1$ ) ( $\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))$ )
  by (rule logEnergyC-prod[OF f1 f1 i[OF assms(5,5,2,2)]])
have e12: logEnergyC  $\nu 1 \nu 2$  =
  integralL ( $\nu 1 \otimes_M \nu 2$ ) ( $\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))$ )
  by (rule logEnergyC-prod[OF f1 f2 i[OF assms(5,6,2,3)]])
have e22: logEnergyC  $\nu 2 \nu 2$  =
  integralL ( $\nu 2 \otimes_M \nu 2$ ) ( $\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))$ )
  by (rule logEnergyC-prod[OF f2 f2 i[OF assms(6,6,3,3)]])
show ?thesis unfolding e11 e12 e22 Eform-def ..
qed
have lim: (( $\lambda n. \text{Eform } \nu 1 \nu 2 (Kn\ n)$ )  $\longrightarrow \text{Eform } \nu 1 \nu 2$ 
  ( $\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))$ )) at-top
  unfolding Eform-def
  by (intro tendsto-intros Kn-integral-tendsto[OF pair-measure-mono[OF - - -
    assms(4,1)]] assms)
  (simp-all add: assms sets-pair-measure)
have nonpos:  $\bigwedge n. \text{Eform } \nu 1 \nu 2 (Kn\ n) \leq 0$ 
  by (rule Eform-Kn-nonpos[OF f1 f2 assms(2,3,7)])
show ?thesis unfolding E by (rule tendsto-upperbound[OF lim]) (simp-all add:
nonpos)
qed

end

theory Andreief-Identity
  imports Jordan-Normal-Form.Determinant HOL-Analysis.Analysis
begin

```

13 Andreief's identity (continuous Cauchy–Binet)

For a σ -finite measure μ and functions φ_i, ψ_j with $\varphi_i \psi_j$ integrable, $\det [\int \varphi_i \psi_j d\mu] = (1/n!) \int \det [\varphi_a(x_b)] \det [\psi_a(x_b)] d\mu^{\otimes n}$, with the determinants of the Jordan_Normal_Form library and the product measure Pi_M ; also the transposition and column-scaling rules for determinants used with it.

The general continuous Cauchy-Binet (Andreief) identity: $\det [\int \varphi_i \psi_j d\mu] = (1/n!) \int \det [\varphi_a(x_b)] \det [\psi_a(x_b)] d\mu^{\otimes n}$. Proof: Leibniz expansion, symmetrisation over permutations, Fubini on Pi_M .

theorem *andreief*:

fixes $\mu :: \text{real measure}$ **and** $\varphi \psi :: \text{nat} \Rightarrow \text{real} \Rightarrow \text{real}$ **and**
 $n :: \text{nat}$
assumes *sf*: *sigma-finite-measure* μ

and *integ*: $\bigwedge i j. i < n \implies j < n \implies$
 $\text{integrable } \mu (\lambda x. \varphi i x * \psi j x)$
shows $\text{Determinant.det } (\text{Matrix.mat } n n (\lambda(i,$
 $j). \text{integral}^L \mu (\lambda x. \varphi i x * \psi j x))) =$
 $1 / \text{fact } n * \text{integral}^L (\text{PiM } \{..<n\} (\lambda-. \mu))$
 $(\lambda x. \text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). \varphi a (x b))) *$
 $\text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). \psi a (x b))))$
proof –
define $G :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real}$ **where** $G =$
 $(\lambda i j. \text{integral}^L \mu (\lambda x. \varphi i x * \psi j x))$
define $P :: (\text{nat} \Rightarrow \text{nat}) \text{ set}$ **where** $P = \{p. p \text{ permutes } \{0..<n\}\}$
interpret *product-sigma-finite* $\lambda::\text{nat}. \mu$
by (*rule product-sigma-finite.intro*) (*rule sf*)
have *cardP*: $\text{card } P = \text{fact } n$
unfolding *P-def* **by** (*simp add: card-permutations*)
have *inP*: $\bigwedge \sigma a. \sigma \in P \implies a < n \implies$
 $\sigma a < n$
by (*simp add: P-def permutes-in-image*)
have *permP*: $\bigwedge \sigma. \sigma \in P \implies \sigma \text{ permutes } \{..<n\}$
by (*simp add: P-def atLeast0LessThan*)
have *permutationP*: $\bigwedge \sigma. \sigma \in P \implies \text{permutation } \sigma$
by (*auto simp: P-def permutes-imp-permutation intro: permutes-imp-permutation*)

Leibniz formula in column form: $\det (f a b)\text{-}\{a,b\} = \sum -\sigma \text{sgn } \sigma \prod -a f (\sigma$
 $a) a.$
have *colform*: $\text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). f a b)) = (\sum \sigma \in$
 $P. \text{signof } \sigma * (\prod a < n. f (\sigma a) a))$
for $f :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real}$
proof –
have $\text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). f a b)) =$
 $\text{Determinant.det } (\text{transpose-mat } (\text{Matrix.mat } n n (\lambda(a, b). f a b)))$
by (*simp add: Determinant.det-transpose[OF mat-carrier]*)
also have $\dots = (\sum \sigma \in P. \text{signof } \sigma * (\prod a =$
 $0..<n. \text{transpose-mat } (\text{Matrix.mat } n n (\lambda(a, b). f a b)) \$\$ (a, \sigma a)))$
by (*simp add: det-def'[of - n] P-def*)
also have $\dots = (\sum \sigma \in P. \text{signof } \sigma * (\prod a <$
 $n. f (\sigma a) a))$
by (*simp add: inP atLeast0LessThan*)
finally show *?thesis* .
qed
have *detA*: $\bigwedge x. \text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). \varphi a (x b))) =$
 $(\sum \sigma \in P. \text{signof } \sigma * (\prod a < n. \varphi (\sigma a) (x a)))$
by (*simp add: colform*)
have *detB*: $\bigwedge x. \text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). \psi a (x b))) =$
 $(\sum \tau \in P. \text{signof } \tau * (\prod a < n. \psi (\tau a) (x a)))$
by (*simp add: colform*)
have *leib*: $\bigwedge x. \text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a,$
 $b). \varphi a (x b))) * \text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). \psi a (x b)))$
 $= (\sum \sigma \in P. \sum \tau \in$

$P. \text{signof } \sigma * \text{signof } \tau * (\prod a <$
 $n. \varphi (\sigma a) (x a) * \psi (\tau a) (x a)))$
unfolding *detA detB sum-product prod.distrib by (simp add: mult-ac)*

Integrate each term with Fubini on the finite product.

have *integ-a*: $\bigwedge \sigma \tau a. \sigma \in P \implies$
 $\tau \in P \implies a < n \implies$
 $\text{integrable } \mu (\lambda x. \varphi (\sigma a) x * \psi (\tau a) x)$
using *integ inP by simp*
have *intg*: $\bigwedge \sigma \tau. \sigma \in P \implies$
 $\tau \in P \implies \text{integrable } (PiM \{..<n\} (\lambda-. \mu)) (\lambda x. \prod a <$
 $\prod a < n. \varphi (\sigma a) (x a) * \psi (\tau a) (x a))$
proof –
fix $\sigma \tau$ **assume** $s: \sigma \in P$ **and** $t: \tau \in P$
show $\text{integrable } (PiM \{..<n\} (\lambda-. \mu)) (\lambda x. \prod a <$
 $n. \varphi (\sigma a) (x a) * \psi (\tau a) (x a))$
using *product-integrable-prod[of {..<n} $\lambda a x. \varphi (\sigma a) x *$*
 $\psi (\tau a) x] *integ-a[OF s t] by simp*
qed
have *int*: $\bigwedge \sigma \tau. \sigma \in P \implies$
 $\tau \in P \implies$
 $\text{integral}^L (PiM \{..<n\} (\lambda-. \mu)) (\lambda x. \prod a <$
 $n. \varphi (\sigma a) (x a) * \psi (\tau a) (x a)) = (\prod a <$
 $n. G (\sigma a) (\tau a))$
proof –
fix $\sigma \tau$ **assume** $s: \sigma \in P$ **and** $t: \tau \in P$
show $\text{integral}^L (PiM \{..<n\} (\lambda-. \mu)) (\lambda x. \prod a <$
 $n. \varphi (\sigma a) (x a) * \psi (\tau a) (x a)) = (\prod a <$
 $n. G (\sigma a) (\tau a))$
using *product-integral-prod[of {..<n} $\lambda a x. \varphi (\sigma a) x *$*
 $\psi (\tau a) x] *integ-a[OF s t] by (simp add: G-def)*
qed
have *E*: $\text{integral}^L (PiM \{..<n\} (\lambda-. \mu)) (\lambda x. \text{Determinant.det}$
 $(\text{Matrix.mat } n n (\lambda(a, b). \varphi a (x b))) *$
 $\text{Determinant.det } (\text{Matrix.mat } n n (\lambda(a, b). \psi a (x b))))$
 $= (\sum \sigma \in P. \sum \tau \in$
 $P. \text{signof } \sigma * \text{signof } \tau * (\prod a < n. G (\sigma a) (\tau a)))$
proof –
have *i1*: $\bigwedge \sigma \tau. \sigma \in P \implies$
 $\tau \in P \implies \text{integrable } (PiM \{..<n\} (\lambda-. \mu)) (\lambda x.$
 $\text{signof } \sigma * \text{signof } \tau * (\prod a <$
 $n. \varphi (\sigma a) (x a) * \psi (\tau a) (x a)))$
using *intg by simp*
have *i2*: $\bigwedge \sigma. \sigma \in P \implies$
 $\text{integrable } (PiM \{..<n\} (\lambda-. \mu)) (\lambda x. \sum \tau \in$
 $P. \text{signof } \sigma * \text{signof } \tau * (\prod a <$
 $n. \varphi (\sigma a) (x a) * \psi (\tau a) (x a)))$
using *i1 by (auto intro: Bochner-Integration.integrable-sum)*
have $\text{integral}^L (PiM \{..<n\} (\lambda-. \mu)) (\lambda x. \text{Determinant.det } (\text{Matrix.mat}$$$

```

n n (λ(a, b). φ a (x b))) * Determinant.det (Matrix.mat n n (λ(a,
b). ψ a (x b))))
= integralL (PiM {..n} (λ-. μ)) (λx. ∑ σ ∈
P. ∑ τ ∈ P. signof σ * signof τ * (∏ a <
n. φ (σ a) (x a) * ψ (τ a) (x a)))
by (simp add: leib)
also have ... = (∑ σ ∈ P. ∑ τ ∈
P. integralL (PiM {..n} (λ-. μ)) (λx. signof σ *
signof τ * (∏ a < n. φ (σ a) (x a) * ψ (τ a) (x a))))
proof -
have integralL (PiM {..n} (λ-. μ)) (λx. ∑ σ ∈
P. ∑ τ ∈ P. signof σ * signof τ * (∏ a <
n. φ (σ a) (x a) * ψ (τ a) (x a)))
= (∑ σ ∈ P. integralL (PiM {..n} (λ-. μ)) (λx.
∑ τ ∈ P. signof σ * signof τ * (∏ a <
n. φ (σ a) (x a) * ψ (τ a) (x a))))
using i2 by (auto intro: Bochner-Integration.integral-sum)
also have ... = (∑ σ ∈ P. ∑ τ ∈
P. integralL (PiM {..n} (λ-. μ)) (λx. signof σ *
signof τ * (∏ a < n. φ (σ a) (x a) * ψ (τ a) (x a))))
using i1 by (simp add: Bochner-Integration.integral-sum)
finally show ?thesis .
qed
also have ... = (∑ σ ∈ P. ∑ τ ∈
P. signof σ * signof τ * (∏ a < n. G (σ a) (τ a)))
proof (rule sum.cong[OF refl], rule sum.cong[OF refl])
fix σ τ assume s: σ ∈ P and t: τ ∈ P
show integralL (PiM {..n} (λ-. μ)) (λx. signof σ *
signof τ * (∏ a < n. φ (σ a) (x a) * ψ (τ a) (x a))) =
signof σ * signof τ * (∏ a < n. G (σ a) (τ a))
using int[OF s t] by simp
qed
finally show ?thesis .
qed

```

For fixed τ , substitute $\varrho = \sigma \circ \text{Hilbert-Choice.inv } \tau$: the inner sum is $\det G$.

```

have reidx: ∧σ τ. σ ∈ P ⇒
τ ∈ P ⇒ (∏ a < n. G (σ a) (τ a)) = (∏ b <
n. G ((σ ∘ Hilbert-Choice.inv τ) b) b)
proof -
fix σ τ assume s: σ ∈ P and t: τ ∈ P
show (∏ a < n. G (σ a) (τ a)) = (∏ b <
n. G ((σ ∘ Hilbert-Choice.inv τ) b) b)
using prod.permute[OF permP[OF t],
of λb. G ((σ ∘ Hilbert-Choice.inv τ) b) b] permutes-inverses[OF
permP[OF t]] by (simp add: o-def)
qed
have signmul: ∧σ τ. σ ∈ P ⇒
τ ∈ P ⇒ (signof σ :: real) * signof τ =

```

```

    signof ( $\sigma \circ \text{Hilbert-Choice.inv } \tau$ )
  proof -
    fix  $\sigma \tau$  assume  $s: \sigma \in P$  and  $t: \tau \in P$ 
    show ( $\text{signof } \sigma :: \text{real}$ ) *  $\text{signof } \tau =$ 
       $\text{signof } (\sigma \circ \text{Hilbert-Choice.inv } \tau)$ 
    using sign-compose[OF permutationP[OF s] permutation-inverse[OF permutationP[OF t]]]
      sign-inverse[OF permutationP[OF t]] by simp
  qed
  have bij:  $\bigwedge \tau. \tau \in P \implies$ 
     $\text{bij-betw } (\lambda \sigma. \sigma \circ \text{Hilbert-Choice.inv } \tau) P P$ 
  proof -
    fix  $\tau$  assume  $t: \tau \in P$ 
    have tp:  $\tau$  permutes  $\{0..<n\}$  using t by (simp add: P-def)
    have h1:  $(\lambda \sigma. \sigma \circ \text{Hilbert-Choice.inv } \tau) \in$ 
       $P \rightarrow P$ 
    using permutes-compose[OF permutes-inv[OF tp]] by (simp add: P-def)
    have h2:  $(\lambda \varrho. \varrho \circ \tau) \in P \rightarrow P$ 
    using permutes-compose[OF tp] by (simp add: P-def)
    have h3:  $\bigwedge \sigma. \sigma \in P \implies$ 
       $\sigma \circ \text{Hilbert-Choice.inv } \tau \circ \tau = \sigma$ 
    by (simp add: o-assoc[symmetric] permutes-inv-o[OF tp])
    have h4:  $\bigwedge \varrho. \varrho \in P \implies$ 
       $\varrho \circ \tau \circ \text{Hilbert-Choice.inv } \tau = \varrho$ 
    by (simp add: o-assoc[symmetric] permutes-inv-o[OF tp])
    show bij-betw  $(\lambda \sigma. \sigma \circ \text{Hilbert-Choice.inv } \tau) P P$ 
    using bij-betwI[OF h1 h2] h3 h4 by simp
  qed
  have inner:  $\bigwedge \tau. \tau \in P \implies$ 
     $(\sum \sigma \in P. \text{signof } \sigma * \text{signof } \tau * (\prod a <$ 
       $n. G (\sigma a) (\tau a))) = \text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(i, j). G i j))$ 
  proof -
    fix  $\tau$  assume  $t: \tau \in P$ 
    have  $(\sum \sigma \in P. \text{signof } \sigma * \text{signof } \tau * (\prod a <$ 
       $n. G (\sigma a) (\tau a))) = (\sum \sigma \in$ 
       $P. \text{signof } (\sigma \circ \text{Hilbert-Choice.inv } \tau) * (\prod b <$ 
       $n. G ((\sigma \circ \text{Hilbert-Choice.inv } \tau) b) b))$ 
    by (simp add: reidx[OF - t] signmul[OF - t])
    also have  $\dots = (\sum \varrho \in P. \text{signof } \varrho * (\prod b < n. G (\varrho b) b))$ 
    using sum.reindex-bij-betw[OF bij[OF t], of  $\lambda \varrho. \text{signof } \varrho * (\prod b <$ 
       $n. G (\varrho b) b)$ ] by simp
    also have  $\dots = \text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(i, j). G i j))$ 
    using colform[of G] by simp
    finally show  $(\sum \sigma \in P. \text{signof } \sigma * \text{signof } \tau * (\prod a <$ 
       $n. G (\sigma a) (\tau a))) = \text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(i, j). G i j))$  .
  qed
  have comb:  $(\sum \sigma \in P. \sum \tau \in$ 
     $P. \text{signof } \sigma * \text{signof } \tau * (\prod a < n. G (\sigma a) (\tau a))) =$ 
     $\text{fact } n * \text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(i, j). G i j))$ 

```

```

proof -
  have ( $\sum \sigma \in P. \sum \tau \in$ 
     $P. \text{signof } \sigma * \text{signof } \tau * (\prod a < n. G (\sigma a) (\tau a))$ ) =
    ( $\sum \tau \in P. \sum \sigma \in$ 
     $P. \text{signof } \sigma * \text{signof } \tau * (\prod a < n. G (\sigma a) (\tau a))$ )
    using sum.swap by fast
  also have ... = ( $\sum \tau \in$ 
     $P. \text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(i, j). G \ i \ j))$ )
    by (simp add: inner)
  also have ... = fact n *  $\text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(i, j). G \ i \ j))$ 
    by (simp add: cardP)
  finally show ?thesis .
qed
have fn: (fact n :: real)  $\neq 0$  by simp
show ?thesis
  unfolding G-def [symmetric] E comb using fn by simp
qed

(6.10):  $\Delta\text{-}K(\zeta(5)) = (1/h!) \int_{\{(0,\infty)^h\}} \prod_{\{i < j\}} (y_i^2 - y_j^2)^2 \prod_i D\text{-}N(y_i^2)^6 / D\text{-}K(y_i^2) w(y_i) dy.$ 

lemma andreief-det-swap:
  fixes f :: nat  $\Rightarrow$  nat  $\Rightarrow$  'a :: comm-ring-1
  shows  $\text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ b \ a)) =$ 
     $\text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ a \ b))$ 
proof -
  have  $\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ b \ a) =$ 
     $\text{transpose-mat } (\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ a \ b))$ 
    by (auto intro: eq-matI)
  thus ?thesis
    using  $\text{Determinant.det-transpose}$ [OF mat-carrier] by auto
qed

lemma andreief-det-scale-cols:
  fixes f :: nat  $\Rightarrow$  nat  $\Rightarrow$  'a :: comm-ring-1 and c :: nat  $\Rightarrow$  'a
  shows  $\text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ a \ b * c \ b)) = (\prod b <$ 
     $n. c \ b) * \text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ a \ b))$ 
proof -
  have inP:  $\bigwedge p \ j. p \text{ permutes } \{0..<n\} \Longrightarrow j < n \Longrightarrow p \ j < n$ 
    by (simp add: permutes-in-image)
  have d1:  $\text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ a \ b * c \ b)) =$ 
    ( $\sum p \mid p \text{ permutes } \{0..<n\}. \text{signof } p * (\prod j < n. f \ (p \ j) \ j * c \ j)$ )
    by (simp add: det-col[OF mat-carrier] inP)
  have d2:  $\text{Determinant.det } (\text{Matrix.mat } n \ n \ (\lambda(a, b). f \ a \ b)) =$ 
    ( $\sum p \mid p \text{ permutes } \{0..<n\}. \text{signof } p * (\prod j < n. f \ (p \ j) \ j)$ )
    by (simp add: det-col[OF mat-carrier] inP)
  show ?thesis
    unfolding d1 d2 prod.distrib sum-distrib-left
    by (simp add: mult-ac)
qed

```


end

theory *Zeta5-Definitions*

imports *Rational-Padic-Valuations Arcsine-Distribution Logarithmic-Energy*
Zeta-Function.Zeta-Function
Kummer-Congruence.Rat-Congruence
Jordan-Normal-Form.Determinant
Jordan-Normal-Form.DL-Rank
HOL-Computational-Algebra.Polynomial-FPS
HOL-Library.Log-Nat

begin

14 Definitions of the construction

All definitions of the proof, transcribed from the Lean 4 theory *Defs* of the domino14/zeta5 formalisation, which fixed the conventions used throughout:

- A rational function in t is a numerator polynomial together with a finite set of simple poles: $\text{mu}X\ S\ P$ is $\mu_X(P(t) / \prod_{j \in S} (t + j^2))$ and $\text{tau}X\ S\ P$ is $\tau_X(P(x) / \prod_{r \in S} (x - r))$. Values are polynomials in $\mathbb{Q}[X]$; the indeterminate X is later specialised to $\zeta(5)$.
- The p -adic field $\mathbb{Q}_p$ of the paper's Sections 3.1–3.2 is not used. Every p -adically convergent series is replaced by its partial sum of length M (the *-tr* definitions) and the identities become congruences modulo high powers of p , so that the whole development stays in HOL over $\mathbb{Q}$.
- Matrices are those of the *Jordan_Normal_Form* library, indexed by explicit lists of index pairs instead of Lean's Σ -types. Equation and section numbers refer to Fauzan's paper.

14.1 Generic helpers

Evaluation of a rational polynomial at a real (Lean: *aeval* $x\ P$).

definition *real-eval* :: *rat poly* $\Rightarrow$ *real* $\Rightarrow$ *real* **where**
real-eval $P\ x = \text{poly } (\text{map-poly of-rat } P)\ x$

14.2 Section 2.1 Parameters (2.1)

definition *Nof* :: *nat* $\Rightarrow$ *nat* **where** *Nof* $K = 3 * K \text{ div } 40$
definition *hof* :: *nat* $\Rightarrow$ *nat* **where** *hof* $K = 37 * K \text{ div } 40$
definition *alpha-q* :: *rat* **where** *alpha-q* = $3 / 40$
definition *lam-q* :: *rat* **where** *lam-q* = $37 / 40$
definition *Hc-q* :: *rat* **where** *Hc-q* = $1 + 2 * \text{alpha-q}$

The real number $\zeta(5)$; *zeta* is the AFP's Riemann zeta function.

definition $\text{zeta5} :: \text{real}$ **where** $\text{zeta5} = \text{Re } (\text{zeta } 5)$

$$H\text{-}j^{\wedge}\{(5)\} = \sum -\{v=1\}^{\wedge} j \text{ } v^{\wedge}\{-5\}.$$

definition $H5 :: \text{nat} \Rightarrow \text{rat}$ **where** $H5 \text{ } j = (\sum \text{ } v = 1..j. \text{ } 1 \text{ } / \text{ } \text{of-nat } v^{\wedge} 5)$

$$D\text{-}m(t) = \prod -\{j=1\}^{\wedge} m \text{ } (t + j^{\wedge} 2).$$

definition $Dm :: \text{nat} \Rightarrow \text{rat poly}$ **where** $Dm \text{ } m = (\prod \text{ } j = 1..m. \text{ } pX + [: \text{of-nat } j^{\wedge} 2 :])$

14.3 Section 2.1 The functional mu_X (2.2), (2.3)

(2.2): $\text{mu}(t^{\wedge} e) = (-1)^{\wedge} e \text{ } B\text{-}\{2e+2\} (2e+3)(2e+4)(2e+5)/24$, with $B\text{-}1 = -1/2$ (both Mathlib's *bernoulli* and the AFP's polymorphic *bernoulli* use this convention).

definition $\text{muMono} :: \text{nat} \Rightarrow \text{rat}$ **where**
 $\text{muMono } e = (-1)^{\wedge} e * \text{bernoulli } (2 * e + 2) * ((2 * \text{of-nat } e + 3) * (2 * \text{of-nat } e + 4) * (2 * \text{of-nat } e + 5)) / 24$

definition $\text{muPoly} :: \text{rat poly} \Rightarrow \text{rat}$ **where**
 $\text{muPoly } P = (\sum \text{ } e \leq \text{degree } P. \text{ } \text{Polynomial.coeff } P \text{ } e * \text{muMono } e)$

$$(2.3): \text{mu}\text{-}X(1/(t+j^{\wedge} 2)) = j^{\wedge} 4 (X - H\text{-}j) - 1/4 + 1/(2j).$$

definition $\text{muPole} :: \text{nat} \Rightarrow \text{rat poly}$ **where**
 $\text{muPole } j = [: \text{of-nat } j^{\wedge} 4 :] * (pX - [: H5 \text{ } j :]) - [: 1 / 4 :] + [: 1 / (2 * \text{of-nat } j) :]$

definition $\text{sqPoleProd} :: \text{nat set} \Rightarrow \text{rat poly}$ **where**
 $\text{sqPoleProd } S = (\prod \text{ } j \in S. \text{ } pX + [: \text{of-nat } j^{\wedge} 2 :])$

Residue of $P/\prod -\{k \in S\}(t+k^{\wedge} 2)$ at $t = -j^{\wedge} 2$.

definition $\text{sqResidue} :: \text{nat set} \Rightarrow \text{rat poly} \Rightarrow \text{nat} \Rightarrow \text{rat}$ **where**
 $\text{sqResidue } S \text{ } P \text{ } j = \text{poly } P \text{ } (- (\text{of-nat } j^{\wedge} 2)) / (\prod \text{ } k \in S - \{j\}. \text{ } \text{of-nat } k^{\wedge} 2 - (\text{of-nat } j^{\wedge} 2))$

$\text{mu}\text{-}X(P(t)/\prod -\{j \in S\}(t+j^{\wedge} 2))$ by polynomial division and simple partial fractions (meaningful for finite S with $0 \notin S$).

definition $\text{muX} :: \text{nat set} \Rightarrow \text{rat poly} \Rightarrow \text{rat poly}$ **where**
 $\text{muX } S \text{ } P = [: \text{muPoly } (P \text{ div } \text{sqPoleProd } S) :] + (\sum \text{ } j \in S. \text{ } [: \text{sqResidue } S \text{ } P \text{ } j :]) * \text{muPole } j)$

14.4 Section 2.1 The matrix, determinant and normalisation (2.4), (2.5)

$$(2.4): G\text{-}K(X) = [\text{mu}\text{-}X(D\text{-}N(t)^{\wedge} 6 \text{ } t^{\wedge}\{i+j\} \text{ } / \text{ } D\text{-}K(t))] \text{-}\{0 \leq i, j < h\}.$$

definition $GK :: \text{nat} \Rightarrow \text{rat poly mat}$ **where**
 $GK \text{ } K = \text{Matrix.mat } (\text{hof } K) (\text{hof } K) (\lambda(i, j). \text{muX } \{1..K\} (Dm (\text{Nof } K)^{\wedge} 6 * pX^{\wedge}(i + j)))$

definition $\Delta K :: \text{nat} \Rightarrow \text{rat poly}$ **where** $\Delta K K = \text{Determinant.det } (GK K)$

$$(2.5): S-K = (K!) \cdot 4^{h-1} / ((N!) \cdot 12^h \prod_{i=1}^{h-1} ((2i)! \cdot 2)).$$

definition $SK :: \text{nat} \Rightarrow \text{rat}$ **where**

$$SK K = (\text{fact } K \cdot (2 * \text{hof } K) * 4^{(\text{hof } K - 1)}) / (\text{fact } (\text{Nof } K) \cdot (12 * \text{hof } K) * (\prod_{i=1}^{\text{hof } K - 1} \text{fact } (2 * i) \cdot 2))$$

definition $FK :: \text{nat} \Rightarrow \text{rat poly}$ **where** $FK K = [\text{SK } K] * \Delta K K$

14.5 Section 3 The functional τ_X

$L(x^k) = B \cdot k$, extended linearly.

definition $Lfun :: \text{rat poly} \Rightarrow \text{rat}$ **where**

$$Lfun P = (\sum_{k \leq \text{degree } P} \text{Polynomial.coeff } P k * \text{bernoulli } k)$$

$\tau(P) = L(P''')/24$ on polynomials.

definition $\tau :: \text{rat poly} \Rightarrow \text{rat}$ **where**

$$\tau P = Lfun ((\text{pderiv } 3) P) / 24$$

$d(r) = r$ for $r \geq 0$ and $-r-1$ for $r < 0$.

definition $dIdx :: \text{int} \Rightarrow \text{nat}$ **where**

$$dIdx r = (\text{if } 0 \leq r \text{ then nat } r \text{ else nat } (-r-1))$$

$$\tau_X(1/(x-r)) = H\{d(r)\} - X.$$

definition $\tau Pole :: \text{int} \Rightarrow \text{rat poly}$ **where**

$$\tau Pole r = [\text{H5 } (dIdx r)] - pX$$

definition $\text{linPoleProd} :: \text{int set} \Rightarrow \text{rat poly}$ **where**

$$\text{linPoleProd } S = (\prod_{r \in S} pX - [\text{of-int } r])$$

definition $\text{linResidue} :: \text{int set} \Rightarrow \text{rat poly} \Rightarrow \text{int} \Rightarrow \text{rat}$ **where**

$$\text{linResidue } S P r = \text{poly } P (\text{of-int } r) / (\prod_{s \in S - \{r\}} \text{of-int } s - \text{of-int } s)$$

$\tau_X(P(x)/\prod_{r \in S}(x-r))$ by polynomial division and simple partial fractions.

definition $\tau X :: \text{int set} \Rightarrow \text{rat poly} \Rightarrow \text{rat poly}$ **where**

$$\tau X S P = [\tau (P \text{ div } \text{linPoleProd } S)] + (\sum_{r \in S} [\text{linResidue } S P r] * \tau Pole r)$$

The fourth Taylor coefficient $g'''(0)/24$ of $g = P/Q$ at 0 (needs $Q(0) \neq 0$).

definition $\text{taylor4} :: \text{rat poly} \Rightarrow \text{rat poly} \Rightarrow \text{rat}$ **where**

$$\text{taylor4 } P Q = \text{fps-nth } (\text{fps-of-poly } P * \text{inverse } (\text{fps-of-poly } Q)) 4$$

The poles $\pm j, j \in S$ (Lean: signedPoles).

definition *signedPoles* :: *nat set* $\Rightarrow$ *int set* **where**
signedPoles *S* = *int* ‘ *S* $\cup$ ($\lambda j.$ – *int* *j*) ‘ *S*

The poles $-K \leq r \leq K$, $r \neq 0$, of Lemma 3.3.

definition *symPoles* :: *nat* $\Rightarrow$ *int set* **where**
symPoles *K* = {*r.* – *int* *K* $\leq$ *r* $\wedge$ *r* $\leq$ *int* *K* $\wedge$ *r* $\neq$ 0}

A is *p*-integer-valued on $\mathbb{Z}$ (Lean: *IntValuedAt*).

definition *IntValuedAt* :: *nat* $\Rightarrow$ *rat poly* $\Rightarrow$ *bool* **where**
IntValuedAt *p* *A* $\longleftrightarrow$
 $(\forall x::\text{int. } \text{poly } A \text{ (of-int } x) \neq 0 \longrightarrow$
 $0 \leq \text{vp } p \text{ (poly } A \text{ (of-int } x)))$

14.6 Section 3.1-3.2 The *p*-adic extension, in truncated rational form

$\kappa\text{-}d = d(d-1)(d-2) \dots B\text{-}\{d-3\}/24$, so that $\tau(z^{\wedge}d) = \kappa\text{-}d$; it is 0 for $d < 3$.

definition *kappa* :: *nat* $\Rightarrow$ *rat* **where**
kappa *d* = *of-nat* *d* * (*of-nat* *d* – 1) * (*of-nat* *d* – 2) * *bernoulli* (*d* – 3) / 24

τ on polynomials through κ (equal to *tau*).

definition *tauK* :: *rat poly* $\Rightarrow$ *rat* **where**
tauK *Q* = ($\sum d \leq \text{degree } Q. \text{Polynomial.coeff } Q \text{ } d * \text{kappa } d$)

(3.5), truncated: the partial sum of length *M* of $\tau \text{ } \{an\} (1/(z-s)) = -(1/4) \sum_{k \geq 0} C(k+3,3) B\text{-}k \text{ } s^{\wedge}\{-k-4\}$ (for $v\text{-}p(s) < 0$ the *k*-th term has valuation $\geq k+3$, so the tail is $\equiv 0 \text{ mod } p^{\wedge}\{M+3\}$).

definition *tauAnPole-tr* :: *nat* $\Rightarrow$ *rat* $\Rightarrow$ *nat* $\Rightarrow$ *rat* **where**
tauAnPole-tr *p* *s* *M* =
 $-(1/4) * (\sum k < M. \text{of-nat } ((k+3) \text{ choose } 3) * \text{bernoulli } k * s \text{ powi } (-$
 $(\text{int } k + 4)))$

(3.6), truncated: $C\text{-}p = \sum_{a=1} \{a=1\} \{p-1\} \tau \text{ } \{an\} (1/(x + a/p))$.

definition *Cp-tr* :: *nat* $\Rightarrow$ *nat* $\Rightarrow$ *rat* **where**
Cp-tr *p* *M* = ($\sum a = 1..p - 1. \text{tauAnPole-tr } p \text{ } (- \text{of-nat } a / \text{of-nat } p) \text{ } M$)

(3.6): $Y = p^{\wedge}5 X + C\text{-}p$.

definition *Yp-tr* :: *nat* $\Rightarrow$ *nat* $\Rightarrow$ *rat poly* **where**
Yp-tr *p* *M* = [*of-nat* $p^{\wedge}5$] * *pX* + [*Cp-tr* *p* *M*]

Value of $\tau\text{-}Y^{\wedge}\{ext\}$ on the simple pole $1/(z-s)$: near (integer) poles take $H\text{-}\{d(s)\} - Y$, far poles the truncated series.

definition *tauPoleP-tr* :: *nat* $\Rightarrow$ *rat poly* $\Rightarrow$ *rat* $\Rightarrow$ *nat*
 $\Rightarrow$ *rat poly* **where**
tauPoleP-tr *p* *Y* *s* *M* =
 $(\text{if } \text{snd } (\text{quotient-of } s) = 1 \text{ then } [\text{H5 } (dIdx (\text{fst } (\text{quotient-of } s)))] - Y$

else [:tauAnPole-tr p s M:])

definition $qPoleProd :: \text{rat set} \Rightarrow \text{rat poly}$ **where**
 $qPoleProd S = (\prod s \in S. pX - [:s:])$

definition $qRes :: \text{rat set} \Rightarrow \text{rat poly} \Rightarrow \text{rat} \Rightarrow \text{rat}$ **where**
 $qRes S P s = \text{poly } P s / (\prod s' \in S - \{s\}. s - s')$

$\tau\text{-}Y^{\sim\{ext\}}(U(z)/\prod -\{s \in S\}(z-s))$ for distinct rational poles S , truncated at M .

definition $\text{tauExt-tr} :: \text{nat} \Rightarrow \text{rat poly} \Rightarrow \text{rat poly} \Rightarrow \text{rat set}$
 $\Rightarrow \text{nat} \Rightarrow \text{rat poly}$ **where**
 $\text{tauExt-tr } p Y U S M =$
 $[:\text{tauK } (U \text{ div } qPoleProd S):] + (\sum s \in S. [:qRes S U s:] * \text{tauPoleP-tr } p Y s$
 $M)$

$g(a + p x)$ written as $U(x)/\prod (x - s)$: the numerator $p^{\sim\{-|S|\}} P(a + p x)$.

definition $\text{distribNum} :: \text{nat} \Rightarrow \text{int set} \Rightarrow \text{rat poly} \Rightarrow \text{nat}$
 $\Rightarrow \text{rat poly}$ **where**
 $\text{distribNum } p S P a =$
 $[:\text{of-nat } p \text{ powi } (- \text{int } (\text{card } S)):] * \text{pcompose } P ([:\text{of-nat } a:] + [: \text{of-nat } p:] * pX)$

The poles $(r - a)/p$ of $g(a + p x)$.

definition $\text{distribPoles} :: \text{nat} \Rightarrow \text{int set} \Rightarrow \text{nat} \Rightarrow \text{rat set}$ **where**
 $\text{distribPoles } p S a = (\lambda r. (\text{of-int } r - \text{of-nat } a) / \text{of-nat } p) ' S$

The Taylor coefficients of the analytic part $Q + \sum -\{s \in F\} \varrho s / (z-s)$.

definition $fCoeff :: \text{rat poly} \Rightarrow \text{rat set} \Rightarrow (\text{rat} \Rightarrow \text{rat})$
 $\Rightarrow \text{nat} \Rightarrow \text{rat}$ **where**
 $fCoeff Q F \varrho d = \text{Polynomial.coeff } Q d - (\sum s \in F. \varrho s * \text{inverse } s \wedge (d + 1))$

14.7 Section 3.3 Small primes

$q-0 = 1, q-i(t) = (-1)^{\wedge i} 2t D-\{i-1\}(t)/(2i)!$.

definition $qBasis :: \text{nat} \Rightarrow \text{rat poly}$ **where**
 $qBasis i = (\text{if } i = 0 \text{ then } 1 \text{ else } [:(-1)^{\wedge i} * 2 / \text{fact } (2 * i):] * pX * Dm (i - 1))$

$f-i(t) = (D-N(t)/(N!)^{\wedge 2})^{\wedge 3} q-i(t)$.

definition $fBasis :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{rat poly}$ **where**
 $fBasis K i = ([:1 / \text{fact } (\text{Nof } K)^{\wedge 2}:] * Dm (\text{Nof } K))^{\wedge 3} * qBasis i$

14.8 Section 4.1 The inner range

$l\text{-}A(a) = \#\{1 \leq j \leq A : j \equiv \pm a \pmod{p}\}$.

definition $ell :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat$ **where**

$ell\ p\ A\ a = card\ \{j \in \{1..A\}. j\ mod\ p = a\ mod\ p \vee (j + a)\ mod\ p = 0\}$

definition $mHalf :: nat \Rightarrow nat$ **where** $mHalf\ p = (p - 1)\ div\ 2$

definition $L0 :: nat \Rightarrow nat$ **where** $L0\ M = 4 * M + 10$

Right-hand side of (4.4): $h - L-0 + 3(N - m-N)$.

definition $rhs44 :: nat \Rightarrow nat \Rightarrow nat \Rightarrow int$ **where**

$rhs44\ K\ M\ p = int\ (hof\ K) - int\ (L0\ M) + 3 * (int\ (Nof\ K) - int\ (Nof\ K\ div\ p))$

(4.4): T and E (integer division by the positive number $m = (p-1)/2$).

definition $Tin :: nat \Rightarrow nat \Rightarrow nat \Rightarrow int$ **where** $Tin\ K\ M\ p =$

$rhs44\ K\ M\ p\ div\ int\ (mHalf\ p)$

definition $Ein :: nat \Rightarrow nat \Rightarrow nat \Rightarrow int$ **where** $Ein\ K\ M\ p =$

$rhs44\ K\ M\ p\ mod\ int\ (mHalf\ p)$

$b-a = 3\ l-N(a)$.

definition $bIn :: nat \Rightarrow nat \Rightarrow nat \Rightarrow int$ **where** $bIn\ K\ p\ a =$

$3 * int\ (ell\ p\ (Nof\ K)\ a)$

Position of class a in the ordering by decreasing $l-K$, ties by increasing a .

definition $defaultPos :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat$ **where**

$defaultPos\ K\ p\ a = card\ \{c \in \{1..mHalf\ p\}. ell\ p\ K\ a < ell\ p\ K\ c \vee (ell\ p\ K\ c = ell\ p\ K\ a \wedge c < a)\}$

$\varepsilon-a = 1$ for the first E classes in that ordering.

definition $epsIn :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \Rightarrow int$

where

$epsIn\ K\ M\ p\ a =$

$(if\ int\ (card\ \{c \in \{1..mHalf\ p\}. defaultPos\ K\ p\ c < defaultPos\ K\ p\ a\}) < Ein\ K\ M\ p\ then\ 1\ else\ 0)$

$L-a = T - b-a + \varepsilon-a$ for $a \geq 1$, and $L-0 = 4M + 10$.

definition $Lin :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \Rightarrow int$

where

$Lin\ K\ M\ p\ a = (if\ a = 0\ then\ int\ (L0\ M)\ else\ Tin\ K\ M\ p - bIn\ K\ p\ a + epsIn\ K\ M\ p\ a)$

$Z-a = T + \varepsilon-a$.

definition $Zin :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \Rightarrow int$

where $Zin\ K\ M\ p\ a = Tin\ K\ M\ p + epsIn\ K\ M\ p\ a$

(4.6), (4.7): the row weights $w-\{a,i\}$.

definition $wIn :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \Rightarrow nat$

$\Rightarrow rat$ **where**

$wIn\ K\ M\ p\ a\ i =$

$$\begin{aligned}
& \text{(if } a = 0 \text{ then} \\
& \quad \text{Min (insert (2 * of-nat } i + 6 * \text{of-nat (Nof } K \text{ div } p) - \text{of-nat (} K \text{ div } p) + \\
& \quad 1 / 2) \\
& \quad \quad ((\lambda c. \text{ of-int (Zin } K \text{ M } p \text{ c) - (of-nat (ell } p \text{ K } c) + 4) /} \\
& \quad \quad \quad 2) \text{ ' } \{1..m\text{Half } p\})) \\
& \quad \text{else of-nat } i + \text{of-int (bIn } K \text{ p } a) - (\text{of-nat (ell } p \text{ K } a) + 4) / 2)
\end{aligned}$$

$$(4.8): \gamma\text{-}p^{\wedge}in = 2 \sum \text{-}\{a=0\}^{\wedge}m \sum \text{-}\{i < L-a\} w\text{-}\{a,i\}.$$

definition $\text{gammaIn} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{rat}$ **where**
 $\text{gammaIn } K \text{ M } p = 2 * (\sum a \leq m\text{Half } p. \sum i < \text{nat (Lin } K \text{ M } p \text{ a)}. w\text{In } K \text{ M } p \text{ a } i)$

Index list of the inner basis: pairs (a, i) with $0 \leq a \leq m, i < L-a$ (Lean's Σ -type InnerIdx , enumerated).

definition $\text{inner-idx} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow (\text{nat} \times \text{nat}) \text{ list}$ **where**
 $\text{inner-idx } K \text{ M } p = \text{concat (map (}\lambda a. \text{map (}\lambda i. (a, i)) [0..<\text{nat (Lin } K \text{ M } p \text{ a)}]) [0..<m\text{Half } p + 1])$

$$(4.5): E\text{-}\{a,i\}(t) = \prod \text{-}\{c \neq a\} (t + c^{\wedge}2)^{\wedge}\{L-c\} (t + a^{\wedge}2)^{\wedge}i.$$

definition $\text{innerBasis} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \times \text{nat} \Rightarrow \text{rat poly}$ **where**
 $\text{innerBasis } K \text{ M } p \text{ x} =$
 $(\prod c \in \{0..m\text{Half } p\} - \{\text{fst } x\}. (pX + [: \text{of-nat } c^{\wedge}2:])^{\wedge} \text{nat (Lin } K \text{ M } p \text{ c)})$
 $*$
 $(pX + [: \text{of-nat (fst } x)^{\wedge}2:])^{\wedge} \text{snd } x$

The Gram matrix of $\mu\text{-}X(D\text{-}N^{\wedge}6 \cdot / D\text{-}K)$ in the inner basis.

definition $\text{innerGram} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{rat poly mat}$ **where**
 $\text{innerGram } K \text{ M } p = (\text{let idx} = \text{inner-idx } K \text{ M } p \text{ in}$
 $\text{Matrix.mat (length idx) (length idx)}$
 $(\lambda(x, y). \mu X \{1..K\} (Dm (\text{Nof } K)^{\wedge}6 * \text{innerBasis } K \text{ M } p (\text{idx ! } x) * \text{innerBasis } K \text{ M } p (\text{idx ! } y))))$

definition $\text{wInIdx} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \times \text{nat} \Rightarrow \text{rat}$ **where**
 $\text{wInIdx } K \text{ M } p \text{ x} = \text{wIn } K \text{ M } p (\text{fst } x) (\text{snd } x)$

14.9 Section 4.2 The outer range

$r\text{-}p = \max(0, K + 4N - 2p + 2)$ from (4.10) (truncated subtraction).

definition $\text{rOut} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat}$ **where** $\text{rOut } K \text{ p} =$
 $K + 4 * \text{Nof } K + 2 - 2 * p$

$$(4.14): \gamma\text{-}p^{\wedge}out, \text{ with } v = K - p\lfloor K/p \rfloor, u = \max(0, N+v-p+1), t\text{-}p = \min(N, v) + u.$$

definition $\text{gammaOut} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{int}$ **where**

$\text{gammaOut } K \ p = (\text{let } v = K \bmod p; u = \text{Nof } K + v + 1 - p; tp = \min (\text{Nof } K) \ v + u \text{ in}$
 $\quad \text{if } K < 2 * p$
 $\quad \text{then } - 7 * (\text{int } K - \text{int } p) + 6 * \text{int } tp - 1 - \min (\text{int } (rOut \ K \ p)) (\text{int } p$
 $- 1 - \text{int } (\text{Nof } K) +$
 $\quad \text{int } u)$
 $\quad \text{else } - 7 * (\text{int } K - \text{int } p) + 3 + 12 * \text{int } (\text{Nof } K) + 5 * \text{int } tp - \min (\text{int } (rOut \ K \ p)) (\text{int } p +$
 $\quad \text{int } u))$

(4.9): the hypotheses of the outer-range estimate.

definition $\text{OuterHyp} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{bool}$ **where**
 $\text{OuterHyp } K \ p \longleftrightarrow 7 \leq p \wedge p \leq K \wedge K < 3 * p \wedge$
 $2 * K < p \wedge 2 * \text{Nof } K < p \wedge 5 * \text{Nof } K \leq 2 * p - 2$

The poles of $D\text{-tail}$ in the class $\pm a \pmod{p}$ ($a = 0$: multiples of p).

definition $\text{outerClassPoles} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat set}$ **where**
 $\text{outerClassPoles } K \ p \ a = \{j \in \{\text{Nof } K + 1..K\}. j \bmod p = a \bmod p \vee (j + a) \bmod p = 0\}$

definition $\text{outerCount} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat}$ **where**
 $\text{outerCount } K \ p \ a = \text{card } (\text{outerClassPoles } K \ p \ a)$

definition $\text{outer-idx} :: \text{nat} \Rightarrow \text{nat} \Rightarrow (\text{nat} \times \text{nat}) \text{ list}$ **where**
 $\text{outer-idx } K \ p = \text{concat } (\text{map } (\lambda a. \text{map } (\lambda i. (a,$
 $\quad i)) [0..<\text{outerCount } K \ p \ a]) [0..<mHalf \ p + 1])$

(4.11): rows $P\text{-}a \ q\{a, i\}$ with $P\text{-}a = D\text{-tail}/Q\text{-}a$; the zero class is read as $P\text{-}0 \ (t + p \wedge 2) \wedge i$ (ambiguity recorded in the Lean NOTES, finding 1).

definition $\text{outerBasis} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \times$
 $\text{nat} \Rightarrow \text{rat poly}$ **where**
 $\text{outerBasis } K \ p \ x = (\text{let } a = \text{fst } x; i = \text{snd } x;$
 $\quad Pa = \text{sqPoleProd } (\{\text{Nof } K + 1..K\} - \text{outerClassPoles } K \ p \ a) \text{ in}$
 $\quad \text{if } a = 0 \text{ then } Pa * (pX + [:of\text{-nat } p \wedge 2:]) \wedge i$
 $\quad \text{else } (\text{let } l = \text{ell } p \ K \ a \text{ in}$
 $\quad \quad \text{if } i < l - 2 \text{ then } Pa * (pX + [:of\text{-nat } a \wedge 2:]) \wedge i$
 $\quad \quad \text{else } Pa * \text{sqPoleProd } \{j \in \text{outerClassPoles } K \ p \ a. p <$
 $\quad \quad j\} * (pX + [:of\text{-nat } a \wedge 2:]) \wedge (i - (l - 2)))$

(4.12) and the zero-class weights.

definition $\text{wOut} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \times \text{nat} \Rightarrow \text{rat}$ **where**
 $\text{wOut } K \ p \ x = (\text{let } a = \text{fst } x; i = \text{snd } x \text{ in}$
 $\quad \text{if } a = 0 \text{ then } (\text{if } \text{outerCount } K \ p \ 0 = 1 \text{ then } - 1 / 2 \text{ else if } i = 0 \text{ then } - 2$
 $\quad \text{else } 0)$
 $\quad \text{else } (\text{let } l = \text{ell } p \ K \ a; d = (\text{if } a \leq \text{Nof } K \text{ then } 1 \text{ else } 0 :: \text{rat}) \text{ in}$
 $\quad \text{if } i < l - 2 \text{ then } \min 0 \ (\text{of\text{-nat } } i + 3 * d - (\text{of\text{-nat } } l + 4) / 2) \text{ else } 0))$

The integer $c\text{-}e \equiv p \ \mu(t \wedge e) \pmod{p}$ in $[0, p)$, with $c\text{-}e = 0$ for $e < 2p-3$ ($(q \bmod)$ is the AFP's residue of a rational modulo an integer).

definition $cCorr :: nat \Rightarrow nat \Rightarrow rat$ **where**

$cCorr\ p\ e = (if\ e < 2 * p - 3\ then\ 0\ else\ of-int\ ((of-nat\ p * muMono\ e)\ qmod\ int\ p))$

$mu-X$ with the polynomial moments replaced by $mu^{\wedge}0(t^{\wedge}e) = mu(t^{\wedge}e) - c-e/p$.

definition $muXmod :: nat \Rightarrow nat\ set \Rightarrow rat\ poly \Rightarrow rat\ poly$ **where**

$muXmod\ p\ S\ P =$
 $[\sum\ e \leq degree\ (P\ div\ sqPoleProd\ S). Polynomial.coeff\ (P\ div\ sqPoleProd\ S)$
 $e * (muMono\ e - cCorr\ p\ e / of-nat\ p):]$
 $+ (\sum\ j \in S. [sqResidue\ S\ P\ j:] * muPole\ j)$

The matrix A of (4.10).

definition $Aout :: nat \Rightarrow nat \Rightarrow rat\ poly\ mat$ **where**

$Aout\ K\ p = Matrix.mat\ (hof\ K)\ (hof\ K)\ (\lambda(i,$
 $j). muXmod\ p\ \{1..K\}\ (Dm\ (Nof\ K)\ ^6 * pX\ ^{(i+j)}))$

A in the outer basis.

definition $outerGram :: nat \Rightarrow nat \Rightarrow rat\ poly\ mat$ **where**

$outerGram\ K\ p = (let\ idx = outer-idx\ K\ p\ in$
 $Matrix.mat\ (length\ idx)\ (length\ idx)$
 $(\lambda(x, y). muXmod\ p\ \{1..K\}\ (Dm\ (Nof\ K)\ ^6 * outerBasis\ K\ p\ (idx\ !\ x) * outerBasis\ K\ p\ (idx\ !\ y))))$

14.10 Section 5 Normalisation (5.1), (5.2), (2.6)

definition $vpS :: nat \Rightarrow nat \Rightarrow int$ **where** $vpS\ K\ p = vp\ p\ (SK\ K)$

(5.1): $L-p(K, M)$; $\gamma-p^{\wedge}in$ is an integer (Lean *gammaIn-isInt*), the floor just keeps the definition total.

definition $Lp :: nat \Rightarrow nat \Rightarrow nat \Rightarrow int$ **where**

$Lp\ K\ M\ p =$
 $(if\ p * M \leq K\ then\ -\ 6 * int\ (hof\ K) * int\ (nat-log\ p\ (5 * K)) - int\ (hof\ K)$
 $*$
 $int\ (multiplicity\ p\ (24::nat))$
 $else\ if\ 3 * p \leq K\ then\ vpS\ K\ p + floor\ (gammaIn\ K\ M\ p)$
 $else\ if\ p \leq K\ then\ vpS\ K\ p + gammaOut\ K\ p$
 $else\ vpS\ K\ p)$

(5.2): $m-\{K, M\} = \prod -\{p \leq 2h\}\ p^{\wedge}\{-L-p(K, M)\}$.

definition $mKM :: nat \Rightarrow nat \Rightarrow rat$ **where**

$mKM\ K\ M = (\prod\ p \in \{p. p \leq 2 * hof\ K \wedge prime\ p\}. of-nat\ p\ powi\ (-\ Lp\ K\ M\ p))$

(2.6): $Q-\{K, M\} = m-\{K, M\}\ F-K$.

definition $QKM :: nat \Rightarrow nat \Rightarrow rat\ poly$ **where** $QKM\ K\ M =$

$[:mKM\ K\ M:] * FK\ K$

14.11 Section 5.1-5.3 Limiting functions (real-valued; Lean namespace *Lim*)

definition *alpha-r* :: real **where** *alpha-r* = 3 / 40

definition *lam-r* :: real **where** *lam-r* = 37 / 40

definition *Hc-r* :: real **where** *Hc-r* = 23 / 20

definition *pos-r* :: real $\Rightarrow$ real **where** *pos-r* *v* = max *v* 0

$$l(x, z) = \lfloor x - z \rfloor + \lfloor x + z \rfloor + 1.$$

definition *ell-r* :: real $\Rightarrow$ real $\Rightarrow$ real **where**

$$\text{ell-r } x \ z = \text{of-int } (\text{floor } (x - z)) + \text{of-int } (\text{floor } (x + z)) + 1$$

$$(5.4): \Gamma(x).$$

definition *Gam-r* :: real $\Rightarrow$ real **where**

$$\begin{aligned} \text{Gam-r } x = & (\text{let } T = \text{of-int } (\text{floor } (2 * \text{Hc-r} * x)); s = \text{Hc-r} * x - T / 2; \\ & q = \text{of-int } (\text{floor } (2 * x)); \text{nplus} = (2 * x - q) / 2 \text{ in} \\ & \text{interval-lebesgue-integral lborel } (\text{ereal } 0) (\text{ereal } (1 / 2)) \\ & (\lambda z. (T - 3 * \text{ell-r } (\text{alpha-r} * x) z) * (T + 3 * \text{ell-r } (\text{alpha-r} * x) z - \text{ell-r} \\ & x \ z - \\ & 5)) \\ & + s * (2 * T - q - 5) + \text{pos-r } (s - \text{nplus})) \end{aligned}$$

$$J(u) = m \ u - m(m+1)/4, \ m = \lfloor 2u \rfloor.$$

definition *J-r* :: real $\Rightarrow$ real **where**

$$\text{J-r } u = \text{of-int } (\text{floor } (2 * u)) * u - \text{of-int } (\text{floor } (2 * u)) * (\text{of-int } (\text{floor } (2 * u)) + 1) / 4$$

$$(5.5): N(x).$$

definition *Nfun-r* :: real $\Rightarrow$ real **where**

$$\begin{aligned} \text{Nfun-r } x = & 2 * \text{lam-r} * x * \text{of-int } (\text{floor } x) - 12 * \text{lam-r} * x * \text{of-int } (\text{floor} \\ & (\text{alpha-r} * x)) \\ & - 2 * \text{J-r } (\text{lam-r} * x) \end{aligned}$$

$$(5.6): R(x) = -\Gamma(x) - N(x).$$

definition *R-r* :: real $\Rightarrow$ real **where** *R-r* *x* = - *Gam-r* *x* - *Nfun-r* *x*

$$(5.8): R\text{-}0(y).$$

definition *R0-r* :: real $\Rightarrow$ real **where**

$$\begin{aligned} \text{R0-r } y = & \\ & (\text{if } 1 / 3 < y \wedge y < 1 / 2 \text{ then} \\ & \quad 8 - 9 * y - 8 * \text{alpha-r} - 5 * \min \text{alpha-r } (1 - 2 * y) - 5 * \text{pos-r } (1 + \\ & \quad \text{alpha-r} - 3 * y) \\ & \text{else if } 1 / 2 < y \wedge y < 1 \text{ then} \\ & \quad 7 * (1 - y) - 6 * \min \text{alpha-r } (1 - y) - 6 * \text{pos-r } (1 + \text{alpha-r} - 2 * y) \\ & + \text{pos-r } (1 + 4 * \\ & \quad \text{alpha-r} - 2 * y) \\ & \text{else } 0) \end{aligned}$$

$$(5.9): d(y).$$

definition $dfun-r :: real \Rightarrow real$ **where**

$dfun-r\ y = (if\ 1 / 3 < y \wedge y < 1 / 2\ then\ pos-r\ ((1 + 4 * alpha-r - 3 * y) - pos-r\ (1 + alpha-r - 3 * y))\ else\ 0)$

Integrand of (5.10).

definition $outer-integrand :: real \Rightarrow real$ **where**

$outer-integrand\ y = R0-r\ y - dfun-r\ y - 2 * lam-r * of-int\ (floor\ (1 / y)) + (\sum\ j = 1..5.\ pos-r\ (2 * lam-r - real\ j * y))$

(5.12)-(5.13): fractional parts and the pieces P , C of the tail.

definition $G0-r :: real \Rightarrow real$ **where** $G0-r\ v = v * (1 - v) * (2 * v - 1) / 6$

definition $Pper :: real \Rightarrow real$ **where**

$Pper\ x = 74 * frac\ (alpha-r * x) * (1 - frac\ (alpha-r * x)) - lam-r * frac\ x * (1 - frac\ x)$

definition $Cper :: real \Rightarrow real$ **where**

$Cper\ x = 74 / alpha-r * G0-r\ (frac\ (alpha-r * x)) - lam-r * G0-r\ (frac\ x)$

(5.10): $I-out = 127751 / 96000$.

definition $Iout :: rat$ **where** $Iout = 127751 / 96000$

(5.19).

definition $Astar :: rat$ **where** $Astar = 9928298118277006344769 / 7535670527041937280000$

(5.20).

definition $AM :: nat \Rightarrow rat$ **where**

$AM\ M = Astar + 7 * lam-q / of-nat\ M - (2923 / 240 - 1 / 4) / of-nat\ M ^ 2 + 32 / of-nat\ M ^ 3$

(6.4): the constant $Ubar$.

definition $Ubar :: rat$ **where** $Ubar = - 2733991 / 2000000$

14.12 Section 6 The real determinant

(2.10): $w(y) = (2\pi)^4 y^5 / 12 \cdot \sum_{l \geq 1} l^4 e^{-2\pi l y} (suminf; summable\ for\ y > 0)$.

definition $weight :: real \Rightarrow real$ **where**

$weight\ y = (2 * pi) ^ 4 * y ^ 5 / 12 * suminf\ (\lambda l.\ real\ (l + 1) ^ 4 * exp\ (- 2 * pi * real\ (l + 1) * y))$

(6.1): the external field V .

definition $Vf :: real \Rightarrow real$ **where**

$Vf\ t = 2 * pi * sqrt\ t + integral\ \{0..1\}\ (\lambda u.\ ln\ (t + u ^ 2)) - 6 * integral\ \{0..3 / 4\}\ (\lambda u.\ ln\ (t + u ^ 2))$

Table 1, scaled by 10^{12} : (a - j , b - j , c - j) for $j = 1, \dots, 16$.

definition *table1* :: ($\text{nat} \times \text{nat} \times \text{nat}$) list **where**

table1 =
 $[(3906748086, 8992695531, 10515596180), (2312248264, 15340997855, 29471737793),$
 $(1402286665, 25730180724, 42934365099), (881725356, 41909578246, 58204231966),$
 $(578197906, 65851089563, 69037621310), (396324613, 99481037884, 78873099189),$
 $(283911191, 144325727458, 84856120711), (212206188, 201105762729, 88396082127),$
 $(165097686, 269345996903, 88303382125), (133347132, 347089554156, 85472321255),$
 $(111522114, 430806704415, 78899184238), (96349355, 515561896511, 70353471918),$
 $(85815639, 595448778546, 58838976615), (78667711, 664241383483, 44421321106),$
 $(74129565, 716160577112, 30462865791), (71741310, 746637295669, 5959622577)]$

definition *tA* :: $\text{nat} \times \text{nat} \times \text{nat} \Rightarrow \text{real}$ **where** *tA* $e =$
 $\text{real } (\text{fst } e) / 10^{12}$

definition *tB* :: $\text{nat} \times \text{nat} \times \text{nat} \Rightarrow \text{real}$ **where** *tB* $e =$
 $\text{real } (\text{fst } (\text{snd } e)) / 10^{12}$

definition *tC* :: $\text{nat} \times \text{nat} \times \text{nat} \Rightarrow \text{real}$ **where** *tC* $e =$
 $\text{real } (\text{snd } (\text{snd } e)) / 10^{12}$

The comparison measure $\varrho = \sum c\text{-}j \ \omega\text{-}\{[a\text{-}j, b\text{-}j]\}$ of Lemma 6.1, as one density w.r.t. Lebesgue measure (Lean sums the sixteen scaled measures; the two agree).

definition *rho* :: *real measure* **where**

rho = *density lborel* ($\lambda t. \text{ennreal } (\sum e \leftarrow \text{table1}. \text{tC } e * \text{arcsine-dens } (\text{tA } e) (\text{tB } e) \ t)$)

$I(\mu) = \text{integral integral } \log|t - u| \ d\mu(t) \ d\mu(u).$

definition *logEnergy* :: *real measure* $\Rightarrow$ *real* **where**

logEnergy $\mu = \text{integral}^L \ \mu \ (\lambda t. \text{logPot } \mu \ t)$

$M\text{-}0 = -1329/200.$

definition *M0c* :: *rat* **where** $M0c = -1329 / 200$

(6.3): $C^* = -2\lambda + 12\alpha\lambda(1 - \log \alpha) + 3\lambda^2 - 2\lambda^2 \log(2\lambda).$

definition *Cstar* :: *real* **where**

$Cstar = -2 * \text{lam-r} + 12 * \text{alpha-r} * \text{lam-r} * (1 - \ln \text{alpha-r}) + 3 * \text{lam-r}^2 - 2 * \text{lam-r}^2 * \ln (2 * \text{lam-r})$

The partial masses $S\text{-}j = \sum \text{-}\{i \leq j\} \ c\text{-}i$ of Appendix A.

definition *Spart* :: $\text{nat} \Rightarrow \text{real}$ **where** *Spart* $j =$
 $(\sum e \leftarrow \text{take } j \ \text{table1}. \text{tC } e)$

end

theory *Zeta5-Functionals*

imports *Zeta5-Definitions*

begin

15 The functionals μ_X and τ_X : algebra and partial fractions

Well-definedness (a common factor of numerator and pole product cancels), linearity, the Lagrange partial-fraction decomposition, τ on polynomials and the pullback identity (3.1) of the paper. Corresponds to the Lean theories *Functionals* and parts of *Distribution*.

15.1 Linearity

lemma *muPoly-eq-sum*:

assumes $\text{degree } P \leq N$

shows $\text{muPoly } P = (\sum e \leq N. \text{Polynomial.coeff } P \ e * \text{muMono } e)$

unfolding *muPoly-def*

by (*rule sum.mono-neutral-left*) (*use assms in <auto simp: coeff-eq-0>*)

lemma *muPoly-add*: $\text{muPoly } (P + Q) = \text{muPoly } P + \text{muPoly } Q$

proof –

define N **where** $N = \max (\text{degree } P) (\text{degree } Q)$

have $\text{degree } (P + Q) \leq N$ $\text{degree } P \leq N$ $\text{degree } Q \leq N$

using *degree-add-le-max[of P Q]* **by** (*auto simp: N-def*)

thus *?thesis* **by** (*simp add: muPoly-eq-sum[of - N] sum.distrib algebra-simps*)

qed

lemma *muPoly-smult*: $\text{muPoly } (\text{Polynomial.smult } c \ P) = c * \text{muPoly } P$

using *degree-smult-le[of c P]*

by (*simp add: muPoly-eq-sum[of - degree P] sum-distrib-left mult.assoc*)

lemma *Lfun-eq-sum*:

assumes $\text{degree } P \leq N$

shows $\text{Lfun } P = (\sum k \leq N. \text{Polynomial.coeff } P \ k * \text{bernoulli } k)$

unfolding *Lfun-def*

by (*rule sum.mono-neutral-left*) (*use assms in <auto simp: coeff-eq-0>*)

lemma *Lfun-add*: $\text{Lfun } (P + Q) = \text{Lfun } P + \text{Lfun } Q$

proof –

define N **where** $N = \max (\text{degree } P) (\text{degree } Q)$

have $\text{degree } (P + Q) \leq N$ $\text{degree } P \leq N$ $\text{degree } Q \leq N$

using *degree-add-le-max[of P Q]* **by** (*auto simp: N-def*)

thus *?thesis* **by** (*simp add: Lfun-eq-sum[of - N] sum.distrib algebra-simps*)

qed

lemma *Lfun-smult*: $\text{Lfun } (\text{Polynomial.smult } c \ P) = c * \text{Lfun } P$

using *degree-smult-le[of c P]*

by (*simp add: Lfun-eq-sum[of - degree P] sum-distrib-left mult.assoc*)

lemma *tau-add*: $\tau (P + Q) = \tau P + \tau Q$

by (*simp add: tau-def numeral-3-eq-3 pderiv-add Lfun-add add-divide-distrib*)

lemma *tau-smult*: $\text{tau } (\text{Polynomial.smult } c \ P) = c * \text{tau } P$
by (*simp add: tau-def numeral-3-eq-3 pderiv-smult Lfun-smult*)

lemma *sqResidue-add*: $\text{sqResidue } S \ (P + Q) \ j = \text{sqResidue } S \ P \ j + \text{sqResidue } S \ Q \ j$
by (*simp add: sqResidue-def add-divide-distrib*)

lemma *sqResidue-smult*: $\text{sqResidue } S \ (\text{Polynomial.smult } c \ P) \ j = c * \text{sqResidue } S \ P \ j$
by (*simp add: sqResidue-def*)

lemma *linResidue-add*: $\text{linResidue } S \ (P + Q) \ r = \text{linResidue } S \ P \ r + \text{linResidue } S \ Q \ r$
by (*simp add: linResidue-def add-divide-distrib*)

lemma *linResidue-smult*: $\text{linResidue } S \ (\text{Polynomial.smult } c \ P) \ r = c * \text{linResidue } S \ P \ r$
by (*simp add: linResidue-def*)

lemma *muX-add*: $\text{muX } S \ (P + Q) = \text{muX } S \ P + \text{muX } S \ Q$
unfolding *muX-def*
by (*simp add: poly-div-add-left muPoly-add sqResidue-add smult-add-left sum.distrib algebra-simps*)

lemma *smult-sum-right*: $\text{Polynomial.smult } c \ (\sum i \in S. f \ i) = (\sum i \in S. \text{Polynomial.smult } c \ (f \ i))$
by (*induction S rule: infinite-finite-induct*) (*auto simp: smult-add-right*)

lemma *muX-C-mul*: $\text{muX } S \ ([:c:] * P) = [:c:] * \text{muX } S \ P$
unfolding *muX-def*
by (*auto simp: div-smult-left muPoly-smult sqResidue-smult smult-add-right smult-smult smult-sum-right*)

lemma *tauX-add*: $\text{tauX } S \ (P + Q) = \text{tauX } S \ P + \text{tauX } S \ Q$
unfolding *tauX-def*
by (*simp add: poly-div-add-left tau-add linResidue-add smult-add-left sum.distrib algebra-simps*)

lemma *tauX-C-mul*: $\text{tauX } S \ ([:c:] * P) = [:c:] * \text{tauX } S \ P$
unfolding *tauX-def*
by (*simp add: div-smult-left tau-smult linResidue-smult smult-add-right smult-smult smult-sum-right*)

15.2 Well-definedness: common factors cancel

muX depends only on the rational function (Lean: *muX-mul-cancel*).

lemma *muX-mul-cancel*:
 $\text{finite } S \implies j \notin S \implies$

$\text{muX } (\text{insert } j \text{ } S) (P * (pX + [:of\text{-}nat \ j \ ^2:])) = \text{muX } S \ P$
proof –
 assume S : *finite* S and j : $j \notin S$
 define c where $c = pX + [:of\text{-}nat \ j \ ^2:]$
 have $c0$: $c \neq 0$
proof
 assume $c = 0$
 hence $\text{degree } c = 0$ by *simp*
 moreover have $\text{degree } c = 1$ unfolding $c\text{-def}$ by (*simp add: monom-Suc*)
 ultimately show *False* by *simp*
qed
 have pc : $\text{poly } c \ x = x + of\text{-}nat \ j \ ^2$ for x by (*simp add: c-def poly-monom*)
 have SP : $\text{sqPoleProd } (\text{insert } j \ S) = c * \text{sqPoleProd } S$
 unfolding $\text{sqPoleProd-def } c\text{-def}$ using $S \ j$ by *simp*
 have quot : $(P * c) \text{ div } \text{sqPoleProd } (\text{insert } j \ S) = P \text{ div } \text{sqPoleProd } S$
 unfolding SP by (*subst mult.commute*) (*rule div-mult-mult1 [OF c0]*)
 have resj : $\text{sqResidue } (\text{insert } j \ S) (P * c) \ j = 0$
 by (*simp add: sqResidue-def pc*)
 have resk : $\text{sqResidue } (\text{insert } j \ S) (P * c) \ k = \text{sqResidue } S \ P \ k$ if k : $k \in S$ for k
proof –
 have jk : $j \neq k$ using $j \ k$ by *auto*
 have $d0$: $(of\text{-}nat \ j \ ^2 - of\text{-}nat \ k \ ^2 :: \text{rat}) \neq 0$ using jk by *simp*
 have ins : $\text{insert } j \ S - \{k\} = \text{insert } j \ (S - \{k\})$ using jk by *auto*
 have $\text{sqResidue } (\text{insert } j \ S) (P * c) \ k$
 = $\text{poly } P \ (- (of\text{-}nat \ k \ ^2)) * (of\text{-}nat \ j \ ^2 - of\text{-}nat \ k \ ^2)$
 / $((of\text{-}nat \ j \ ^2 - of\text{-}nat \ k \ ^2) * (\prod_{l \in S - \{k\}. of\text{-}nat \ l \ ^2 - of\text{-}nat \ k \ ^2}))$
 unfolding sqResidue-def ins using $S \ j$ by (*simp add: pc*)
 also have $\dots = \text{sqResidue } S \ P \ k$
 unfolding sqResidue-def by (*rule nonzero-mult-divide-mult-cancel-right2 [OF d0]*)
 finally show *?thesis* .
qed
 have $\text{muX } (\text{insert } j \ S) (P * c)$
 = $[:\text{muPoly } (P \text{ div } \text{sqPoleProd } S):] + (\sum_{k \in S. [: \text{sqResidue } (\text{insert } j \ S) (P * c) \ k:]} * \text{muPole } k)$
 unfolding muX-def quot using $S \ j$ by (*simp add: resj*)
 also have $(\sum_{k \in S. [: \text{sqResidue } (\text{insert } j \ S) (P * c) \ k:]} * \text{muPole } k) = (\sum_{k \in S. [: \text{sqResidue } S \ P \ k:]} * \text{muPole } k)$
 by (*rule sum.cong*) (*simp-all add: resk*)
 finally show *?thesis* unfolding $c\text{-def muX-def}$.
qed

lemma *muX-mul-cancel-set*:

$\text{finite } S \implies \text{finite } T \implies T \cap S = \{\} \implies$
 $\text{muX } (T \cup S) (P * \text{sqPoleProd } T) = \text{muX } S \ P$

proof –

assume S : *finite* S and T : *finite* T

show $T \cap S = \{\} \implies \text{muX } (T \cup S) (P * \text{sqPoleProd } T) = \text{muX } S \ P$

```

    using T
  proof (induction T rule: finite-induct)
    case empty
    thus ?case by (simp add: sqPoleProd-def)
  next
    case (insert j T)
    have jTS:  $j \notin T \cup S$  using insert by auto
    have e:  $P * sqPoleProd (insert j T) = (P * sqPoleProd T) * (pX + [:of-nat j \wedge 2:])$ 
    unfolding sqPoleProd-def using insert.hyps by (simp add: ac-simps)
    have muX (insert j T  $\cup$  S) (P * sqPoleProd (insert j T))
      = muX (insert j (T  $\cup$  S)) ((P * sqPoleProd T) * (pX + [:of-nat j  $\wedge$  2:]))
    unfolding e by simp
    also have ... = muX (T  $\cup$  S) (P * sqPoleProd T)
      by (rule muX-mul-cancel) (use S insert.hyps(1) jTS in auto)
    also have ... = muX S P using insert by auto
    finally show ?case .
  qed
qed

```

15.3 Partial fractions (Lagrange decomposition)

$P = \Pi (P \text{ div } \Pi) + \sum -s \text{ res-}s \prod -\{s' \neq s\} (X - s')$ (Lean: *pt-lagrange*).

```

lemma pX-minus-const:  $pX - [:s:] = [- s, 1:]$ 
  by (simp add: monom-Suc)

```

```

lemma qPoleProd-lead-coeff:  $\text{finite } S \implies \text{lead-coeff } (qPoleProd S) = 1$ 
  unfolding qPoleProd-def pX-minus-const
proof (induction S rule: finite-induct)
  case (insert x F)
  show ?case
    unfolding prod.insert[OF insert.hyps] lead-coeff-mult insert.IH by simp
qed simp

```

Monicity is written as $\text{lead-coeff} = 1$ (*monic* is not a constant in the base image).

```

lemma qPoleProd-monic:  $\text{finite } S \implies \text{lead-coeff } (qPoleProd S) = 1$ 
  by (rule qPoleProd-lead-coeff)

```

```

lemma degree-qPoleProd:  $\text{finite } S \implies \text{degree } (qPoleProd S) = \text{card } S$ 
  unfolding qPoleProd-def pX-minus-const
proof (induction S rule: finite-induct)
  case (insert x F)
  have ( $\prod_{s \in F} [- s, 1:]$ )  $\neq 0$ 
    using insert.hyps(1) by (simp add: prod-zero-iff)
  thus ?case
    unfolding prod.insert[OF insert.hyps] using insert
    by (subst degree-mult-eq) auto

```


qed simp

lemma poly-qPoleProd: poly (qPoleProd T) x = ($\prod_{t \in T} x - t$)
 unfolding qPoleProd-def by (simp add: poly-prod poly-monom)

lemma poly-eq-0-of-roots:

fixes Q :: rat poly

assumes finite S degree Q < card S $\bigwedge s. s \in S \implies \text{poly } Q s = 0$

shows Q = 0

proof (rule ccontr)

assume Q0: Q $\neq$ 0

have S $\subseteq \{x. \text{poly } Q x = 0\}$ using assms(3) by auto

hence card S $\leq$ card $\{x. \text{poly } Q x = 0\}$

by (intro card-mono) (use Q0 poly-roots-finite in auto)

also have ... $\leq$ degree Q by (rule poly-roots-degree[OF Q0])

finally show False using assms(2) by simp

qed

lemma lagrange-decomp-aux:

assumes S: finite S

shows P = qPoleProd S * (P div qPoleProd S) + ($\sum_{s \in S} [:qRes S P s:] * qPoleProd (S - \{s\})$)

proof -

define Pi where Pi = qPoleProd S

define R where R = P mod Pi

define L where L = ($\sum_{s \in S} [:qRes S P s:] * qPoleProd (S - \{s\})$)

have Pi0: Pi $\neq$ 0 using qPoleProd-lead-coeff[OF S] by (auto simp: Pi-def)

have dPi: degree Pi = card S unfolding Pi-def by (rule degree-qPoleProd[OF S])

have PR: P = Pi * (P div Pi) + R unfolding R-def by simp

have R = L

proof (cases S = {})

case True

hence Pi = 1 by (simp add: Pi-def qPoleProd-def)

thus ?thesis using True by (simp add: R-def L-def)

next

case Sne: False

have cS: 0 < card S using S Sne by (simp add: card-gt-0-iff)

have dR: degree R < card S

proof (cases R = 0)

case True thus ?thesis using cS by simp

next

case False thus ?thesis using degree-mod-less'[OF Pi0] dPi by (simp add: R-def)

qed

have dL: degree L < card S

proof -

have degree L $\leq$ card S - 1

unfolding L-def

```

proof (rule degree-sum-le[OF S])
  fix  $s$  assume  $s: s \in S$ 
  have  $\text{degree } ([:qRes\ S\ P\ s:] * qPoleProd\ (S - \{s\})) \leq \text{degree } (qPoleProd\ (S - \{s\}))$ 
  by (simp add: degree-smult-le)
  also have  $\dots = \text{card } S - 1$  using  $S\ s$  by (simp add: degree-qPoleProd)
  finally show  $\text{degree } ([:qRes\ S\ P\ s:] * qPoleProd\ (S - \{s\})) \leq \text{card } S - 1$  .
qed
thus ?thesis using  $cS$  by simp
qed
have roots:  $\text{poly } (R - L)\ s = 0$  if  $s: s \in S$  for  $s$ 
proof -
  have  $\text{poly } Pi\ s = 0$  unfolding  $Pi\text{-def } \text{poly-qPoleProd}$  using  $S\ s$  by (auto simp: prod-zero-iff)
  hence  $pR: \text{poly } R\ s = \text{poly } P\ s$  using arg-cong[OF PR, of  $\lambda q. \text{poly } q\ s$ ] by simp
  have  $nz: (\prod t \in S - \{s\}. s - t) \neq 0$  using  $S$  by (simp add: prod-zero-iff)
  have  $\text{poly } L\ s = (\sum s' \in S. qRes\ S\ P\ s' * (\prod t \in S - \{s'\}. s - t))$ 
  unfolding  $L\text{-def}$  by (simp add: poly-sum poly-qPoleProd)
  also have  $\dots = qRes\ S\ P\ s * (\prod t \in S - \{s\}. s - t)$ 
   $+ (\sum s' \in S - \{s\}. qRes\ S\ P\ s' * (\prod t \in S - \{s'\}. s - t))$ 
  using  $S\ s$  by (simp add: sum.remove)
  also have  $(\sum s' \in S - \{s\}. qRes\ S\ P\ s' * (\prod t \in S - \{s'\}. s - t)) = 0$ 
  using  $S\ s$  by (intro sum.neutral) (auto simp: prod-zero-iff)
  also have  $qRes\ S\ P\ s * (\prod t \in S - \{s\}. s - t) = \text{poly } P\ s$ 
  using  $nz$  by (simp add: qRes-def)
  finally show ?thesis using  $pR$  by simp
qed
have  $R - L = 0$ 
proof (rule poly-eq-0-of-roots[OF S - roots])
  show  $\text{degree } (R - L) < \text{card } S$ 
  using degree-diff-le-max[of R L]  $dR\ dL$  by simp
qed
thus ?thesis by simp
qed
thus ?thesis using PR by (simp add: Pi-def L-def)
qed

lemma lagrange-decomp:
   $\text{finite } S \implies P = qPoleProd\ S * (P \text{ div } qPoleProd\ S) + (\sum s \in S. [:qRes\ S\ P\ s:] * qPoleProd\ (S - \{s\}))$ 
by (rule lagrange-decomp-aux)

```

15.4 (3.1) The pullback identity

The pullback identity (3.1) is stated and proved at the end of this theory.

15.5 τ on polynomials

$\tau(z^d) = \kappa \cdot d$, and $\tau K = \tau$.

lemma *pX-pow-monom*: $pX^d = \text{Polynomial.monom } 1 \ d$
by (*simp add: monom-power*)

lemma *Lfun-monom*: $Lfun (\text{Polynomial.monom } c \ k) = c * \text{bernoulli } k$
proof –

have $Lfun (\text{Polynomial.monom } c \ k) = (\sum j \leq k. \text{Polynomial.coeff } (\text{Polynomial.monom } c \ k) \ j * \text{bernoulli } j)$
by (*rule Lfun-eq-sum*) (*simp add: degree-monom-le*)
also have $\dots = (\sum j \leq k. \text{if } j = k \text{ then } c * \text{bernoulli } k \text{ else } 0)$
by (*rule sum.cong*) (*auto simp: coeff-monom*)
finally show *?thesis* **by** *simp*
qed

lemma *tau-X-pow-eq-kappa*: $\tau (pX^d) = \kappa \ d$

proof –

have $(pderiv \ \sim 3) (pX^d) = \text{Polynomial.monom } (of\text{-nat } (d-2) * (of\text{-nat } (d-1) * of\text{-nat } d)) \ (d-3)$
unfolding *pX-pow-monom* **by** (*simp add: numeral-3-eq-3 numeral-2-eq-2 pderiv-monom*)
hence $\tau (pX^d) = of\text{-nat } (d-2) * (of\text{-nat } (d-1) * of\text{-nat } d) * \text{bernoulli } (d-3) / 24$
by (*simp add: tau-def Lfun-monom*)
also have $\dots = \kappa \ d$
proof –
consider $d = 0 \mid d = 1 \mid d = 2 \mid 3 \leq d$ **by** *arith*
thus *?thesis*
by *cases* (*auto simp: kappa-def of-nat-diff*)
qed
finally show *?thesis* .

qed

lemma *tau-zero*: $\tau 0 = 0$

using *tau-smult[of 0 0]* **by** *simp*

lemma *tau-sum*: $\tau (\sum i \in S. f \ i) = (\sum i \in S. \tau (f \ i))$

by (*induction S rule: infinite-finite-induct*) (*auto simp: tau-add tau-zero*)

lemma *tau-monom*: $\tau (\text{Polynomial.monom } c \ d) = c * \kappa \ d$

proof –

have $\text{Polynomial.monom } c \ d = \text{Polynomial.smult } c \ (pX^d)$
unfolding *pX-pow-monom* **by** (*simp add: smult-monom*)
thus *?thesis* **by** (*simp only: tau-smult tau-X-pow-eq-kappa*)
qed

lemma *tauK-eq-tau*: $\tau K \ Q = \tau \ Q$

proof –

have $\tau \ Q = \tau (\sum d \leq \text{degree } Q. \text{Polynomial.monom } (\text{Polynomial.coeff } Q \ d) \ d)$

by (simp only: poly-as-sum-of-monoms)
 also have ... = tauK Q
 by (simp add: tau-sum tau-monom tauK-def)
 finally show ?thesis by simp
 qed

lemma sum-binomial-bernoulli-lt:
 $(\sum k < n. \text{of_nat } (n \text{ choose } k) * \text{bernoulli } k) = (\text{if } n = 1 \text{ then } 1 \text{ else } (0 :: \text{rat}))$
proof –
 have real-of-rat $(\sum k < n. \text{of_nat } (n \text{ choose } k) * \text{bernoulli } k) = \text{real-of-rat } (\text{if } n = 1 \text{ then } 1 \text{ else } 0)$
 by (cases n = 1)
 (simp-all add: of-rat-sum of-rat-mult of-rat-bernoulli sum-binomial-times-bernoulli')
 thus ?thesis by (simp only: of-rat-eq-iff)
 qed

lemma kappa-0: kappa 0 = 0 and kappa-1: kappa 1 = 0 and kappa-2: kappa 2 = 0
 and kappa-Suc-0: kappa (Suc 0) = 0 and kappa-Suc-Suc-0: kappa (Suc (Suc 0)) = 0
 by (simp-all add: kappa-def)

lemma kappa-Suc3:
 $\text{kappa } (\text{Suc } (\text{Suc } (\text{Suc } j))) = \text{of_nat } (\text{Suc } (\text{Suc } (\text{Suc } j)) * \text{Suc } (\text{Suc } j) * \text{Suc } j) * \text{bernoulli } j / 24$
 by (simp add: kappa-def algebra-simps)

lemma choose-falling3:
 $(\text{Suc } (\text{Suc } (\text{Suc } m)) \text{ choose } \text{Suc } (\text{Suc } (\text{Suc } j))) * (\text{Suc } (\text{Suc } (\text{Suc } j)) * \text{Suc } (\text{Suc } j) * \text{Suc } j)$
 $= \text{Suc } (\text{Suc } (\text{Suc } m)) * \text{Suc } (\text{Suc } m) * \text{Suc } m * (m \text{ choose } j)$
proof –
 have $(\text{Suc } (\text{Suc } (\text{Suc } m)) \text{ choose } \text{Suc } (\text{Suc } (\text{Suc } j))) * (\text{Suc } (\text{Suc } (\text{Suc } j)) * \text{Suc } (\text{Suc } j) * \text{Suc } j)$
 $= (\text{Suc } (\text{Suc } (\text{Suc } j)) * (\text{Suc } (\text{Suc } (\text{Suc } m)) \text{ choose } \text{Suc } (\text{Suc } (\text{Suc } j)))) * \text{Suc } (\text{Suc } j) * \text{Suc } j$
 by (simp only: ac-simps)
 also have ... = $\text{Suc } (\text{Suc } (\text{Suc } m)) * ((\text{Suc } (\text{Suc } j)) * (\text{Suc } (\text{Suc } m) \text{ choose } \text{Suc } (\text{Suc } j))) * \text{Suc } j$
 by (subst Suc-times-binomial) (simp only: ac-simps)
 also have ... = $\text{Suc } (\text{Suc } (\text{Suc } m)) * \text{Suc } (\text{Suc } m) * (\text{Suc } j * (\text{Suc } m \text{ choose } \text{Suc } j))$
 by (subst Suc-times-binomial) (simp only: ac-simps)
 also have ... = $\text{Suc } (\text{Suc } (\text{Suc } m)) * \text{Suc } (\text{Suc } m) * \text{Suc } m * (m \text{ choose } j)$
 by (subst Suc-times-binomial) (simp only: ac-simps)
 finally show ?thesis .
 qed

```

lemma sum-choose-kappa-aux:
  ( $\sum d \leq n. \text{of-nat } (n \text{ choose } d) * \text{kappa } d$ ) = (if  $n = 4$  then 1 else 0) +  $\text{kappa } n$ 
proof (cases  $n < 3$ )
  case True
    have  $k0: \text{kappa } d = 0$  if  $d \leq n$  for  $d$ 
    using that True kappa-0 kappa-1 kappa-2 by (cases  $d = 0 \vee d = 1 \vee d = 2$ )
  auto
    have ( $\sum d \leq n. \text{of-nat } (n \text{ choose } d) * \text{kappa } d$ ) = 0
    by (rule sum.neutral) (simp add:  $k0$ )
    thus ?thesis using True k0[of n] by simp
  next
    case False
    define  $m$  where  $m = n - 3$ 
    have  $n: n = \text{Suc } (\text{Suc } (\text{Suc } m))$  using False unfolding m-def by arith
    define  $c :: \text{rat}$  where  $c = \text{of-nat } (\text{Suc } (\text{Suc } (\text{Suc } m)) * \text{Suc } (\text{Suc } m) * \text{Suc } m)$ 
    / 24
    have  $tm: \text{of-nat } (\text{Suc } (\text{Suc } (\text{Suc } m)) \text{ choose } \text{Suc } (\text{Suc } (\text{Suc } j))) * \text{kappa } (\text{Suc } (\text{Suc } (\text{Suc } j)))$ 
    =  $c * (\text{of-nat } (m \text{ choose } j) * \text{bernoulli } j)$  for  $j$ 
    proof -
      have  $\text{of-nat } (\text{Suc } (\text{Suc } (\text{Suc } m)) \text{ choose } \text{Suc } (\text{Suc } (\text{Suc } j))) * \text{kappa } (\text{Suc } (\text{Suc } (\text{Suc } j)))$ 
      =  $\text{of-nat } ((\text{Suc } (\text{Suc } (\text{Suc } m)) \text{ choose } \text{Suc } (\text{Suc } (\text{Suc } j))) * (\text{Suc } (\text{Suc } (\text{Suc } j)) * \text{Suc } (\text{Suc } j) * \text{bernoulli } j / 24$ 
      unfolding kappa-Suc3 by (simp only: of-nat-mult)
      also have  $\dots = \text{of-nat } (\text{Suc } (\text{Suc } (\text{Suc } m)) * \text{Suc } (\text{Suc } m) * \text{Suc } m * (m \text{ choose } j)) * \text{bernoulli } j / 24$ 
      by (simp only: choose-falling3)
      also have  $\dots = c * (\text{of-nat } (m \text{ choose } j) * \text{bernoulli } j)$ 
      unfolding c-def by (simp only: of-nat-mult) (simp add: algebra-simps)
      finally show ?thesis .
    qed
    have ( $\sum d \leq n. \text{of-nat } (n \text{ choose } d) * \text{kappa } d$ )
    = ( $\sum j \leq m. \text{of-nat } (\text{Suc } (\text{Suc } (\text{Suc } m)) \text{ choose } \text{Suc } (\text{Suc } (\text{Suc } j))) * \text{kappa } (\text{Suc } (\text{Suc } (\text{Suc } j)))$ )
    unfolding  $n$  by (simp only: sum.atMost-Suc-shift) (simp add: kappa-0 kappa-Suc-0 kappa-Suc-Suc-0)
    also have  $\dots = c * (\sum j \leq m. \text{of-nat } (m \text{ choose } j) * \text{bernoulli } j)$ 
    by (simp only: tm sum-distrib-left)
    also have ( $\sum j \leq m. \text{of-nat } (m \text{ choose } j) * \text{bernoulli } j$ ) = (if  $m = 1$  then 1 else 0) + ( $\text{bernoulli } m :: \text{rat}$ )
    by (simp only: lessThan-Suc-atMost[symmetric] sum.lessThan-Suc sum-binomial-bernoulli-lt)
  simp
    also have  $c * ((\text{if } m = 1 \text{ then } 1 \text{ else } 0) + \text{bernoulli } m)$  = (if  $n = 4$  then 1 else 0) +  $\text{kappa } n$ 

```

unfolding n kappa-Suc3 $c\text{-def}$ **by** ($\text{cases } m = 1$) simp-all
finally show $?thesis$.
qed

lemma tau-uminus : $\text{tau } (- P) = - \text{tau } P$
using $\text{tau-smult}[of \ -1 \ P]$ **by** simp

lemma tau-diff : $\text{tau } (P - Q) = \text{tau } P - \text{tau } Q$
using $\text{tau-add}[of \ P - Q]$ **by** ($\text{simp add: tau-uminus}$)

lemma tau-of-nat-mult : $\text{tau } (\text{of-nat } c * P) = \text{of-nat } c * \text{tau } P$
proof -
have $\text{of-nat } c * P = \text{Polynomial.smult } (\text{of-nat } c) \ P$ **by** ($\text{simp add: of-nat-poly}$)
thus $?thesis$ **by** ($\text{simp only: tau-smult}$)
qed

lemma tau-shift-pow : $\text{tau } ((pX + 1) ^ n) = (\text{if } n = 4 \text{ then } 1 \text{ else } 0) + \text{kappa } n$
proof -
have $\text{eq: } (pX + 1) ^ n = (\sum k \leq n. \text{of-nat } (n \text{ choose } k) * pX ^ k)$
using $\text{binomial-ring}[of \ pX \ 1 \ n]$ **by** simp
have $\text{tau } ((pX + 1) ^ n) = (\sum k \leq n. \text{of-nat } (n \text{ choose } k) * \text{kappa } k)$
unfolding eq **by** ($\text{simp only: tau-sum tau-of-nat-mult tau-X-pow-eq-kappa}$)
thus $?thesis$ **by** ($\text{simp only: sum-choose-kappa-aux}$)
qed

lemma tau-fwd-monom :
 $\text{tau } (\text{pcompose } (\text{Polynomial.monom } c \ n) \ (pX + 1) - \text{Polynomial.monom } c \ n) =$
 $(\text{if } n = 4 \text{ then } c \text{ else } 0)$
proof -
have m : $\text{Polynomial.monom } c \ n = \text{Polynomial.smult } c \ (pX ^ n)$
unfolding $pX\text{-pow-monom}$ **by** ($\text{simp add: smult-monom}$)
have $\text{tau } (\text{pcompose } (\text{Polynomial.monom } c \ n) \ (pX + 1) - \text{Polynomial.monom } c \ n)$
 $= c * ((\text{if } n = 4 \text{ then } 1 \text{ else } 0) + \text{kappa } n) - c * \text{kappa } n$
unfolding m $\text{pcompose-smult pcompose-power-left pcompose-pX tau-diff tau-smult}$
 tau-shift-pow
 $\text{tau-X-pow-eq-kappa ..}$
thus $?thesis$ **by** ($\text{simp add: algebra-simps}$)
qed

lemma tau-fwd-aux : $\text{tau } (\text{pcompose } f \ (pX + 1) - f) = \text{Polynomial.coeff } f \ 4$
proof -
define D **where** $D = \text{degree } f$
define M **where** $M \ n = \text{Polynomial.monom } (\text{Polynomial.coeff } f \ n) \ n$ **for** n
have e : $(\sum n \leq D. M \ n) = f$ **unfolding** $D\text{-def } M\text{-def}$ **by** ($\text{rule poly-as-sum-of-monoms}$)
have $\text{tau } (\text{pcompose } f \ (pX + 1) - f) = \text{tau } (\text{pcompose } (\sum n \leq D. M \ n) \ (pX + 1) - (\sum n \leq D. M \ n))$
by (simp only: e)
also have $\dots = \text{tau } (\sum n \leq D. (\text{pcompose } (M \ n) \ (pX + 1) - M \ n))$

by (simp only: pcompose-sum sum-subtractf)
 also have ... = $(\sum n \leq D. \text{if } n = 4 \text{ then } \text{Polynomial.coeff } f \ n \text{ else } 0)$
 by (simp only: tau-sum M-def tau-fwd-monom)
 also have ... = $\text{Polynomial.coeff } f \ 4$
 by (cases $4 \leq D$) (auto simp: D-def coeff-eq-0)
 finally show ?thesis .
 qed

(3.3) for polynomials: $\tau(f(x+1) - f(x)) = f''''(0)/24 = [x^4] f$ (Lean: tau-fwd).

lemma tau-fwd: $\text{tau } (pcompose \ f \ (pX + 1) - f) = \text{Polynomial.coeff } f \ 4$
 by (rule tau-fwd-aux)

$\sum \{d \leq n\} C(n, d) \kappa \cdot d = [n = 4] + \kappa \cdot n.$

lemma sum-choose-kappa:
 $(\sum d \leq n. \text{of-nat } (n \text{ choose } d) * \kappa \ d) = (\text{if } n = 4 \text{ then } 1 \text{ else } 0) + \kappa \ n$
 by (rule sum-choose-kappa-aux)

Every polynomial is a forward difference.

lemma pX-shift-pow-diff:
 $pcompose \ (pX \wedge m) \ (pX + 1) - pX \wedge m = (\sum k < m. \text{Polynomial.monom } (\text{of-nat } (m \text{ choose } k)) \ k)$

proof -
 have $(pX + 1) \wedge m = (\sum k \leq m. \text{of-nat } (m \text{ choose } k) * pX \wedge k)$
 using binomial-ring[of pX 1 m] by simp
 also have ... = $(\sum k \leq m. \text{Polynomial.monom } (\text{of-nat } (m \text{ choose } k)) \ k)$
 by (rule sum.cong[OF refl], unfold pX-pow-monom, simp add: of-nat-poly smult-monom)
 also have ... = $(\sum k < m. \text{Polynomial.monom } (\text{of-nat } (m \text{ choose } k)) \ k) + pX \wedge m$
 unfolding pX-pow-monom by (simp add: lessThan-Suc-atMost[symmetric])
 finally show ?thesis unfolding pcompose-power-left pcompose-pX by simp
 qed

lemma exists-antidiff-aux: $\exists g. pcompose \ g \ (pX + 1) - g = f$

proof (induction f rule: measure-induct-rule[where f = degree])

case (less f)

show ?case

proof (cases f = 0)

case True

thus ?thesis by (intro exI[of - 0]) simp

next

case False

define n where n = degree f

define c where c = lead-coeff f

define a where a = c / of-nat (Suc n)

define E where $E = (\sum k < \text{Suc } n. \text{Polynomial.monom } (\text{of-nat } (\text{Suc } n \text{ choose } k)) \ k :: \text{rat poly})$

```

define g0 where g0 = Polynomial.smult a (pX ^ Suc n)
have Dg0: pcompose g0 (pX + 1) - g0 = Polynomial.smult a E
unfolding g0-def pcompose-smult smult-diff-right[symmetric] pX-shift-pow-diff
E-def ..
have dE: degree E ≤ n
unfolding E-def by (rule degree-sum-le) (auto intro: order-trans[OF de-
gree-monom-le])
have cE: Polynomial.coeff E n = of-nat (Suc n)
unfolding E-def by (simp add: coeff-sum coeff-monom)
define f' where f' = f - Polynomial.smult a E
have df': degree f' ≤ n
unfolding f'-def n-def using dE[unfolded n-def]
by (intro degree-diff-le) (auto intro: order-trans[OF degree-smult-le])
have cf': Polynomial.coeff f' n = 0
unfolding f'-def using cE by (simp add: a-def c-def n-def)
obtain g' where g': pcompose g' (pX + 1) - g' = f'
proof (cases f' = 0)
  case True
    thus ?thesis using that[of 0] by simp
  next
    case False
      hence degree f' ≠ n using cf' by (metis leading-coeff-0-iff)
      hence degree f' < degree f using df' by (simp add: n-def)
      thus ?thesis using less.IH that by blast
    qed
  have pcompose (g0 + g') (pX + 1) - (g0 + g')
    = (pcompose g0 (pX + 1) - g0) + (pcompose g' (pX + 1) - g')
    by (simp add: pcompose-add algebra-simps)
  also have ... = f unfolding Dg0 g' f'-def by simp
  finally show ?thesis by blast
qed
qed

```

```

lemma exists-antidiff: ∃ g. pcompose g (pX + 1) - g = f
by (rule exists-antidiff-aux)

```

15.6 (3.1) The pullback identity: proof

```

definition pullbackR :: nat set ⇒ rat poly ⇒ rat poly where
  pullbackR S P = tauX (signedPoles S) ([:(-1) ^ card S:] * pX ^ 5 * pcompose
P (- (pX ^ 2)))

```

```

lemma muX-zero: muX S 0 = 0
using muX-C-mul[of S 0 0] by simp

```

```

lemma muX-sum: muX S (∑ i ∈ I. f i) = (∑ i ∈ I. muX S (f i))
by (induction I rule: infinite-finite-induct) (simp-all add: muX-add muX-zero)

```

```

lemma tauX-zero: tauX T 0 = 0

```


using *tauX-C-mul*[*of T 0 0*] **by** *simp*

lemma *pullbackR-add*: $\text{pullbackR } S (P + Q) = \text{pullbackR } S P + \text{pullbackR } S Q$
unfolding *pullbackR-def pcompose-add distrib-left tauX-add ..*

lemma *pullbackR-C-mul*: $\text{pullbackR } S ([:a:] * P) = [:a:] * \text{pullbackR } S P$
proof –
have $[:(-1) \wedge \text{card } S:] * pX \wedge 5 * \text{pcompose } ([:a:] * P) (- (pX \wedge 2))$
 $= [:a:] * ([:(-1) \wedge \text{card } S:] * pX \wedge 5 * \text{pcompose } P (- (pX \wedge 2)))$
unfolding *pcompose-mult pcompose-const* **by** (*simp only: ac-simps*)
thus *?thesis* **unfolding** *pullbackR-def* **by** (*simp only: tauX-C-mul*)
qed

lemma *pullbackR-zero*: $\text{pullbackR } S 0 = 0$
by (*simp add: pullbackR-def tauX-zero*)

lemma *pullbackR-sum*: $\text{pullbackR } S (\sum i \in I. f i) = (\sum i \in I. \text{pullbackR } S (f i))$
by (*induction I rule: infinite-finite-induct*) (*simp-all add: pullbackR-add pullbackR-zero*)

lemma *pcompose-sq-factor*:
 $\text{pcompose } (pX + [: \text{of-nat } j \wedge 2:]) (- (pX \wedge 2))$
 $= [:- 1:] * ((pX - [: \text{of-int } (\text{int } j):]) * (pX - [: \text{of-int } (- \text{int } j):]))$
proof –
have $\text{pcompose } (pX + [: \text{of-nat } j \wedge 2:]) (- (pX \wedge 2)) = - (pX \wedge 2) + [: \text{of-nat } j \wedge 2:]$
unfolding *pcompose-add pcompose-pX pcompose-const ..*
also have $\dots = [:- 1:] * ((pX - [: \text{of-nat } j:]) * (pX + [: \text{of-nat } j:]))$
by (*simp add: algebra-simps power2-eq-square*)
finally show *?thesis* **by** *simp*
qed

lemma *const-poly-power*: $[:a:] \wedge n = [:a \wedge n:]$
by (*induction n*) (*simp-all add: mult.commute*)

lemma *minus-one-poly*: $(- 1 :: \text{rat poly}) = [:- 1:]$
by (*simp add: one-pCons*)

lemma *pcompose-sqPoleProd*:
assumes *S: finite S and S0: 0 ∉ S*
shows $\text{pcompose } (\text{sqPoleProd } S) (- (pX \wedge 2)) = [:(-1) \wedge \text{card } S:] * \text{linPoleProd } (\text{signedPoles } S)$
proof –
define *A* **where** $A j = pX - [: \text{of-int } (\text{int } j) :: \text{rat}:]$ **for** $j :: \text{nat}$
define *B* **where** $B j = pX - [: \text{of-int } (- \text{int } j) :: \text{rat}:]$ **for** $j :: \text{nat}$
have $\text{pcompose } (\text{sqPoleProd } S) (- (pX \wedge 2)) = (\prod j \in S. [:- 1:] * (A j * B j))$
unfolding *sqPoleProd-def pcompose-prod pcompose-sq-factor A-def B-def ..*
also have $\dots = [:- 1:] \wedge \text{card } S * ((\prod j \in S. A j) * (\prod j \in S. B j))$

```

    by (simp only: prod.distrib prod-constant)
  also have  $(\prod j \in S. A j) * (\prod j \in S. B j) = \text{linPoleProd } (\text{signedPoles } S)$ 
proof -
  have inj1: inj-on int S by simp
  have inj2: inj-on  $(\lambda j. - \text{int } j)$  S by (auto intro: inj-onI)
  have  $\bigwedge x j. \text{int } x = - \text{int } j \implies x = 0$  by linarith
  with S0 have int ' S  $\cap (\lambda j. - \text{int } j)$  ' S = {} by fastforce
  with S have linPoleProd (signedPoles S)
    =  $(\prod r \in \text{int ' S}. pX - [: \text{of-int } r: ]) * (\prod r \in$ 
       $(\lambda j. - \text{int } j)$  ' S.  $pX - [: \text{of-int } r: ])$ 
    unfolding linPoleProd-def signedPoles-def by (simp add: prod.union-disjoint)
  also have ... =  $(\prod j \in S. A j) * (\prod j \in S. B j)$ 
    by (simp add: prod.reindex[OF inj1] prod.reindex[OF inj2] A-def B-def o-def)
  finally show ?thesis by simp
qed
finally show ?thesis by (simp add: const-poly-power)
qed

lemma kappa-odd-muMono:  $(- 1) ^ e * \text{kappa } (2 * e + 5) = \text{muMono } e$ 
  by (simp add: kappa-def muMono-def algebra-simps)

lemma tau-pullback-monom:
  tau  $(pX ^ 5 * \text{pcompose } (\text{Polynomial.monom } c e) (- (pX ^ 2))) = c * \text{muMono } e$ 
proof -
  have m: Polynomial.monom c e = Polynomial.smult c  $(pX ^ e)$ 
    unfolding pX-pow-monom by (simp add: smult-monom)
  have neg:  $(- (pX ^ 2) :: \text{rat poly}) ^ e = [:(- 1) ^ e:] * pX ^ (2 * e)$ 
proof -
  have  $(- (pX ^ 2) :: \text{rat poly}) ^ e = (- 1) ^ e * (pX ^ 2) ^ e$  by (rule
power-minus)
  also have  $(- 1 :: \text{rat poly}) ^ e = [:(- 1) ^ e:]$ 
    unfolding minus-one-poly by (rule const-poly-power)
  also have  $(pX ^ 2) ^ e = pX ^ (2 * e)$  by (simp add: power-mult)
  finally show ?thesis .
qed
have  $pX ^ 5 * \text{pcompose } (\text{Polynomial.monom } c e) (- (pX ^ 2))$ 
  = Polynomial.smult  $(c * (- 1) ^ e) (pX ^ (2 * e + 5))$ 
  unfolding m pcompose-smult pcompose-power-left pcompose-pX neg
  by (simp add: power-add ac-simps)
hence tau  $(pX ^ 5 * \text{pcompose } (\text{Polynomial.monom } c e) (- (pX ^ 2))) =$ 
   $c * ((- 1) ^ e * \text{kappa } (2 * e + 5))$ 
  by (simp only: tau-smult tau-X-pow-eq-kappa mult.assoc)
thus ?thesis by (simp only: kappa-odd-muMono)
qed

lemma tau-pullback-poly: tau  $(pX ^ 5 * \text{pcompose } Q (- (pX ^ 2))) = \text{muPoly } Q$ 
proof -
  define M where M e = Polynomial.monom (Polynomial.coeff Q e) e for e

```

have $e: (\sum e \leq \text{degree } Q. M e) = Q$ **unfolding** $M\text{-def}$ **by** (*rule poly-as-sum-of-monoms*)
have $\text{tau } (pX \wedge 5 * \text{pcompose } Q (- (pX \wedge 2))) = \text{tau } (pX \wedge 5 * \text{pcompose } (\sum$
 $e \leq$
 $\text{degree } Q. M e) (- (pX \wedge 2)))$
by (*simp only: e*)
also have $\dots = (\sum e \leq \text{degree } Q. \text{tau } (pX \wedge 5 * \text{pcompose } (M e) (- (pX \wedge$
 $2))))$
by (*simp only: pcompose-sum sum-distrib-left tau-sum*)
also have $\dots = (\sum e \leq \text{degree } Q. \text{Polynomial.coeff } Q e * \text{muMono } e)$
by (*simp only: M-def tau-pullback-monom*)
finally show $?thesis$ **by** (*simp add: muPoly-def*)
qed

lemma $\text{sqPoleProd-nonzero: finite } S \implies \text{sqPoleProd } S \neq 0$

proof –
assume $S: \text{finite } S$
have $pX + [:of\text{-nat } j \wedge 2:] \neq (0 :: \text{rat poly})$ **for** j
proof
assume $pX + [:of\text{-nat } j \wedge 2:] = (0 :: \text{rat poly})$
hence $\text{Polynomial.coeff } (pX + [:of\text{-nat } j \wedge 2:]) 1 = 0$ **by** *simp*
thus False **by** (*simp add: coeff-monom*)
qed
thus $?thesis$ **using** S **by** (*simp add: sqPoleProd-def*)
qed

lemma $\text{linPoleProd-nonzero: finite } T \implies \text{linPoleProd } T \neq 0$

proof –
assume $T: \text{finite } T$
have $pX - [:of\text{-int } r:] \neq (0 :: \text{rat poly})$ **for** r
proof
assume $pX - [:of\text{-int } r:] = (0 :: \text{rat poly})$
hence $\text{Polynomial.coeff } (pX - [:of\text{-int } r:]) 1 = 0$ **by** *simp*
thus False **by** (*simp add: coeff-monom*)
qed
thus $?thesis$ **using** T **by** (*simp add: linPoleProd-def*)
qed

lemma muX-poly-mult:

assumes $S: \text{finite } S$
shows $\text{muX } S (\text{sqPoleProd } S * Q) = [: \text{muPoly } Q:]$
proof –
have $q: (\text{sqPoleProd } S * Q) \text{ div } \text{sqPoleProd } S = Q$
using $\text{sqPoleProd-nonzero}[OF S]$ **by** *simp*
have $r: \text{sqResidue } S (\text{sqPoleProd } S * Q) j = 0$ **if** $j: j \in S$ **for** j
proof –
have $\text{poly } (\text{sqPoleProd } S) (- (of\text{-nat } j \wedge 2)) = 0$
unfolding $\text{sqPoleProd-def poly-prod}$ **using** $S j$ **by** (*auto simp: prod-zero-iff poly-monom*)
thus $?thesis$ **by** (*simp add: sqResidue-def*)

```

qed
show ?thesis unfolding muX-def q using r by simp
qed

lemma tauX-poly-mult:
  assumes T: finite T
  shows tauX T (G * linPoleProd T) = [:tau G:]
proof -
  have q: (G * linPoleProd T) div linPoleProd T = G
    using linPoleProd-nonzero[OF T] by simp
  have r: linResidue T (G * linPoleProd T) r = 0 if r: r ∈ T for r
  proof -
    have poly (linPoleProd T) (of-int r) = 0
      unfolding linPoleProd-def poly-prod using T r by (auto simp: prod-zero-iff
poly-monom)
    thus ?thesis by (simp add: linResidue-def)
  qed
  show ?thesis unfolding tauX-def q using r by simp
qed

lemma pullback-poly:
  assumes S: finite S and S0: 0 ∉ S
  shows muX S (sqPoleProd S * Q) = pullbackR S (sqPoleProd S * Q)
proof -
  define c :: rat where c = (- 1) ^ card S
  define L where L = linPoleProd (signedPoles S)
  have cc: [:c:] * [:c:] = 1 by (simp add: c-def power-mult-distrib[symmetric])
  have TS: finite (signedPoles S) using S by (simp add: signedPoles-def)
  have [:c:] * pX ^ 5 * pcompose (sqPoleProd S * Q) (- (pX ^ 2))
    = ([:c:] * [:c:]) * (pX ^ 5 * pcompose Q (- (pX ^ 2))) * L
    unfolding pcompose-mult pcompose-sqPoleProd[OF S S0] c-def L-def by (simp
only: ac-simps)
  also have ... = (pX ^ 5 * pcompose Q (- (pX ^ 2))) * L by (simp only: cc
mult-1)
  finally have F: [:c:] * pX ^ 5 * pcompose (sqPoleProd S * Q) (- (pX ^ 2)) =
    (pX ^ 5 * pcompose Q (- (pX ^ 2))) * L .
  show ?thesis
    unfolding pullbackR-def c-def[symmetric] F L-def tauX-poly-mult[OF TS]
muX-poly-mult[OF S]
    tau-pullback-poly ..
qed

lemma degree-pX-plus-const: degree (pX + [:a:]) = 1
  by (simp add: monom-Suc)

lemma degree-pX-minus-const: degree (pX - [:a:]) = 1
  by (simp add: monom-Suc)

lemma pX-plus-const-nonzero: pX + [:a:] ≠ 0

```

```

using degree-pX-plus-const[of a] by auto

lemma pX-minus-const-nonzero:  $pX - [:a:] \neq 0$ 
  using degree-pX-minus-const[of a] by auto

lemma muX-single-pole:
  assumes  $S$ : finite  $S$  and  $j$ :  $j \in S$ 
  shows  $\text{muX } S (\text{sqPoleProd } (S - \{j\})) = \text{muPole } j$ 
proof -
  define  $Pj$  where  $Pj = \text{sqPoleProd } (S - \{j\})$ 
  have  $D$ :  $\text{sqPoleProd } S = (pX + [: \text{of-nat } j^2 :]) * Pj$ 
    unfolding sqPoleProd-def Pj-def using  $S$   $j$  by (simp add: prod.remove)
  have  $Pj0$ :  $Pj \neq 0$  unfolding Pj-def using  $S$  by (simp add: sqPoleProd-nonzero)
  have  $\text{degree } (\text{sqPoleProd } S) = 1 + \text{degree } Pj$ 
    unfolding  $D$  degree-mult-eq[OF pX-plus-const-nonzero  $Pj0$ ] degree-pX-plus-const
  ..
  hence  $q$ :  $Pj \text{ div } \text{sqPoleProd } S = 0$  by (intro div-poly-less) simp
  have  $r$ :  $\text{sqResidue } S Pj k = (\text{if } k = j \text{ then } 1 \text{ else } 0)$  if  $k$ :  $k \in S$  for  $k$ 
  proof (cases  $k = j$ )
    case True
      have  $\text{nz}$ :  $(\prod l \in S - \{j\}. \text{of-nat } l^2 - \text{of-nat } j^2 :: \text{rat}) \neq 0$ 
        using  $S$  by (simp add: prod-zero-iff)
      have  $\text{poly } Pj (- (\text{of-nat } j^2)) = (\prod l \in S - \{j\}. \text{of-nat } l^2 - \text{of-nat } j^2)$ 
        unfolding Pj-def sqPoleProd-def poly-prod by (simp add: poly-monom)
      thus ?thesis using True  $\text{nz}$  by (simp add: sqResidue-def)
    next
      case False
      have  $\text{poly } Pj (- (\text{of-nat } k^2)) = 0$ 
        unfolding Pj-def sqPoleProd-def poly-prod using  $S$   $k$  False
        by (auto simp: prod-zero-iff poly-monom intro!: bexI[of -  $k$ ])
      thus ?thesis using False by (simp add: sqResidue-def)
  qed
  have  $\text{muX } S Pj = [: \text{muPoly } 0:] + (\sum k \in S. [: \text{if } k = j \text{ then } 1 \text{ else } 0:] * \text{muPole } k)$ 
  unfolding muX-def  $q$  using  $r$  by (simp cong: sum.cong)
  also have  $(\sum k \in S. [: \text{if } k = j \text{ then } 1 \text{ else } 0:] * \text{muPole } k) = (\sum k \in S. \text{if } k = j \text{ then } \text{muPole } k \text{ else } 0)$ 
    by (rule sum.cong) auto
  also have  $[: \text{muPoly } 0:] + (\sum k \in S. \text{if } k = j \text{ then } \text{muPole } k \text{ else } 0) = \text{muPole } j$ 
    using  $S$   $j$  by (simp add: muPoly-def)
  finally show ?thesis unfolding Pj-def .
qed

lemma quintic-division-identity:
  fixes  $x$   $A$  :: 'a :: comm-ring-1
  shows  $x^5 = ((x - A) * (x - A)) * (x^3 + A^2 * x) + A^4 * x$ 
  by (simp add: numeral-eq-Suc algebra-simps)

lemma H5-Suc-pred:  $j \neq 0 \implies H5 j = H5 (j - 1) + 1 / \text{of-nat } j^5$ 

```

```

proof –
  assume  $j \neq 0$ 
  then obtain  $i$  where  $i: j = \text{Suc } i$  by (cases j) auto
  show ?thesis unfolding  $i$   $H5\text{-def}$  by simp
qed

lemma pullbackR-single-pole:
  assumes  $S$ : finite S and  $S0$ :  $0 \notin S$  and  $j$ :  $j \in S$ 
  shows  $\text{pullbackR } S (\text{sqPoleProd } (S - \{j\})) = \text{muPole } j$ 
proof –
  define  $S'$  where  $S' = S - \{j\}$ 
  define  $T'$  where  $T' = \text{signedPoles } S'$ 
  define  $M$  where  $M = \text{linPoleProd } T'$ 
  define  $a :: \text{rat}$  where  $a = \text{of-nat } j$ 
  define  $A :: \text{rat poly}$  where  $A = [:a:]$ 
  have  $j0$ :  $j \neq 0$ 
  proof
    assume  $j = 0$ 
    thus False using  $S0$   $j$  by simp
  qed
  have  $a0$ :  $a \neq 0$  using  $j0$  by (simp add: a-def)
  have  $a5$ :  $a^5 = a * a^4$  by (simp add: eval-nat-numeral)
  have  $S'f$ : finite S' 0 ∉ S' using  $S$   $S0$  by (auto simp: S'-def)
  have  $T'f$ : finite T' using  $S'f$  by (simp add: T'-def signedPoles-def)
  have  $\text{card } S$ :  $\text{card } S = \text{Suc } (\text{card } S')$  using  $S$   $j$  card-gt-0-iff[of S] by (auto simp: S'-def)
  have  $nn$ :  $\bigwedge x y :: \text{nat. int } x = - \text{int } y \implies x = 0$  by linarith
  have  $jT1$ :  $\text{int } j \notin T'$  using  $j0$   $nn$  by (auto simp: T'-def signedPoles-def S'-def)
  have  $jT2$ :  $-\text{int } j \notin T'$  using  $j0$   $nn$  by (auto simp: T'-def signedPoles-def S'-def)
  have  $\text{Teq}$ :  $\text{signedPoles } S = \text{insert } (\text{int } j) (\text{insert } (- \text{int } j) T')$ 
    using  $j$  by (auto simp: T'-def signedPoles-def S'-def)
  have  $jne$ :  $\text{int } j \neq - \text{int } j$  using  $j0$  by simp
  have  $M0$ :  $M \neq 0$  unfolding  $M\text{-def}$  by (rule linPoleProd-nonzero[OF T'f])
  have  $L$ :  $\text{linPoleProd } (\text{signedPoles } S) = (pX - A) * ((pX - -A) * M)$ 
    unfolding  $\text{Teq}$   $\text{linPoleProd-def}$   $M\text{-def}$  using  $T'f$   $jT1$   $jT2$   $jne$  by (simp add: A-def a-def)
  – the pulled-back numerator
  define  $F$  where  $F = [:(-1)^{\text{card } S}:] * pX^5 * \text{pcompose } (\text{sqPoleProd } S') (- (pX^2))$ 
  have  $\text{sq1}$ :  $(-1 :: \text{rat})^n * (-1)^n = 1$  for  $n$  by (simp add: power-mult-distrib[symmetric])
  have  $F'$ :  $F = - (pX^5 * M)$ 
  proof –
    have  $F = ([:(-1)^{\text{Suc } (\text{card } S')}:] * [:(-1)^{\text{card } S'}:] * (pX^5 * M))$ 
    unfolding  $F\text{-def}$   $\text{pcompose-sqPoleProd[OF S'f]}$   $\text{card } S$   $M\text{-def}$   $T'\text{-def}$  by (simp only: ac-simps)
    also have  $[:(-1)^{\text{Suc } (\text{card } S')}:] * [:(-1)^{\text{card } S'}:] = [: - 1 :: \text{rat}:]$ 
    using  $\text{sq1}$  [of card S'] by simp
    finally show ?thesis by simp
  qed

```

— polynomial division by the pole product

```

define  $G$  where  $G = - (pX \wedge 3 + A \wedge 2 * pX)$ 
define  $R$  where  $R = - (A \wedge 4 * pX * M)$ 
have  $FGR$ :  $F = R + G * \text{linPoleProd } (\text{signedPoles } S)$ 
  unfolding  $F'$   $L$   $G$ -def  $R$ -def quintic-division-identity[of  $pX$   $A$ ] by (simp add:
algebra-simps)
  have  $L0$ :  $\text{linPoleProd } (\text{signedPoles } S) \neq 0$ 
  unfolding  $L$  using  $M0$   $pX$ -minus-const-nonzero[of  $a$ ]  $pX$ -minus-const-nonzero[of
  -  $a$ ]
  by (simp add:  $A$ -def)
  have  $dL$ :  $\text{degree } (\text{linPoleProd } (\text{signedPoles } S)) = 2 + \text{degree } M$ 
proof -
  have  $n1$ :  $pX - A \neq 0$   $pX - - A \neq 0$ 
    using  $pX$ -minus-const-nonzero[of  $a$ ]  $pX$ -minus-const-nonzero[of -  $a$ ] by
(simp-all add:  $A$ -def)
  have  $n2$ :  $(pX - - A) * M \neq 0$  using  $n1(2)$   $M0$  by simp
  have  $d1$ :  $\text{degree } (pX - A) = 1$   $\text{degree } (pX - - A) = 1$ 
    using  $\text{degree-}pX$ -minus-const[of  $a$ ]  $\text{degree-}pX$ -minus-const[of -  $a$ ] by (simp-all
add:  $A$ -def)
  show ?thesis unfolding  $L$   $\text{degree-mult-eq}$ [OF  $n1(1)$   $n2$ ]  $\text{degree-mult-eq}$ [OF
 $n1(2)$   $M0$ ]  $d1$  by simp
qed
have  $dR$ :  $\text{degree } R < \text{degree } (\text{linPoleProd } (\text{signedPoles } S))$ 
proof -
  have  $A \wedge 4 * pX * M = \text{Polynomial.smult } (a \wedge 4) (pX * M)$ 
    by (simp add:  $A$ -def const-poly-power)
  hence  $\text{degree } R \leq \text{degree } (pX * M)$  unfolding  $R$ -def by (simp add: de-
gree-smult-le)
  also have  $\dots \leq \text{degree } (pX :: \text{rat poly}) + \text{degree } M$  by (rule degree-mult-le)
  also have  $\text{degree } (pX :: \text{rat poly}) = 1$  by (simp add: degree-monom-eq)
  finally show ?thesis unfolding  $dL$  by simp
qed
have quot:  $F \text{ div } \text{linPoleProd } (\text{signedPoles } S) = G$ 
  unfolding  $FGR$  div-mult-self1[OF  $L0$ ] div-poly-less[OF  $dR$ ] by simp
have  $\text{tau}G$ :  $\text{tau } G = - (1 / 4)$ 
proof -
  have  $G = - (pX \wedge 3 + \text{Polynomial.smult } (a \wedge 2) (pX \wedge 1))$ 
    by (simp add:  $G$ -def  $A$ -def const-poly-power)
  hence  $\text{tau } G = - (kappa\ 3 + a \wedge 2 * kappa\ 1)$ 
    by (simp only: tau-uminus tau-add tau-smult tau-X-pow-eq-kappa)
  thus ?thesis by (simp add: kappa-def bernoulli-0 numeral-3-eq-3)
qed
— residues
have  $\text{poly}M$ :  $\text{poly } M\ x = (\prod s \in T'.\ x - \text{of-int } s)$  for  $x$ 
  unfolding  $M$ -def  $\text{linPoleProd}$ -def poly-prod by (simp add: poly-monom)
have  $\text{poly}F$ :  $\text{poly } F\ x = - (x \wedge 5 * \text{poly } M\ x)$  for  $x$  unfolding  $F'$  by (simp add:
poly-monom)
have  $PM0$ :  $\text{poly } M\ x \neq 0$  if  $x = a \vee x = - a$  for  $x$ 
proof -

```

```

have  $x - \text{of-int } s \neq 0$  if  $s: s \in T'$  for  $s$ 
proof
  assume  $x - \text{of-int } s = 0$ 
  hence  $x = \text{of-int } s$  by simp
  hence  $s = \text{int } j \vee s = - \text{int } j$  using  $\langle x = a \vee x = -a \rangle$ 
  by (auto simp: a-def)
  thus False using  $s \text{ jT1 jT2}$  by auto
qed
thus ?thesis unfolding polyM using  $T'f$  by (simp add: prod-zero-iff)
qed
define  $lr$  where  $lr \ r = \text{linResidue } (\text{signedPoles } S) \ F \ r$  for  $r$ 
have  $lrT': lr \ r = 0$  if  $r: r \in T'$  for  $r$ 
proof -
  have  $\text{poly } M \ (\text{of-int } r) = 0$  unfolding polyM using  $T'f \ r$  by (auto simp:
prod-zero-iff)
  thus ?thesis by (simp add: lr-def linResidue-def polyF)
qed
have  $lrj: lr \ (\text{int } j) = - (a^4 / 2)$ 
proof -
  have  $e: \text{signedPoles } S - \{\text{int } j\} = \text{insert } (- \text{int } j) \ T'$  unfolding Teq using
jT1 jne by auto
  have  $(\prod s \in \text{signedPoles } S - \{\text{int } j\}. \text{of-int } (\text{int } j) - \text{of-int } s) =$ 
 $(a - -a) * \text{poly } M \ a$ 
  unfolding e polyM using  $T'f \ jT2$  by (simp add: a-def)
  hence  $lr \ (\text{int } j) = - (a^5 * \text{poly } M \ a) / ((a - -a) * \text{poly } M \ a)$ 
  by (simp add: lr-def linResidue-def polyF a-def)
  also have  $\dots = - (a^4 / 2)$  using a0 PM0[of a] unfolding a5 by (simp
add: field-simps)
  finally show ?thesis .
qed
have  $lrmj: lr \ (- \text{int } j) = - (a^4 / 2)$ 
proof -
  have  $e: \text{signedPoles } S - \{- \text{int } j\} = \text{insert } (\text{int } j) \ T'$  unfolding Teq using
jT2 jne by auto
  have  $(\prod s \in \text{signedPoles } S - \{- \text{int } j\}. \text{of-int } (- \text{int } j) - \text{of-int } s) =$ 
 $(-a - a) * \text{poly } M \ (-a)$ 
  unfolding e polyM using  $T'f \ jT1$  by (simp add: a-def)
  hence  $lr \ (- \text{int } j) = - ((-a)^5 * \text{poly } M \ (-a)) / ((-a - a) * \text{poly } M \ (-a))$ 
  by (simp add: lr-def linResidue-def polyF a-def)
  also have  $\dots = - (a^4 / 2)$  using a0 PM0[of -a] by (simp add: field-simps
a5)
  finally show ?thesis .
qed
have  $tp1: \text{tauPole } (\text{int } j) = [:H5 \ j:] - pX$  by (simp add: tauPole-def dIdx-def)
have  $tp2: \text{tauPole } (- \text{int } j) = [:H5 \ (j - 1):] - pX$ 
  using j0 by (simp add: tauPole-def dIdx-def nat-diff-distrib')
have  $\text{sumres}: (\sum r \in \text{signedPoles } S. [:lr \ r:] * \text{tauPole } r)$ 
 $= [: - (a^4 / 2):] * ([:H5 \ j:] - pX) + [: - (a^4 / 2):] * ([:H5 \ (j - 1):] -$ 

```



```

pX)
  proof -
    have ( $\sum r \in T'. [lr\ r:] * \text{tauPole } r = 0$ ) by (rule sum.neutral) (simp add:
    lrT')
    thus ?thesis unfolding Teq using T'f jT1 jT2 jne by (simp add: lrj lrmj tp1
    tp2)
  qed
  have pullbackR S (sqPoleProd S') = [ $:- (1 / 4):$ ] + ([ $:- (a^4 / 2):$ ] * ([ $:H5\ j:$ ] - pX)
    + [ $:- (a^4 / 2):$ ] * ([ $:H5\ (j - 1):$ ] - pX))
  unfolding pullbackR-def F-def[symmetric] tauX-def quot tauG lr-def[symmetric]
  sumres ..
  also have ... = muPole j
  proof (rule poly-eqI)
    fix n
    have H:  $H5\ j = H5\ (j - 1) + 1 / a^5$  using H5-Suc-pred[OF j0] by (simp
    add: a-def)
    show Polynomial.coeff ([ $:- (1 / 4):$ ] + ([ $:- (a^4 / 2):$ ] * ([ $:H5\ j:$ ] - pX)
      + [ $:- (a^4 / 2):$ ] * ([ $:H5\ (j - 1):$ ] - pX))) n = Polynomial.coeff
    (muPole j) n
    proof (cases n = 0)
      case True
        have i:  $a^4 * (1 / a^5) = 1 / a$  using a0 unfolding a5 by simp
        have c0L: Polynomial.coeff ([ $:- (1 / 4):$ ] + ([ $:- (a^4 / 2):$ ] * ([ $:H5\ j:$ ] -
        pX)
          + [ $:- (a^4 / 2):$ ] * ([ $:H5\ (j - 1):$ ] - pX))) 0
          =  $-(1 / 4) - a^4 / 2 * H5\ j - a^4 / 2 * H5\ (j - 1)$ 
        by (simp add: coeff-monom)
        have c0R: Polynomial.coeff (muPole j) 0 =  $-(a^4 * H5\ j) - 1 / 4 + 1 / (2 * a)$ 
        unfolding muPole-def a-def[symmetric] by (simp add: coeff-monom)
        have  $-(1 / 4) - a^4 / 2 * H5\ j - a^4 / 2 * H5\ (j - 1) =$ 
           $-(a^4 * H5\ j) - 1 / 4 + 1 / (2 * a)$ 
        unfolding H using i by (simp add: algebra-simps add-divide-distrib)
        thus ?thesis unfolding True c0L c0R .
      case False
        then obtain m where  $n = \text{Suc } m$  by (cases n) auto
        thus ?thesis unfolding muPole-def a-def[symmetric] by (cases m) (simp-all
        add: coeff-monom)
    qed
  qed
  finally show ?thesis unfolding S'-def .
qed

```

lemma sq-lagrange:

assumes S: finite S

shows $P = \text{sqPoleProd } S * (P \text{ div } \text{sqPoleProd } S) + (\sum j \in S. [\text{sqResidue } S\ P\ j:] * \text{sqPoleProd } (S - \{j\}))$

```

proof -
  define  $g :: \text{nat} \Rightarrow \text{rat}$  where  $g\ j = - (of\text{-}nat\ j \wedge 2)$  for  $j$ 
  define  $S'$  where  $S' = g \text{ ` } S$ 
  have  $inj$ :  $inj\text{-}on\ g\ S$  by (auto intro!:  $inj\text{-}onI$  simp:  $g\text{-}def$ )
  have  $S'f$ :  $finite\ S'$  using  $S$  by (simp add:  $S'\text{-}def$ )
  have  $fac$ :  $pX - [:g\ j:] = pX + [:of\text{-}nat\ j \wedge 2:]$  for  $j$  by (simp add:  $g\text{-}def$ )
  have  $qp$ :  $qPoleProd\ (g \text{ ` } T) = sqPoleProd\ T$  if  $T \subseteq S$  for  $T$ 
  proof -
    have  $iT$ :  $inj\text{-}on\ g\ T$  using  $inj$  that by (rule  $inj\text{-}on\text{-}subset$ )
    show ?thesis unfolding  $qPoleProd\text{-}def\ sqPoleProd\text{-}def\ prod.reindex[OF\ iT]$ 
  o-def fac ..
  qed
  have  $rm$ :  $S' - \{g\ j\} = g \text{ ` } (S - \{j\})$  if  $j \in S$  for  $j$ 
    using  $inj$  that unfolding  $S'\text{-}def$  by (auto simp:  $inj\text{-}on\text{-}def$ )
  have  $res$ :  $qRes\ S'\ P\ (g\ j) = sqResidue\ S\ P\ j$  if  $j \in S$  for  $j$ 
  proof -
    have  $i2$ :  $inj\text{-}on\ g\ (S - \{j\})$  using  $inj$  by (rule  $inj\text{-}on\text{-}subset$ ) auto
    have  $(\prod s' \in S' - \{g\ j\}. g\ j - s') = (\prod k \in S - \{j\}. g\ j - g\ k)$ 
      unfolding  $rm[OF\ j]$  by (simp add:  $prod.reindex[OF\ i2]$ )
    also have  $\dots = (\prod k \in S - \{j\}. of\text{-}nat\ k \wedge 2 - of\text{-}nat\ j \wedge 2)$ 
      by (simp add:  $g\text{-}def$ )
    finally show ?thesis by (simp add:  $qRes\text{-}def\ sqResidue\text{-}def\ g\text{-}def$ )
  qed
  have  $P = qPoleProd\ S' * (P\ div\ qPoleProd\ S') + (\sum s \in S'. [:qRes\ S'\ P\ s:] * qPoleProd\ (S' - \{s\}))$ 
    by (rule  $lagrange\text{-}decomp[OF\ S'f]$ )
  also have  $qPoleProd\ S' = sqPoleProd\ S$  unfolding  $S'\text{-}def$  by (rule  $qp$ ) simp
  also have  $(\sum s \in S'. [:qRes\ S'\ P\ s:] * qPoleProd\ (S' - \{s\}))$ 
     $= (\sum j \in S. [:qRes\ S'\ P\ (g\ j):] * qPoleProd\ (S' - \{g\ j\}))$ 
    unfolding  $S'\text{-}def$  by (simp add:  $sum.reindex[OF\ inj]$ )
  also have  $\dots = (\sum j \in S. [:sqResidue\ S\ P\ j:] * sqPoleProd\ (S - \{j\}))$ 
    by (rule  $sum.cong$ ) (simp-all add:  $res\ rm\ qp$ )
  finally show ?thesis .
qed

theorem pullback-aux:
  assumes  $S$ :  $finite\ S$  and  $S0$ :  $0 \notin S$ 
  shows  $\mu X\ S\ P = pullbackR\ S\ P$ 
  proof -
    define  $Q$  where  $Q = P\ div\ sqPoleProd\ S$ 
    define  $r$  where  $r\ j = sqResidue\ S\ P\ j$  for  $j$ 
    have  $P$ :  $P = sqPoleProd\ S * Q + (\sum j \in S. [:r\ j:] * sqPoleProd\ (S - \{j\}))$ 
      unfolding  $Q\text{-}def\ r\text{-}def$  by (rule  $sq\text{-}lagrange[OF\ S]$ )
    have  $\mu X\ S\ P = \mu X\ S\ (sqPoleProd\ S * Q) + (\sum j \in S. [:r\ j:] * \mu X\ S\ (sqPoleProd\ (S - \{j\})))$ 
      by (subst  $P$ ) (simp only:  $\mu X\text{-}add\ \mu X\text{-}sum\ \mu X\text{-}C\text{-}mul$ )
    also have  $\dots = pullbackR\ S\ (sqPoleProd\ S * Q) + (\sum j \in S. [:r\ j:] * pullbackR\ S\ (sqPoleProd\ (S - \{j\})))$ 
      by (simp add:  $pullback\text{-}poly[OF\ S\ S0]\ \mu X\text{-}single\text{-}pole[OF\ S]\ pullbackR\text{-}single\text{-}pole[OF\ S]$ )

```

```

S S0])
  also have ... = pullbackR S P
    by (subst P) (simp only: pullbackR-add pullbackR-sum pullbackR-C-mul)
  finally show ?thesis .
qed

mu-X(R) =  $\tau$ -X( $x^5 R(-x^2)$ ); the numerator picks up  $(-1)^{|S|}$  because
 $\prod_{j \in S} (j^2 - x^2) = (-1)^{|S|} \prod_{r=\pm j} (x - r)$ .

theorem pullback:
  assumes finite S 0  $\notin$  S
  shows muX S P = tauX (signedPoles S) ( $[-1]^{card S} * pX^5 * pcompose$ 
P ( $-(pX^2)$ ))
  using pullback-aux[OF assms] by (simp only: pullbackR-def)

end

theory Zeta5-IntValued
  imports Zeta5-Functionals
begin

```

16 Integer-valued polynomials and the bound (3.9)

Binomial polynomials, forward differences and Newton interpolation, used as a self-contained replacement for the $\mathbb{Z}_p$ -analytic arguments of Section 3.3 of the paper: values at integers with valuation bounds give coefficient bounds ((3.8) and (3.9), *tau-vge-of-values*, *diffQuot-vge*). Lean theory *IntValued*.

```

lemma IntValuedAt-vge: IntValuedAt p A  $\implies$  vge p (poly A (of-int x)) 0
  by (simp add: IntValuedAt-def vge-def)

```

16.1 Binomial polynomials

$binom(x, k) = x(x-1)\cdots(x-k+1)/k!$.

```

definition bp :: nat  $\Rightarrow$  rat poly where
  bp k = Polynomial.smult (1 / fact k) ( $\prod_{i < k} pX - [:of-nat i:]$ )

```

```

lemma bp-0: bp 0 = 1
  by (simp add: bp-def)

```

```

lemma bp-Suc: bp (Suc k) = Polynomial.smult (1 / (of-nat k + 1)) (bp k * (pX
- [:of-nat k:]))
  unfolding bp-def prod.lessThan-Suc
  by (simp add: fact-Suc smult-smult mult-smult-left mult.commute add.commute)

```

```

lemma degree-bp-le: degree (bp k)  $\leq$  k
proof (induction k)
  case 0
  thus ?case by (simp add: bp-0)

```

```

next
  case (Suc k)
  have degree (bp (Suc k)) ≤ degree (bp k * (pX - [:of-nat k:]))
    unfolding bp-Suc by (rule degree-smult-le)
  also have ... ≤ degree (bp k) + degree (pX - [:of-nat k:]) by (rule degree-mult-le)
  also have ... ≤ Suc k unfolding degree-pX-minus-const using Suc.IH by simp
  finally show ?case .
qed

```

$\text{binom}(x, k)$ is HOL's $(g\text{choose})$.

```

lemma poly-bp: poly (bp k) a = a gchoose k
  by (simp add: bp-def poly-prod gbinomial-prod-rev atLeast0LessThan poly-monom)

```

$\text{binom}(n, k)$ at naturals.

```

lemma bp-eval-nat: poly (bp k) (of-nat n) = of-nat (n choose k)
  by (simp add: poly-bp binomial-gbinomial)

```

$\text{binom}(-(t+1), k) = (-1)^k \text{binom}(t+k, k)$.

```

lemma bp-eval-neg: poly (bp k) (- (of-nat t + 1)) = (-1) ^ k * of-nat ((t + k)
choose k)
  unfolding poly-bp gbinomial-minus by (simp add: binomial-gbinomial)

```

Binomial polynomials are integer-valued on $\mathbb{Z}$.

```

lemma bp-eval-int: ∃ z::int. poly (bp k) (of-int m) = of-int z
proof (cases 0 ≤ m)
  case True
  then obtain n where n: m = int n by (metis nonneg-int-cases)
  have poly (bp k) (of-int m) = of-int (int (n choose k)) unfolding n by (simp
add: bp-eval-nat)
  thus ?thesis by blast
next
  case False
  then obtain t where t: m = - (int t + 1) by (intro that[of nat (- m - 1)])
simp
  have poly (bp k) (of-int m) = of-int ((-1) ^ k * int ((t + k) choose k))
    unfolding t using bp-eval-neg[of k t] by simp
  thus ?thesis by blast
qed

```

Two polynomials agreeing on all naturals are equal.

```

lemma poly-eq-of-eval-nat:
  fixes P Q :: rat poly
  assumes ∧ n::nat. poly P (of-nat n) = poly Q (of-nat n)
  shows P = Q
proof (rule ccontr)
  assume P ≠ Q
  hence P - Q ≠ 0 by simp
  hence fin: finite {x. poly (P - Q) x = 0} by (rule poly-roots-finite)

```

have $\text{range } (\text{of-nat} :: \text{nat} \Rightarrow \text{rat}) \subseteq \{x. \text{poly } (P - Q) x = 0\}$ **using** *assms*
by *auto*
hence $\text{finite } (\text{range } (\text{of-nat} :: \text{nat} \Rightarrow \text{rat}))$ **using** *fin finite-subset* **by** *blast*
moreover **have** $\text{inj } (\text{of-nat} :: \text{nat} \Rightarrow \text{rat})$ **by** (*simp add: inj-on-def*)
ultimately show *False* **using** *finite-imageD infinite-UNIV-nat* **by** *blast*
qed

Pascal: $\Delta \text{binom}(x, k+1) = \text{binom}(x, k)$.

lemma *bp-fwd*: $\text{pcompose } (\text{bp } (\text{Suc } k)) (pX + 1) - \text{bp } (\text{Suc } k) = \text{bp } k$
by (*rule poly-eq-of-eval-nat*) (*simp add: poly-pcompose poly-bp poly-monom gbinomial-Suc-Suc*)

$\text{binom}(x, k+1) = (x/(k+1)) \text{binom}(x-1, k)$.

lemma *bp-shift*: $\text{bp } (\text{Suc } k) = \text{Polynomial.smult } (1 / (\text{of-nat } k + 1)) (pX * \text{pcompose } (\text{bp } k) (pX - 1))$
by (*rule poly-eq-of-eval-nat*)
(simp add: poly-pcompose poly-bp poly-monom gbinomial-absorption'[of Suc k] add.commute)

$\tau(\text{binom}(x, k)) = [x^4] \text{binom}(x, k+1)$.

lemma *tau-bp*: $\tau(\text{bp } k) = \text{Polynomial.coeff } (\text{bp } (\text{Suc } k)) 4$
using *tau-fwd[of bp (Suc k)]* **unfolding** *bp-fwd* .

16.2 The coefficients of $\text{binom}(x, n)$

Weighted Gauss bound: $v\text{-p}(\text{coeff-m } P) \geq -m \ c$.

definition *WB* :: $\text{nat} \Rightarrow \text{rat} \Rightarrow \text{rat poly} \Rightarrow \text{bool}$ **where**
 $\text{WB } p \ c \ P \longleftrightarrow (\forall m. \text{vge } p (\text{Polynomial.coeff } P \ m) (- \text{of-nat } m * c))$

lemma *WB-mul*: $\text{prime } p \implies \text{WB } p \ c \ P \implies \text{WB } p \ c \ Q \implies \text{WB } p \ c \ (P * Q)$

proof –

assume *p*: *prime p* **and** *hP*: $\text{WB } p \ c \ P$ **and** *hQ*: $\text{WB } p \ c \ Q$

show $\text{WB } p \ c \ (P * Q)$ **unfolding** *WB-def*

proof

fix *m*

have $\text{vge } p (\sum i \leq m. \text{Polynomial.coeff } P \ i * \text{Polynomial.coeff } Q \ (m - i)) (- \text{of-nat } m * c)$

proof (*rule vge-sum[OF p]*)

fix *i* **assume** $i \in \{..m\}$

hence *eq*: $- \text{of-nat } i * c + - \text{of-nat } (m - i) * c = - \text{of-nat } m * c$

by (*simp add: of-nat-diff algebra-simps*)

have $\text{vge } p (\text{Polynomial.coeff } P \ i * \text{Polynomial.coeff } Q \ (m - i)) (- \text{of-nat } i * c + -$

$\text{of-nat } (m - i) * c)$

using *hP hQ* **unfolding** *WB-def* **by** (*intro vge-mul[OF p]*) *auto*

thus $\text{vge } p (\text{Polynomial.coeff } P \ i * \text{Polynomial.coeff } Q \ (m - i)) (- \text{of-nat } m * c)$

```

    by (simp only: eq)
  qed simp
  thus vge p (Polynomial.coeff (P * Q) m) (- of-nat m * c) by (simp add:
coeff-mult)
  qed
  qed

```

```

lemma WB-one: WB p c 1
  by (auto simp: WB-def coeff-1 vge-one vge-zero)

```

```

lemma WB-prod:
  prime p  $\implies$  finite I  $\implies$  ( $\bigwedge i. i \in I \implies$ 
    WB p c (f i))  $\implies$  WB p c ( $\prod i \in I. f i$ )
proof -
  assume p: prime p and fin: finite I and h:  $\bigwedge i. i \in I \implies$  WB p c (f i)
  show WB p c ( $\prod i \in I. f i$ )
    using fin h by (induction I rule: finite-induct) (auto intro: WB-one WB-mul[OF
p])
  qed

```

$u X - v$ with $v \cdot p(u) \geq -c$, $v \cdot p(v) \geq 0$.

```

lemma WB-linear: prime p  $\implies$  vge p u (- c)  $\implies$ 
  vge p v 0  $\implies$  WB p c [:- v, u:]
  by (auto simp: WB-def coeff-pCons vge-neg vge-zero split: nat.split)

```

```

lemma poly-bp-factor:
  fixes a :: rat
  shows poly ([:0, 1 / (of-nat m + 1):] * ( $\prod i < m. [:- 1, 1 / (of-nat i + 1):]$ ))
a =
  a gchoose Suc m
proof (induction m)
  case 0
  thus ?case by simp
next
  case (Suc m)
  define P where P = poly ( $\prod i < m. [:- 1, 1 / (of-nat i + 1):]$ ) a
  have IH: a / (of-nat m + 1) * P = a gchoose Suc m using Suc.IH by (simp
add: P-def)
  have h1: (1 + of-nat m :: rat)  $\neq$  0 and h2: (of-nat m + 2 :: rat)  $\neq$  0
    and h3: (2 + of-nat m :: rat)  $\neq$  0 and h4: (of-nat m + 1 :: rat)  $\neq$  0
    by simp-all
  have e: (of-nat (Suc m) + 1 :: rat) = of-nat m + 2 by simp
  have poly ([:0, 1 / (of-nat (Suc m) + 1):] * ( $\prod i < Suc m. [:- 1, 1 / (of-nat
i + 1):]$ )) a
    = poly ([:0, 1 / (of-nat (Suc m) + 1):] a * (P * poly [:- 1, 1 / (of-nat m +
1):] a)
    by (simp only: prod.lessThan-Suc poly-mult P-def)
  also have ... = a / (of-nat m + 2) * (P * (a / (of-nat m + 1) - 1))
    unfolding e by simp

```

also have ... = (a / (of-nat m + 1) * P) * ((a - of-nat (Suc m)) / (of-nat (Suc m) + 1))
unfolding e **using** h1 h2 h3 h4 **by** (simp add: field-simps)
also have ... = a gchoose Suc (Suc m) **unfolding** IH **by** (rule gbinomial-Suc-rec[symmetric])
finally show ?case .
qed

$\text{binom}(x, m+1) = (x/(m+1)) \prod_{i=0}^m (x/(i+1) - 1).$

lemma bp-factor: bp (Suc m) = [:0, 1 / (of-nat m + 1):] * ($\prod_{i=0}^m$ i < m. [: - 1, 1 / (of-nat i + 1):])
by (rule poly-eq-of-eval-nat) (simp only: poly-bp poly-bp-factor)

$v\text{-}p(\lceil x \hat{4} \rceil \text{ binom}(x, n)) \geq -4 \lfloor \log\text{-}p \ n \rfloor.$

lemma bp-coeff4:
assumes p: prime p
shows vge p (Polynomial.coeff (bp n) 4) (- 4 * of-nat (nat-log p n))
proof (cases n)
case 0
thus ?thesis **by** (simp add: bp-0 vge-zero)
next
case (Suc m)
have WB p (of-nat (nat-log p (Suc m))) (bp (Suc m))
unfolding bp-factor
proof (intro WB-mul[OF p] WB-prod[OF p])
have u: vge p (1 / (of-nat m + 1)) (- of-nat (nat-log p (Suc m)))
using vge-inv-nat[OF p, of Suc m] **by** (simp add: add.commute)
show WB p (of-nat (nat-log p (Suc m))) [:0, 1 / (of-nat m + 1):]
using WB-linear[OF p u vge-zero[of p 0]] **by** simp
show finite {..by simp
fix i **assume** i ∈ {..
have u: vge p (1 / (of-nat i + 1)) (- of-nat (nat-log p (Suc i)))
using vge-inv-nat[OF p, of Suc i] **by** (simp add: add.commute)
have nat-log p (Suc i) ≤ nat-log p (Suc m) **using** ⟨i ∈ {..
by (intro nat-log-mono) simp
hence u': vge p (1 / (of-nat i + 1)) (- of-nat (nat-log p (Suc m)))
by (intro vge-mono[OF u]) simp
show WB p (of-nat (nat-log p (Suc m))) [: - 1, 1 / (of-nat i + 1):]
using WB-linear[OF p u' vge-one] .
qed
thus ?thesis **unfolding** Suc WB-def
by (auto dest: spec[of - 4])
qed

16.3 Newton interpolation

Forward differences $\Delta^k f(y)$ with step 1.

fun fd :: rat poly $\Rightarrow$ rat $\Rightarrow$ nat $\Rightarrow$ rat **where**
 fd f y 0 = poly f y

$$| \text{fd } f \ y \ (\text{Suc } k) = \text{fd } f \ (y + 1) \ k - \text{fd } f \ y \ k$$

lemma *fd-vge*:

assumes *prime* $p \wedge i. i \leq k \implies \text{vge } p \ (\text{poly } f \ (y + \text{of-nat } i)) \ b$

shows $\text{vge } p \ (\text{fd } f \ y \ k) \ b$

using *assms*(2)

proof (*induction* k *arbitrary*: y)

case 0

thus ?*case* **by** *simp*

next

case (*Suc* k)

have 1: $\text{vge } p \ (\text{fd } f \ (y + 1) \ k) \ b$

proof (*rule* *Suc.IH*)

fix i **assume** $i \leq k$

hence $\text{vge } p \ (\text{poly } f \ (y + \text{of-nat } (\text{Suc } i))) \ b$ **by** (*intro* *Suc.prem*s) *simp*

thus $\text{vge } p \ (\text{poly } f \ (y + 1 + \text{of-nat } i)) \ b$ **by** (*simp* *add*: *add-ac*)

qed

have 2: $\text{vge } p \ (\text{fd } f \ y \ k) \ b$

by (*rule* *Suc.IH*) (*use* *Suc.prem*s **in** *simp*)

show ?*case* **using** *vge-add*[*OF* *assms*(1) 1 *vge-neg*[*OF* 2]] **by** *simp*

qed

$$f(y + n) = \sum_{k \leq n} C(n, k) \Delta^k f(y).$$

lemma *newton-eval*: $\text{poly } f \ (y + \text{of-nat } n) = (\sum_{k \leq n. \text{of-nat } (n \text{ choose } k)} * \text{fd } f \ y \ k)$

proof (*induction* n *arbitrary*: y)

case 0

thus ?*case* **by** *simp*

next

case (*Suc* n)

have $\text{poly } f \ (y + \text{of-nat } (\text{Suc } n)) = \text{poly } f \ ((y + 1) + \text{of-nat } n)$ **by** (*simp* *add*: *algebra-simps*)

also have $\dots = (\sum_{k \leq n. \text{of-nat } (n \text{ choose } k)} * \text{fd } f \ (y + 1) \ k)$ **by** (*rule* *Suc.IH*)

also have $\dots = (\sum_{k \leq n. \text{of-nat } (n \text{ choose } k)} * (\text{fd } f \ y \ (\text{Suc } k) + \text{fd } f \ y \ k))$
by *simp*

also have $\dots = (\sum_{k \leq \text{Suc } n. \text{of-nat } (\text{Suc } n \text{ choose } k)} * \text{fd } f \ y \ k)$

proof –

have $(\sum_{k \leq \text{Suc } n. \text{of-nat } (n \text{ choose } k)} * \text{fd } f \ y \ k)$
 $= \text{fd } f \ y \ 0 + (\sum_{k \leq n. \text{of-nat } (n \text{ choose } \text{Suc } k)} * \text{fd } f \ y \ (\text{Suc } k))$
by (*subst* *sum.atMost-Suc-shift*) (*simp* *del*: *fd.simps*)

moreover have $(\sum_{k \leq \text{Suc } n. \text{of-nat } (n \text{ choose } k)} * \text{fd } f \ y \ k) =$
 $(\sum_{k \leq n. \text{of-nat } (n \text{ choose } k)} * \text{fd } f \ y \ k)$
by (*simp* *add*: *binomial-eq-0* *del*: *fd.simps*)

moreover have $(\sum_{k \leq \text{Suc } n. \text{of-nat } (\text{Suc } n \text{ choose } k)} * \text{fd } f \ y \ k)$
 $= \text{fd } f \ y \ 0 + (\sum_{k \leq n. \text{of-nat } (n \text{ choose } k)} * \text{fd } f \ y \ (\text{Suc } k))$
 $+ (\sum_{k \leq n. \text{of-nat } (n \text{ choose } \text{Suc } k)} * \text{fd } f \ y \ (\text{Suc } k))$

by (*subst* *sum.atMost-Suc-shift*) (*simp* *add*: *sum.distrib algebra-simps* *del*: *fd.simps*)

moreover have $(\sum k \leq n. \text{of-nat } (n \text{ choose } k) * (fd \ f \ y \ (Suc \ k) + fd \ f \ y \ k))$
 $= (\sum k \leq n. \text{of-nat } (n \text{ choose } k) * fd \ f \ y \ (Suc \ k)) +$
 $(\sum k \leq n. \text{of-nat } (n \text{ choose } k) * fd \ f \ y \ k)$
unfolding *distrib-left sum.distrib ..*
ultimately show *?thesis* **by** *linarith*
qed
finally show *?case* .
qed

Newton's forward-difference formula as a polynomial identity.

lemma *newton*:

assumes *degree f ≤ D*
shows $f = (\sum k \leq D. [:fd \ f \ y \ k:] * pcompose \ (bp \ k) \ (pX - [:y:])))$
proof –
let $?g = (\sum k \leq D. [:fd \ f \ y \ k:] * pcompose \ (bp \ k) \ (pX - [:y:])))$
have *dg: degree ?g ≤ D*
proof (*rule degree-sum-le*)
fix k **assume** $kD: k \in \{..D\}$
have $degree \ ([:fd \ f \ y \ k:] * pcompose \ (bp \ k) \ (pX - [:y:])))$
 $\leq degree \ [:fd \ f \ y \ k:] + degree \ (pcompose \ (bp \ k) \ (pX - [:y:])))$ **by** (*rule degree-mult-le*)
also have $\dots = degree \ (bp \ k)$ **unfolding** *degree-pcompose degree-pX-minus-const*
by *simp*
also have $\dots \leq D$ **using** *degree-bp-le[of k] kD* **by** *simp*
finally show $degree \ ([:fd \ f \ y \ k:] * pcompose \ (bp \ k) \ (pX - [:y:]))) \leq D$.
qed *simp*
have *agree: $\bigwedge n. n \leq D \implies poly \ ?g \ (y + \text{of-nat } n) = poly \ f \ (y + \text{of-nat } n)$*
proof –
fix n **assume** $nD: n \leq D$
have $poly \ ?g \ (y + \text{of-nat } n) = (\sum k \leq D. \text{of-nat } (n \text{ choose } k) * fd \ f \ y \ k)$
by (*simp add: poly-sum poly-pcompose poly-monom bp-eval-nat mult.commute*)
also have $\dots = (\sum k \leq n. \text{of-nat } (n \text{ choose } k) * fd \ f \ y \ k)$
by (*rule sum.mono-neutral-right*) (*use nD in $\langle auto \ simp: binomial-eq-0 \rangle$*)
also have $\dots = poly \ f \ (y + \text{of-nat } n)$ **by** (*rule newton-eval[symmetric]*)
finally show $poly \ ?g \ (y + \text{of-nat } n) = poly \ f \ (y + \text{of-nat } n)$.
qed
have $f - ?g = 0$
proof (*rule poly-eq-0-of-roots*)
show *finite* $((\lambda n. y + \text{of-nat } n) \ ' \ \{..D\})$ **by** *simp*
have *inj-on* $(\lambda n::nat. y + \text{of-nat } n) \ \{..D\}$ **by** (*simp add: inj-on-def*)
hence $card \ ((\lambda n. y + \text{of-nat } n) \ ' \ \{..D\}) = Suc \ D$ **by** (*simp add: card-image*)
moreover have $degree \ (f - ?g) \leq D$ **using** *degree-diff-le[OF assms dg]* .
ultimately show $degree \ (f - ?g) < card \ ((\lambda n. y + \text{of-nat } n) \ ' \ \{..D\})$ **by** *simp*
next
fix s **assume** $s \in (\lambda n. y + \text{of-nat } n) \ ' \ \{..D\}$
then obtain n **where** $n \leq D \ s = y + \text{of-nat } n$ **by** *auto*
thus $poly \ (f - ?g) \ s = 0$ **using** *agree* **by** *simp*
qed
thus *?thesis* **by** *simp*

qed

lemma *pcompose-pX-minus-0*: $pcompose\ P\ (pX - [:0:]) = P$
unfolding *pX-minus-const* **by** (*simp add: pcompose-idR*)

16.4 (3.9) and (3.8)

(3.9), in the form used: if $\deg f \leq D$ and f is $p\{-a\}$ -integral at $D+1$ consecutive integers, then $v\text{-}p(\tau(f)) \geq -a - 4 \lfloor \log\text{-}p(D+1) \rfloor$ (Lean: *tau-vge-of-values*; the paper's $-v\text{-}p(24)$ term is not needed).

theorem *tau-vge-of-values*:

assumes p : *prime* p **and** dfD : *degree* $f \leq D$

and hv : $\bigwedge i. i \leq D \implies vge\ p\ (poly\ f\ (of\text{-}int\ y + of\text{-}nat\ i))\ (-\ a)$

shows $vge\ p\ (\tau\ f)\ (-\ a - 4 * of\text{-}nat\ (nat\text{-}\log\ p\ (D + 1)))$

proof –

– the bound spreads from $D + 1$ consecutive integers to all integers

have $hall$: $\bigwedge m::int. vge\ p\ (poly\ f\ (of\text{-}int\ m))\ (-\ a)$

proof –

fix $m :: int$

have fdv : $\bigwedge k. k \leq D \implies vge\ p\ (fd\ f\ (of\text{-}int\ y)\ k)\ (-\ a)$

by (*rule fd-vge[OF p]*) (*use hv in simp*)

have $poly\ f\ (of\text{-}int\ m) = (\sum\ k \leq D. fd\ f\ (of\text{-}int\ y)\ k * poly\ (bp\ k)\ (of\text{-}int\ (m - y)))$

proof –

have N : $f = (\sum\ k \leq D. [:fd\ f\ (of\text{-}int\ y)\ k:] * pcompose\ (bp\ k)\ (pX - [:of\text{-}int\ y:])))$

by (*rule newton[OF dfD]*)

have $poly\ f\ (of\text{-}int\ m)$

$= poly\ (\sum\ k \leq D. [:fd\ f\ (of\text{-}int\ y)\ k:] * pcompose\ (bp\ k)\ (pX - [:of\text{-}int\ y:])))\ (of\text{-}int\ m)$

by (*rule arg-cong[where $f = \lambda P. poly\ P\ (of\text{-}int\ m)$, OF N]*)

also have $\dots = (\sum\ k \leq D. fd\ f\ (of\text{-}int\ y)\ k * poly\ (bp\ k)\ (of\text{-}int\ (m - y)))$

by (*simp add: poly-sum poly-pcompose poly-monom*)

finally show *?thesis* .

qed

also have $vge\ p\ \dots\ (-\ a)$

proof (*rule vge-sum[OF p]*)

fix k **assume** $k \in \{..D\}$

obtain $z :: int$ **where** z : $poly\ (bp\ k)\ (of\text{-}int\ (m - y)) = of\text{-}int\ z$ **using** *bp-eval-int* **by** *blast*

have $k \leq D$ **using** $\langle k \in \{..D\} \rangle$ **by** *simp*

from *vge-mul[OF p fdv[OF this] vge-of-int[OF p, of z]]*

show $vge\ p\ (fd\ f\ (of\text{-}int\ y)\ k * poly\ (bp\ k)\ (of\text{-}int\ (m - y)))\ (-\ a)$ **unfolding**

z **by** *simp*

qed *simp*

finally show $vge\ p\ (poly\ f\ (of\text{-}int\ m))\ (-\ a)$.

qed

– Newton at 0, and $\tau(binom(x, k)) = [x^4] binom(x, k+1)$

have $f0$: $f = (\sum\ k \leq D. [:fd\ f\ 0\ k:] * bp\ k)$

```

using newton[OF dfD, of 0] by (simp only: pcompose-pX-minus-0)
have tf: tau f = (∑ k ≤ D. fd f 0 k * Polynomial.coeff (bp (Suc k)) 4)
proof -
  have tau f = tau (∑ k ≤ D. [:fd f 0 k:] * bp k)
    by (rule arg-cong[where f = tau, OF f0])
  also have ... = (∑ k ≤ D. tau ([:fd f 0 k:] * bp k)) by (rule tau-sum)
  also have ... = (∑ k ≤ D. fd f 0 k * Polynomial.coeff (bp (Suc k)) 4)
    by (simp add: tau-smult tau-bp)
  finally show ?thesis .
qed
have trm: ∧k. k ≤ D ⇒
  vge p (fd f 0 k * Polynomial.coeff (bp (Suc k)) 4) (- a - 4 * of-nat (nat-log
p (D + 1)))
proof -
  fix k assume kD: k ≤ D
  have 1: vge p (fd f 0 k) (- a)
    by (rule fd-vge[OF p]) (use hall[of int i for i] in simp)
  have 2: vge p (Polynomial.coeff (bp (Suc k)) 4) (- 4 * of-nat (nat-log p (Suc
k)))
    by (rule bp-coeff4[OF p])
  have nat-log p (Suc k) ≤ nat-log p (D + 1) using kD by (intro nat-log-mono)
  simp
  hence - a - 4 * of-nat (nat-log p (D + 1)) ≤ - a + - 4 * of-nat (nat-log p
(Suc k))
    by simp
  thus vge p (fd f 0 k * Polynomial.coeff (bp (Suc k)) 4) (- a - 4 * of-nat
(nat-log p (D + 1)))
    by (rule vge-mono[OF vge-mul[OF p 1 2]])
qed
show ?thesis unfolding tf by (rule vge-sum[OF p]) (use trm in auto)
qed

```

(3.8): difference quotients of integer-valued polynomials of degree $\leq d$ lose at most $\lfloor \log_p \max(1, d) \rfloor$ (Lean: *diffQuot-vge*).

theorem *diffQuot-vge*:

assumes p: prime p **and** A: IntValuedAt p A **and** dA: degree A ≤ d **and** nr: n ≠ r

shows vge p ((poly A (of-int n) - poly A (of-int r)) / (of-int n - of-int r))
 (- of-nat (nat-log p (max 1 d)))

proof -

define A' **where** A' = pcompose A (pX + [:of-int r:])

have dA': degree A' ≤ d

unfolding A'-def degree-pcompose degree-pX-plus-const **using** dA **by** simp

have valA': ∧m::int. vge p (poly A' (of-int m)) 0

proof -

fix m :: int

have poly A' (of-int m) = poly A (of-int (m + r))

by (simp add: A'-def poly-pcompose poly-monom)

thus vge p (poly A' (of-int m)) 0 **using** IntValuedAt-vge[OF A, of m + r] **by**

```

simp
qed
define u where u = of-int n - (of-int r :: rat)
have u0: u ≠ 0 using nr by (simp add: u-def)
have An: poly A (of-int n) = poly A' u and Ar: poly A (of-int r) = poly A' 0
  by (simp-all add: A'-def u-def poly-pcompose poly-monom)
have N: A' = (∑ k ≤ d. [:fd A' 0 k:] * bp k)
  using newton[OF dA', of 0] by (simp only: pcompose-pX-minus-0)
have split: (poly A' u - poly A' 0) / u =
  (∑ k ≤ d. fd A' 0 k * ((poly (bp k) u - poly (bp k) 0) / u))
proof -
  have h: poly A' x = (∑ k ≤ d. fd A' 0 k * poly (bp k) x) for x
  proof -
    have poly A' x = poly (∑ k ≤ d. [:fd A' 0 k:] * bp k) x
    by (rule arg-cong[where f = λP. poly P x, OF N])
    thus ?thesis by (simp add: poly-sum)
  qed
show ?thesis unfolding h[of u] h[of 0]
  by (simp add: sum-subtractf sum-divide-distrib right-diff-distrib diff-divide-distrib)
qed
have trm: ∧k. k ≤ d ⇒
  vge p (fd A' 0 k * ((poly (bp k) u - poly (bp k) 0) / u)) (- of-nat (nat-log p
(max 1 d)))
proof -
  fix k assume kd: k ≤ d
  show vge p (fd A' 0 k * ((poly (bp k) u - poly (bp k) 0) /
u)) (- of-nat (nat-log p (max 1 d)))
proof (cases k)
  case 0
  thus ?thesis by (simp add: bp-0 vge-zero)
next
  case (Suc j)
  have s: poly (bp (Suc j)) x = 1 / (of-nat j + 1) * (x * poly (bp j) (x - 1))
for x
  by (subst bp-shift) (simp add: poly-pcompose poly-monom)
  have e: (poly (bp (Suc j)) u - poly (bp (Suc j)) 0) / u =
    1 / (of-nat j + 1) * poly (bp j) (u - 1)
  unfolding s using u0 by simp
  obtain z :: int where z: poly (bp j) (u - 1) = of-int z
  using bp-eval-int[of j n - r - 1] by (auto simp: u-def)
  have 1: vge p (fd A' 0 (Suc j)) 0
  by (rule fd-vge[OF p]) (use valA'[of int i for i] in simp)
  have 2: vge p (1 / (of-nat j + 1)) (- of-nat (nat-log p (Suc j)))
  using vge-inv-nat[OF p, of Suc j] by (simp add: add.commute)
  have 3: nat-log p (Suc j) ≤ nat-log p (max 1 d) using kd Suc
  by (intro nat-log-mono) simp
  show ?thesis unfolding Suc e z
  by (rule vge-mono[OF vge-mul[OF p 1 vge-mul[OF p 2 vge-of-int[OF p,
of z]]]] (use 3 in simp)

```

```

    qed
  qed
  have vge p ((poly A' u - poly A' 0) / u) (- of-nat (nat-log p (max 1 d)))
    unfolding split by (rule vge-sum[OF p]) (use trm in auto)
  thus ?thesis unfolding An Ar u-def .
qed

end

```

```

theory Zeta5-Far-Poles
  imports Zeta5-IntValued Legendre-Factorial-Valuation
begin

```

17 The far-pole constant and Lemma 3.3

Section 3 of the paper: the valuations of the Bernoulli numbers (von Staudt–Clausen, from the AFP), the far-pole constant C_p of Section 3.1, counting lemmas for multiples and valuations of integers, partial fractions with linear poles, the product-difference estimate and Lemma 3.3 (*far-pole-valuation-bound*). Lean theories *Section3*, *Lemma33*, *MuMod*.

17.1 Bernoulli valuations (von Staudt–Clausen, from the AFP)

$v_p(\kappa-d) \geq -1$ for odd p (false for $p = 2$: $\kappa-3 = 1/4$). Follows from *Bernoulli-FPS.vonStaudt-Clausen / Bernoulli-FPS.prime-dvd-bernoulli-denom-iff*: the denominator of B_n is squarefree.

```

lemma multiplicity-24:
  assumes prime p p ≠ 2
  shows multiplicity p (24::nat) = (if p = 3 then 1 else 0)
proof -
  have m3: multiplicity p (3::nat) = (if p = 3 then 1 else 0)
    using assms(1) by (force simp: prime-multiplicity-prime)
  have ¬ p dvd (2 ^ 3 :: nat)
  proof
    assume p dvd (2 ^ 3 :: nat)
    hence p dvd 2 by (rule prime-dvd-power[OF assms(1)])
    hence p = 2 using primes-dvd-imp-eq[OF assms(1) two-is-prime-nat] by simp
    thus False using assms(2) by simp
  qed
  hence m8: multiplicity p (2 ^ 3 :: nat) = 0 by (rule not-dvd-imp-multiplicity-0)
  have multiplicity p (3 * 2 ^ 3 :: nat) = multiplicity p 3 + multiplicity p (2 ^ 3)
    using assms(1) by (intro prime-elem-multiplicity-mult-distrib) auto
  thus ?thesis using m3 m8 by simp
qed

```

The denominator of a Bernoulli number is squarefree (von Staudt–Clausen): $v_p(B_n) \geq -1$.

```

lemma vp-bernoulli-ge:
  assumes prime  $p$  (bernoulli  $n :: \text{rat}$ )  $\neq 0$ 
  shows  $-1 \leq \text{vp } p$  (bernoulli  $n$ )
proof -
  have pe: prime-elem (int  $p$ ) using assms(1) by (simp add: prime-nat-int-transfer)
  have bernoulli-num  $n \neq 0$  using assms(2) by (auto simp: bernoulli-conv-num-denom)
  hence  $\text{vp } p$  (bernoulli  $n$ ) = int (multiplicity (int  $p$ ) (bernoulli-num  $n$ )) -
    int (multiplicity  $p$  (bernoulli-denom  $n$ ))
  unfolding vp-def bernoulli-conv-num-denom using pe
    by (simp add: qmultiplicity-divide-of-int multiplicity-int)
  moreover have multiplicity  $p$  (bernoulli-denom  $n$ )  $\leq 1$ 
    using squarefree-bernoulli-denom[of n] assms(1)
    by (intro squarefree-imp-multiplicity-prime-le-1) auto
  ultimately show ?thesis by simp
qed

```

B - n is p -integral unless $(p - 1) \mid n$ (for odd p).

```

lemma vp-bernoulli-nonneg:
  assumes prime  $p$   $p \neq 2 \rightarrow (p - 1) \text{ dvd } n$ 
  shows  $0 \leq \text{vp } p$  (bernoulli  $n$ )
proof (cases (bernoulli  $n :: \text{rat}$ ) = 0)
  case True
    thus ?thesis by (simp add: vp-def)
  next
    case False
    have pe: prime-elem (int  $p$ ) using assms(1) by (simp add: prime-nat-int-transfer)
    have num: bernoulli-num  $n \neq 0$  using False by (auto simp: bernoulli-conv-num-denom)
    have nd:  $\neg p \text{ dvd } \text{bernoulli-denom } n$ 
    proof (cases even  $n \wedge 0 < n$ )
      case True
        thus ?thesis using assms(1,3) by (simp add: prime-dvd-bernoulli-denom-iff)
      next
        case F: False
        hence bernoulli-denom  $n = 1 \vee \text{bernoulli-denom } n = 2$  by (auto simp:
bernoulli-denom-def)
        thus ?thesis using assms(1,2) primes-dvd-imp-eq[OF assms(1) two-is-prime-nat]
by auto
    qed
    have  $\text{vp } p$  (bernoulli  $n$ ) = int (multiplicity (int  $p$ ) (bernoulli-num  $n$ )) -
      int (multiplicity  $p$  (bernoulli-denom  $n$ ))
    unfolding vp-def bernoulli-conv-num-denom using num pe
      by (simp add: qmultiplicity-divide-of-int multiplicity-int)
    thus ?thesis using nd by (simp add: not-dvd-imp-multiplicity-0)
qed

```

```

lemma three-dvd-consec: ( $3 :: \text{int}$ ) dvd  $m * (m - 1) * (m - 2)$ 
proof -
  have  $3 \text{ dvd } m \vee 3 \text{ dvd } (m - 1) \vee 3 \text{ dvd } (m - 2)$  by presburger
  thus ?thesis by (auto intro: dvd-mult dvd-mult2)

```

qed

lemma *kappa-eq*: $\kappa d = \text{of-int } (int\ d * (int\ d - 1) * (int\ d - 2)) * \text{bernoulli } (d - 3) / 24$
by (*simp add: kappa-def*)

lemma *vp-24*: $\text{prime } p \implies p \neq 2 \implies$
 $vp\ p\ 24 = (\text{if } p = 3 \text{ then } 1 \text{ else } 0)$
unfolding *vp-def* **using** *multiplicity-24*[*of p*]
by (*simp add: multiplicity-int*[*of p 24, symmetric*] *qmultiplicity-of-nat*[*of int p 24,*
simplified])

lemma *kappa-padicVal-ge*:
assumes *prime p p ≠ 2 kappa d ≠ 0*
shows $-1 \leq vp\ p\ (\kappa d)$
proof –
have *pe*: *prime-elem* (*int p*) **using** *assms(1)* **by** (*simp add: prime-nat-int-transfer*)
define *m* **where** $m = int\ d * (int\ d - 1) * (int\ d - 2)$
have *m0*: $m \neq 0$ **and** *B0*: $(\text{bernoulli } (d - 3) :: \text{rat}) \neq 0$ **using** *assms(3)*
by (*auto simp: kappa-eq m-def*)
have *eq*: $vp\ p\ (\kappa d) = vp\ p\ (\text{of-int } m) + vp\ p\ (\text{bernoulli } (d - 3)) - vp\ p\ 24$
unfolding *kappa-eq m-def*[*symmetric*] *vp-def* **using** *pe m0 B0*
by (*simp add: qmultiplicity-mult qmultiplicity-divide*)
have $0 \leq vp\ p\ (\text{of-int } m) - vp\ p\ 24$
proof (*cases p = 3*)
case *True*
have *int 3 dvd m* **unfolding** *m-def* **by** (*simp add: three-dvd-consec*)
hence $1 \leq \text{multiplicity } (int\ 3)\ m$ **using** *m0*
by (*force intro: multiplicity-geI*)
hence $1 \leq vp\ p\ (\text{of-int } m)$ **using** *True* **by** (*simp add: vp-def*)
moreover **have** $vp\ p\ 24 = 1$ **using** *vp-24*[*OF assms(1,2)*] *True* **by** *simp*
ultimately show *?thesis* **by** *simp*
next
case *False*
have $vp\ p\ 24 = 0$ **using** *vp-24*[*OF assms(1,2)*] *False* **by** *simp*
moreover **have** $0 \leq vp\ p\ (\text{of-int } m)$ **by** (*simp add: vp-def*)
ultimately show *?thesis* **by** *simp*
qed
moreover **have** $-1 \leq vp\ p\ (\text{bernoulli } (d - 3))$ **by** (*rule vp-bernoulli-ge*[*OF assms(1) B0*])
ultimately show *?thesis* **using** *eq* **by** *linarith*
qed

$\kappa\text{-}d \in \mathbb{Z}\text{-}p$ for $d \leq p + 1$, $p \geq 7$ (in fact for $d \leq 4p - 2$).

lemma *kappa-integral*:
assumes *prime p 7 ≤ p d ≤ p + 1 kappa d ≠ 0*
shows $0 \leq vp\ p\ (\kappa d)$
proof –

```

have pe: prime-elem (int p) using assms(1) by (simp add: prime-nat-int-transfer)
define m where m = int d * (int d - 1) * (int d - 2)
have m0: m ≠ 0 and B0: (bernoulli (d - 3) :: rat) ≠ 0 using assms(4)
  by (auto simp: kappa-eq m-def)
have d3: 3 ≤ d using m0 by (cases d rule: nat.exhaust; cases d - 1; auto simp:
m-def)
have vp p (kappa d) = vp p (of-int m) + vp p (bernoulli (d - 3)) - vp p 24
  unfolding kappa-eq m-def[symmetric] vp-def using pe m0 B0
  by (simp add: qmultiplicity-mult qmultiplicity-divide)
moreover have vp p 24 = 0 using assms(1,2) by (simp add: vp-24)
moreover have 0 ≤ vp p (of-int m) by (simp add: vp-def)
moreover have 0 ≤ vp p (bernoulli (d - 3))
proof (cases d = 3)
  case True
  have (bernoulli 0 :: rat) = 1
  by (rule bernoulli-0)
  thus ?thesis using True by (simp add: vp-def)
next
  case False
  hence ¬ (p - 1) dvd (d - 3) using assms(2,3) d3 dvd-imp-le[of p - 1 d -
3] by auto
  thus ?thesis using assms(1,2) by (intro vp-bernoulli-nonneg) auto
qed
ultimately show ?thesis by linarith
qed

```

The polynomial moments: $v\text{-}p(\mu(t^e)) \geq -1$, and p -integral for $e < 2p - 3$.

lemma *muMono-eq*:

```

muMono e = (-1) ^ e * (bernoulli (2 * e + 2) * of-int ((2 * int e + 3) * (2 *
int e + 4) * (2 *
int e + 5)) / 24)
by (simp add: muMono-def)

```

lemma *muMono-padicVal*:

```

assumes prime p 7 ≤ p muMono e ≠ 0
shows - 1 ≤ vp p (muMono e) ∧ (e < 2 * p - 3 → 0 ≤ vp p (muMono e))
proof -
  have pe: prime-elem (int p) using assms(1) by (simp add: prime-nat-int-transfer)
  define m where m = (2 * int e + 3) * (2 * int e + 4) * (2 * int e + 5)
  have m0: m ≠ 0 unfolding m-def by (auto simp: mult-pos-pos)
  have B0: (bernoulli (2 * e + 2) :: rat) ≠ 0 using assms(3) by (auto simp:
muMono-eq)
  have v24: vp p 24 = 0 using assms(1,2) by (simp add: vp-24)
  have sign: vp p (muMono e) = vp p (bernoulli (2 * e + 2) * of-int m / 24)
  proof (cases even e)
    case True
    thus ?thesis unfolding muMono-eq m-def[symmetric] by (simp add: neg-one-even-power)
  next

```



```

    case False
    thus ?thesis unfolding muMono-eq m-def[symmetric] vp-def by (simp add:
neg-one-odd-power)
qed
have eq: vp p (muMono e) = vp p (bernoulli (2 * e + 2)) + vp p (of-int m)
proof -
  have vp p (bernoulli (2 * e + 2) * of-int m / 24) =
    vp p (bernoulli (2 * e + 2)) + vp p (of-int m) - vp p 24
  unfolding vp-def using pe m0 B0 by (simp add: qmultiplicity-mult qmulti-
plicity-divide)
  thus ?thesis using sign v24 by simp
qed
have vm: 0 ≤ vp p (of-int m) by (simp add: vp-def)
have vB: - 1 ≤ vp p (bernoulli (2 * e + 2)) by (rule vp-bernoulli-ge[OF
assms(1) B0])
have part2: 0 ≤ vp p (muMono e) if e: e < 2 * p - 3
proof (cases (p - 1) dvd (2 * e + 2))
  case False
  hence 0 ≤ vp p (bernoulli (2 * e + 2)) using assms(1,2)
  by (intro vp-bernoulli-nonneg) auto
  thus ?thesis using eq vm by linarith
next
  case True
  then obtain k where k: 2 * e + 2 = (p - 1) * k by (elim dvdE)
  have (p - 1) * k = 2 * e + 2 using k by simp
  also have ... < (p - 1) * 4 using e assms(2) by (simp add: diff-mult-distrib)
  finally have k < 4 using assms(2) mult-less-cancel1 by blast
  moreover have k ≠ 0 using k by (cases k) auto
  ultimately have k = 1 ∨ k = 2 ∨ k = 3 by auto
  moreover have k': 2 * int e + 2 = (int p - 1) * int k
  proof -
    have int (2 * e + 2) = int ((p - 1) * k) using k by simp
    thus ?thesis using assms(2) by (simp add: of-nat-diff)
  qed
  ultimately have int p dvd m
  proof (elim disjE)
    assume k = 1
    hence p: int p = 2 * int e + 3 using k' by simp
    have m = int p * ((2 * int e + 4) * (2 * int e + 5)) unfolding m-def p
    by (simp add: algebra-simps)
    thus ?thesis by simp
  next
    assume k = 2
    hence p: int p = int e + 2 using k' by simp
    have m = int p * (2 * (2 * int e + 3) * (2 * int e + 5)) unfolding m-def p
    by (simp add: algebra-simps)
    thus ?thesis by simp
  next
    assume k = 3

```

```

    hence p: 3 * int p = 2 * int e + 5 using k' by simp
    have m = int p * (3 * ((2 * int e + 3) * (2 * int e + 4))) unfolding m-def
p[symmetric]
    by (simp add: algebra-simps)
    thus ?thesis by simp
qed
    hence 1 ≤ multiplicity (int p) m using m0 pe
    by (metis dvd-imp-multiplicity-le multiplicity-prime)
    hence 1 ≤ vp p (of-int m) by (simp add: vp-def)
    thus ?thesis using eq vB by linarith
qed
    show ?thesis using eq vm vB part2 by linarith
qed

```

The modified moments $\mu(t^e) - c \cdot e/p$ are p -integral (Lean: *muMod-vge*).

```

lemma vge-one-of-qcong-0:
  assumes prime p [z = 0] (qmod int p)
  shows vge p z 1
proof (cases z = 0)
  case True
  thus ?thesis by (simp add: vge-zero)
next
  case False
  have pe: prime-elem (int p) using assms(1) by (simp add: prime-nat-int-transfer)
  obtain a b where ab: quotient-of z = (a, b) by (cases quotient-of z)
  from assms(2) ab have dvd: int p dvd a and cop: coprime b (int p)
    by (auto simp: qcong-0-iff)
  have a0: a ≠ 0 using False quotient-of-div[OF ab] by auto
  have ndvd: ¬ int p dvd b
  proof
    assume d: int p dvd b
    have coprime (int p) b using cop by (simp add: coprime-commute)
    hence is-unit (int p) using coprime-absorb-left[OF d] by simp
    thus False using pe by (simp add: prime-elem-def)
  qed
  have mb: multiplicity (int p) b = 0 by (rule not-dvd-imp-multiplicity-0[OF ndvd])
  have ma: 1 ≤ multiplicity (int p) a
    using dvd a0 pe by (metis dvd-imp-multiplicity-le multiplicity-prime)
  have qmultiplicity (int p) z = int (multiplicity (int p) a) - int (multiplicity (int
p) b)
    by (simp add: qmultiplicity-def ab)
  thus ?thesis using ma mb by (simp add: vge-def vp-def)
qed

```

```

lemma muMod-vge: prime p ⇒ 7 ≤ p ⇒
  vge p (muMono e - cCorr p e / of-nat p) 0
proof -
  assume p: prime p and p7: 7 ≤ p
  have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer)

```

```

have p0: p ≠ 0 and p1: 1 < int p using prime-gt-1-nat[OF p] by auto
show vge p (muMono e - cCorr p e / of-nat p) 0
proof (cases muMono e = 0)
  case True
  thus ?thesis by (simp add: cCorr-def vge-zero)
next
  case mu0: False
  note mv = muMono-padicVal[OF p p7 mu0]
  show ?thesis
  proof (cases e < 2 * p - 3)
    case True
    thus ?thesis using mv by (simp add: cCorr-def vge-def)
  next
    case False
    define y where y = of-nat p * muMono e
    define r where r = y qmod int p
    have cc: cCorr p e = of-int r using False by (simp add: cCorr-def r-def
y-def)
    have vp p y = vp p (of-nat p) + vp p (muMono e)
    unfolding y-def vp-def using pe p0 mu0 by (simp add: qmultiplicity-mult)
    also have vp p (of-nat p) = 1
    using pe by (simp add: vp-def qmultiplicity-of-nat multiplicity-prime)
    finally have vy: 0 ≤ qmultiplicity (int p) y using mv by (simp add: vp-def)
    have coprime (snd (quotient-of y)) (int p)
    by (rule qmultiplicity-prime-nonneg-imp-coprime-denom[OF vy]) (use p in
simp add:
prime-nat-int-transfer)
    hence qc: [y = of-int r] (qmod int p)
    unfolding qcong-def using p1 by (simp add: qmod-of-int r-def qmod-mod-absorb)
    have qr: [of-int r = (of-int r :: rat)] (qmod int p)
    by (rule qcong-refl) simp
    have [y - of-int r = of-int r - of-int r] (qmod int p)
    by (rule qcong-diff[OF qc qr]) (use p1 in simp)
    hence [y - of-int r = 0] (qmod int p) by simp
    hence vz: vge p (y - of-int r) 1 by (rule vge-one-of-qcong-0[OF p])
    have eq: muMono e - cCorr p e / of-nat p = (y - of-int r) * (1 / of-nat p)
    unfolding cc y-def using p0 by (simp add: field-simps)
    show ?thesis
    unfolding eq using vge-mul[OF p vz vge-inv-p[OF p]] by simp
  qed
qed
qed

```

17.2 Section 3.1 The far-pole constant C_p

Section 3.2: $C_p \in p^5 \mathbb{Z}_p$; every partial sum already has valuation ≥ 5 (the $k = 0$ term uses $\sum_{a=1}^{p-1} a^{-4} \equiv 0 \pmod{p}$, Lean: `pt-invPowSum-vge`).

lemma *vge-of-nat*: $\text{prime } p \implies \text{vge } p \text{ (of-nat } n) \ 0$

```

using vge-of-int[of p int n] by simp

lemma vge-p-self: prime p  $\implies$  vge p (of-nat p) 1
proof -
  assume prime p
  hence prime-elem (int p) by (simp add: prime-nat-int-transfer)
  thus ?thesis by (simp add: vge-def vp-def qmultiplicity-of-nat multiplicity-prime)
qed

lemma vge-dvd-nat: prime p  $\implies$  p dvd n  $\implies$  vge p (of-nat n) 1
proof -
  assume p: prime p and p dvd n
  then obtain t where t: n = p * t by (elim dvdE)
  have vge p (of-nat p * of-nat t) (1 + 0)
    by (rule vge-mul[OF p vge-p-self[OF p] vge-of-nat[OF p]])
  thus ?thesis by (simp add: t)
qed

lemma vge-inverse-unit: prime p  $\implies$   $\neg$  p dvd a  $\implies$ 
  vge p (inverse (of-nat a)) 0
proof -
  assume prime p and a:  $\neg$  p dvd a
  have  $\neg$  int p dvd int a using a by simp
  hence multiplicity (int p) (int a) = 0 by (rule not-dvd-imp-multiplicity-0)
  thus ?thesis by (simp add: vge-def vp-def qmultiplicity-inverse qmultiplicity-of-nat)
qed

lemma vge-pow4-diff:
  assumes prime p vge p x 0 vge p y 0 vge p (x - y) 1
  shows vge p (x4 - y4) 1
proof -
  have e: x4 - y4 = (x - y) * (x3 + x2 * y + x * y2 + y3)
    by (simp add: algebra-simps power-numeral-reduce)
  have m0:  $\bigwedge u v. \text{vge } p \ u \ 0 \implies \text{vge } p \ v \ 0 \implies \text{vge } p \ (u * v) \ 0$ 
    using vge-mul[OF assms(1), where a = 0 and b = 0] by simp
  have c: vge p (x3 + x2 * y + x * y2 + y3) 0
    using assms(2,3)
    by (intro vge-add[OF assms(1)] m0 vge-pow[OF assms(1)])
  have vge p ((x - y) * (x3 + x2 * y + x * y2 + y3)) (1 + 0)
    by (rule vge-mul[OF assms(1) assms(4) c])
  thus ?thesis unfolding e by simp
qed

lemma sum-pow4-mod-prime:
  fixes p :: nat
  assumes p: prime p 7  $\leq$  p
  shows p dvd ( $\sum c = 1..p - 1. c^4$ )
proof -
  define A where A = {1..p - 1}

```

```

define f where f c = 2 * c mod p for c
have p0: 0 < p using p by simp
have nd2: ¬ p dvd 2 using p(2) by (auto dest: dvd-imp-le)
have cop2: coprime 2 p by (rule coprime-commute[THEN iffD1, OF prime-imp-coprime[OF
p(1) nd2]])
have fA: f ' A ⊆ A
proof
  fix y assume y ∈ f ' A
  then obtain c where c: c ∈ A y = f c by auto
  have ¬ p dvd 2 * c
  proof
    assume p dvd 2 * c
    hence p dvd c using cop2 by (simp add: coprime-dvd-mult-right-iff co-
prime-commute)
    moreover have 0 < c < p using c(1) p0 by (auto simp: A-def)
    ultimately show False by (simp add: nat-dvd-not-less)
  qed
  hence y ≠ 0 using c(2) by (simp add: f-def mod-eq-0-iff-dvd)
  moreover have y < p using c(2) p0 by (simp add: f-def)
  ultimately show y ∈ A by (auto simp: A-def)
qed
have inj: inj-on f A
proof (rule inj-onI)
  fix x y assume xy: x ∈ A y ∈ A f x = f y
  hence [2 * x = 2 * y] (mod p) by (simp add: f-def cong-def)
  hence [x = y] (mod p) using cop2 by (simp add: cong-mult-lcancel-nat)
  thus x = y using xy(1,2) by (auto simp: A-def cong-def)
qed
have fAA: f ' A = A by (rule endo-inj-surj[OF - fA inj]) (simp add: A-def)
define S where S = (∑ c ∈ A. c ^ 4)
have S = (∑ c ∈ f ' A. c ^ 4) unfolding fAA S-def ..
also have ... = (∑ c ∈ A. f c ^ 4) by (rule sum.reindex[OF inj, unfolded
o-def])
finally have S1: S = (∑ c ∈ A. f c ^ 4) .
have [(∑ c ∈ A. f c ^ 4) = (∑ c ∈ A. (2 * c) ^ 4)] (mod p)
  by (rule cong-sum, rule cong-pow) (simp add: f-def cong-def)
also have (∑ c ∈ A. (2 * c) ^ 4) = 16 * S
  by (simp add: S-def sum-distrib-left power-mult-distrib)
finally have [16 * S = S] (mod p) using S1 by (simp add: cong-sym-eq)
hence p dvd 16 * S - S by (subst cong-altdef-nat[symmetric]) simp-all
hence p dvd 15 * S by simp
hence p dvd 15 ∨ p dvd S using p(1) by (simp add: prime-dvd-mult-iff)
moreover have ¬ p dvd 15
proof
  assume p dvd 15
  hence p dvd 3 * 5 by simp
  hence p dvd 3 ∨ p dvd 5 using p(1) by (simp only: prime-dvd-mult-iff)
  thus False using p(2) by (auto dest: dvd-imp-le)
qed

```

```

ultimately show ?thesis by (simp add: S-def A-def)
qed

lemma invPowSum-vge: prime p  $\implies$   $7 \leq p \implies$ 
  vge p ( $\sum a = 1..p - 1$ . inverse (of-nat a)  $^{\wedge} 4$ ) 1
proof -
  assume p: prime p and p7:  $7 \leq p$ 
  define A where A = {1..p - 1}
  have p0:  $0 < p$  using p7 by simp
  have ex:  $\exists b. b \in A \wedge [a * b = 1] \pmod{p}$  if a:  $a \in A$  for a
  proof -
    have coprime p 1 by simp
    moreover have  $0 < a < p$  using a p0 by (auto simp: A-def)
    ultimately obtain y where y:  $0 < y \wedge y < p \wedge [a * y = 1] \pmod{p}$ 
      using cong-solve-unique-nontrivial[OF p] by blast
    show ?thesis using y by (intro exI[of - y]) (auto simp: A-def)
  qed
  define bf where bf a = (SOME b.  $b \in A \wedge [a * b = 1] \pmod{p}$ ) for a
  have bf:  $bf\ a \in A \wedge [a * bf\ a = 1] \pmod{p}$  if a:  $a \in A$  for a
    using someI-ex[OF ex[OF that]] unfolding bf-def by auto
  have ndvd:  $\neg p \text{ dvd } a$  if a:  $a \in A$  for a
    using that p0 by (auto simp: A-def dest: dvd-imp-le)
  have inj: inj-on bf A
  proof (rule inj-onI)
    fix x y assume xy:  $x \in A \wedge y \in A \wedge bf\ x = bf\ y$ 
    have  $[x * bf\ x = y * bf\ x] \pmod{p}$  using bf(2)[OF xy(1)] bf(2)[OF xy(2)]
      xy(3)
      by (metis cong-sym cong-trans)
    moreover have coprime (bf x) p using ndvd[OF bf(1)[OF xy(1)]] p
      by (simp add: prime-imp-coprime coprime-commute)
    ultimately have  $[x = y] \pmod{p}$  by (simp add: cong-mult-rcancel-nat)
    thus  $x = y$  using xy(1,2) by (auto simp: A-def cong-def)
  qed
  have bAA:  $bf\ ` A = A$  by (rule endo-inj-surj) (use bf(1) inj in <auto simp:
A-def>)
  have diff: vge p (inverse (of-nat a)  $^{\wedge} 4$  - of-nat (bf a)  $^{\wedge} 4$ ) 1 if a:  $a \in A$  for a
  proof -
    have a0:  $a \neq 0$  using a by (auto simp: A-def)
    have abf-pos:  $1 \leq a * bf\ a$  using a0 bf(1)[OF a] by (auto simp: A-def)
    have p dvd a * bf a - 1 using bf(2)[OF a] abf-pos by (simp add: cong-altdef-nat)
    hence v1: vge p ( $- of-nat (a * bf\ a - 1)$ ) 1 by (intro vge-neg vge-dvd-nat[OF
p])
    have vge p ( $- of-nat (a * bf\ a - 1) * inverse (of-nat a)$ ) (1 + 0)
      by (rule vge-mul[OF p v1 vge-inverse-unit[OF p ndvd[OF a]]])
    moreover have  $- of-nat (a * bf\ a - 1) * inverse (of-nat a) =$ 
      inverse (of-nat a) - (of-nat (bf a) :: rat)
      using a0 abf-pos by (simp add: of-nat-diff field-simps)
    ultimately have vge p (inverse (of-nat a) - of-nat (bf a)) 1 by simp
    thus ?thesis

```

by (intro vge-pow4-diff[OF p vge-inverse-unit[OF p ndvd[OF a]] vge-of-nat[OF p]])
 qed
 have $(\sum a \in A. \text{of-nat } (bf \ a) \wedge 4 :: \text{rat}) = (\sum c \in bf \ 'A. \text{of-nat } c \wedge 4)$
 by (rule sum.reindex[OF inj, unfolded o-def, symmetric])
 also have $\dots = \text{of-nat } (\sum c \in A. c \wedge 4)$ unfolding bAA by simp
 finally have sb: $(\sum a \in A. \text{of-nat } (bf \ a) \wedge 4 :: \text{rat}) = \text{of-nat } (\sum c \in A. c \wedge 4)$.
 have v2: vge p $(\sum a \in A. \text{of-nat } (bf \ a) \wedge 4 :: \text{rat})$ 1
 unfolding sb unfolding A-def by (rule vge-dvd-nat[OF p sum-pow4-mod-prime[OF p p7]])
 have v1: vge p $(\sum a \in A. \text{inverse } (\text{of-nat } a) \wedge 4 - \text{of-nat } (bf \ a) \wedge 4)$ 1
 by (rule vge-sum[OF p]) (use diff in ⟨simp-all add: A-def⟩)
 have vge p $((\sum a \in A. \text{inverse } (\text{of-nat } a) \wedge 4 - \text{of-nat } (bf \ a) \wedge 4) + (\sum a \in A. \text{of-nat } (bf \ a) \wedge 4))$ 1
 by (rule vge-add[OF p v1 v2])
 thus ?thesis by (simp add: sum-subtractf A-def)
 qed

 lemma vge-p-pow: prime p $\implies$ vge p $(\text{of-nat } p \wedge n)$ $(\text{of-nat } n)$
 proof -
 assume p: prime p
 have qmultiplicity (int p) $(\text{of-nat } p \text{ powi int } n) = \text{int } n$ by (rule qmultiplicity-power-int[OF p])
 thus ?thesis by (simp add: vge-def vp-def power-int-of-nat)
 qed

 lemma vge-small-unit:
 assumes prime p $7 \leq p$ $0 < c$ $c < 7$
 shows vge p $(\text{inverse } (\text{of-nat } c))$ 0
 by (rule vge-inverse-unit[OF assms(1)]) (use assms in ⟨auto dest: dvd-imp-le⟩)

 lemma powi-neg-frac:
 assumes $a \neq 0$ $p \neq 0$
 shows $(- \text{of-nat } a / \text{of-nat } p :: \text{rat}) \text{ powi } (- (\text{int } k + 4))$
 $= (- 1) \wedge (k + 4) * \text{of-nat } p \wedge (k + 4) * \text{inverse } (\text{of-nat } a) \wedge (k + 4)$
 proof -
 have $- (\text{int } k + 4) = - \text{int } (k + 4)$ by simp
 hence $(- \text{of-nat } a / \text{of-nat } p :: \text{rat}) \text{ powi } (- (\text{int } k + 4)) =$
 $\text{inverse } ((- \text{of-nat } a / \text{of-nat } p) \wedge (k + 4))$
 by (simp only: power-int-minus power-int-of-nat)
 also have $\dots = (- (\text{of-nat } p / \text{of-nat } a)) \wedge (k + 4)$
 by (simp add: power-inverse[symmetric])
 also have $\dots = (- 1) \wedge (k + 4) * \text{of-nat } p \wedge (k + 4) * \text{inverse } (\text{of-nat } a) \wedge (k + 4)$
 by (simp add: power-minus' power-divide divide-inverse power-mult-distrib power-inverse)
 finally show ?thesis .
 qed

```

lemma Cp-tr-vge: prime p  $\implies$   $\gamma \leq p \implies$  vge p (Cp-tr p M) 5
proof -
  assume p: prime p and p7:  $\gamma \leq p$ 
  define A where A = {1..p - 1}
  have p0: p  $\neq$  0 using p7 by simp
  have ndvd:  $\neg$  p dvd a if a  $\in$  A for a
    using that p7 by (auto simp: A-def dest: dvd-imp-le)
  define I where I n = ( $\sum$  a  $\in$  A. inverse (of-nat a :: rat)  $^{\wedge}$  n) for n
  have I0: vge p (I n) 0 for n
    unfolding I-def
  proof (rule vge-sum[OF p])
    fix a assume a  $\in$  A
    thus vge p (inverse (of-nat a)  $^{\wedge}$  n) 0 by (intro vge-pow[OF p] vge-inverse-unit[OF
p] ndvd)
  qed (simp add: A-def)
  have I4: vge p (I 4) 1 unfolding I-def A-def by (rule invPowSum-vge[OF p
p7])
  have u2: vge p (inverse (of-nat 2)) 0 by (rule vge-small-unit[OF p p7]) simp-all
  have u4: vge p (inverse (of-nat 4)) 0 by (rule vge-small-unit[OF p p7]) simp-all
  have mul:  $\bigwedge$  x y b c. vge p x b  $\implies$  vge p y c  $\implies$ 
    vge p (x * y) (b + c) by (rule vge-mul[OF p])
  have sgn: vge p ((- 1)  $^{\wedge}$  n) 0 for n by (rule vge-pow[OF p]) (rule vge-neg[OF
vge-one])
  define T where T k = of-nat ((k + 3) choose 3) * bernoulli k * ((- 1)  $^{\wedge}$  (k +
4) * of-nat p  $^{\wedge}$  (k +
4) * I (k + 4)) for k
  have swap: Cp-tr p M = - (1 / 4) * ( $\sum$  k < M. T k)
  proof -
    have Cp-tr p M = ( $\sum$  a  $\in$  A. - (1 / 4) * ( $\sum$  k <
      M. of-nat ((k + 3) choose 3) * bernoulli k *
        ((- 1)  $^{\wedge}$  (k + 4) * of-nat p  $^{\wedge}$  (k + 4) * inverse (of-nat a)  $^{\wedge}$ 
(k + 4))))
    unfolding Cp-tr-def tauAnPole-tr-def A-def
  proof (rule sum.cong[OF refl])
    fix a assume a: a  $\in$  {1..p - 1}
    have a0': a  $\neq$  0 using a by simp
    show - (1 / 4) * ( $\sum$  k < M. of-nat ((k + 3) choose 3) * (bernoulli k ::
rat) *
      (- of-nat a / of-nat p) powi (- (int k + 4)))
      = - (1 / 4) * ( $\sum$  k < M. of-nat ((k + 3) choose 3) * bernoulli k *
        ((- 1)  $^{\wedge}$  (k + 4) * of-nat p  $^{\wedge}$  (k + 4) * inverse (of-nat a)  $^{\wedge}$ 
(k + 4)))
      by (simp only: powi-neg-frac[OF a0' p0])
    qed
    also have ... = - (1 / 4) * ( $\sum$  k < M. T k)
    unfolding T-def I-def
    by (simp add: sum-distrib-left sum-distrib-right sum.swap[of - A] mult-ac)
    finally show ?thesis .
  
```



```

qed
have Tk: vge p (T k) 5 for k
proof -
  have C: vge p (of-nat ((k + 3) choose 3)) 0 by (rule vge-of-nat[OF p])
  have Pn: vge p (of-nat p ^ (k + 4)) (of-nat (k + 4)) by (rule vge-p-pow[OF
p])
  consider k = 0 | k = 1 | 2 ≤ k by arith
  thus ?thesis
  proof cases
    case 1
    have Ik: vge p (I (k + 4)) 1 using I4 1 by simp
    have B0: vge p (bernoulli k) 0 using 1 by (simp add: bernoulli-0 vge-one)
    have vge p (T k) (0 + 0 + (0 + of-nat (k + 4) + 1))
      unfolding T-def by (intro mul C B0 sgn Pn Ik)
    thus ?thesis using 1 by simp
  next
    case 2
    have be: bernoulli k = - inverse (of-nat 2) unfolding 2 bernoulli-1 by simp
    have B: vge p (bernoulli k) 0 unfolding be by (rule vge-neg[OF u2])
    have vge p (T k) (0 + 0 + (0 + of-nat (k + 4) + 0))
      unfolding T-def by (intro mul C B sgn Pn I0)
    thus ?thesis using 2 by simp
  next
    case 3
    have B: vge p (bernoulli k) (- 1)
      unfolding vge-def
    proof
      assume (bernoulli k :: rat) ≠ 0
      hence - 1 ≤ vp p (bernoulli k) by (rule vp-bernoulli-ge[OF p])
      thus - 1 ≤ (of-int (vp p (bernoulli k)) :: rat)
        by (metis of-int-le-iff of-int-minus of-int-1)
    qed
    have v: vge p (T k) (0 + - 1 + (0 + of-nat (k + 4) + 0))
      unfolding T-def by (intro mul C B sgn Pn I0)
    show ?thesis by (rule vge-mono[OF v]) (use 3 in simp)
  qed
qed
have vge p (- (1 / 4) * (∑ k < M. T k)) (0 + 5)
proof (rule mul)
  have - (1 / 4) = - inverse (of-nat 4 :: rat) by simp
  thus vge p (- (1 / 4)) 0 using vge-neg[OF u4] by (simp only:)
  show vge p (∑ k < M. T k) 5 by (rule vge-sum[OF p]) (simp-all add: Tk)
qed
thus ?thesis unfolding swap by simp
qed

```

17.3 Section 3.3 Lemma 3.3 and the small-prime bound

Lemma 3.3, (3.10): if A is integer-valued with $\deg A \leq d$ and $g = (K!)^{\wedge 2} A(x) / \prod_{0 < |r| \leq K} (x - r)$, then $v\text{-}p \wedge G(\tau\text{-}X(g)) \geq -6 \lfloor \log\text{-}p \max(2K, d+1) \rfloor - v\text{-}p(24)$. Proof: the residues are $\pm (K!)^{\wedge 2} r A(r) / ((K+r)!(K-r)!)$, root counting modulo $p \wedge j$, two-case estimate.

lemma *vge-diff*: $\text{prime } p \implies v\text{ge } p \ x \ b \implies v\text{ge } p \ y \ b \implies$
 $v\text{ge } p \ (x - y) \ b$
using *vge-add*[*of* $p \ x \ b - y$] *vge-neg*[*of* $p \ y \ b$] **by** *simp*

17.4 Counting multiples in an interval

At most $2K \text{ div } q + 1$ elements of $[-K, K]$ lie in one residue class modulo q .

lemma *card-filter-dvd-Icc-le*:

assumes $0 < q$

shows $\text{card } \{s \in \{-\text{int } K .. \text{int } K\}. \text{int } q \text{ dvd } (x - s)\} \leq 2 * K \text{ div } q + 1$

proof –

let $?C = \{s \in \{-\text{int } K .. \text{int } K\}. \text{int } q \text{ dvd } (x - s)\}$

let $?f = \lambda s. (s + \text{int } K) \text{ div int } q$

have *inj*: *inj-on* $?f \ ?C$

proof (*rule inj-onI*)

fix $s \ t$ **assume** $s: s \in ?C$ **and** $t: t \in ?C$ **and** $e: ?f \ s = ?f \ t$

have $\text{int } q \text{ dvd } (x - t) - (x - s)$ **by** (*rule dvd-diff*) (*use* $s \ t$ **in** *simp-all*)

hence $\text{int } q \text{ dvd } (s + \text{int } K) - (t + \text{int } K)$ **by** (*simp add: algebra-simps*)

hence $m: (s + \text{int } K) \bmod \text{int } q = (t + \text{int } K) \bmod \text{int } q$ **by** (*simp only: mod-eq-dvd-iff*)

have $s + \text{int } K = (t + \text{int } K)$

by (*metis e m div-mult-mod-eq*)

thus $s = t$ **by** *simp*

qed

have *sub*: $?f \text{ ' } ?C \subseteq \{0 .. \text{int } (2 * K \text{ div } q)\}$

proof

fix z **assume** $z \in ?f \text{ ' } ?C$

then obtain s **where** $s: s \in ?C$ **and** $z: z = ?f \ s$ **by** *blast*

have $0 \leq z$ **using** s *assms* **unfolding** z **by** (*simp add: pos-imp-zdiv-nonneg-iff*)

moreover have $z \leq \text{int } (2 * K) \text{ div int } q$ **unfolding** z

by (*rule zdiv-mono1*) (*use* s *assms* **in** *auto*)

ultimately show $z \in \{0 .. \text{int } (2 * K \text{ div } q)\}$ **by** (*simp add: zdiv-int*)

qed

have $\text{card } ?C = \text{card } (?f \text{ ' } ?C)$ **by** (*rule card-image[OF inj, symmetric]*)

also have $\dots \leq \text{card } \{0 .. \text{int } (2 * K \text{ div } q)\}$ **by** (*rule card-mono[OF - sub]*) *simp*

also have $\dots = 2 * K \text{ div } q + 1$ **by** *simp*

finally show *thesis* .

qed

lemma *two-mul-div-le-nat*: $0 < q \implies 2 * K \text{ div } q \leq 2 * (K \text{ div } q) + (1 :: \text{nat})$

proof –

```

assume  $q: 0 < q$ 
have  $K: K = K \text{ div } q * q + K \text{ mod } q$  by simp
have  $K \text{ mod } q < q$  using  $q$  by simp
hence  $2 * K < 2 * (K \text{ div } q * q) + 2 * q$  using  $K$  by linarith
hence  $2 * K < (2 * (K \text{ div } q) + 2) * q$  by (simp add: algebra-simps)
hence  $2 * K \text{ div } q < 2 * (K \text{ div } q) + 2$  using div-less-iff-less-mult[ $OF\ q$ ] by
blast
thus ?thesis by simp
qed
lemma card-filter-dvd-le:
 $0 < q \implies T \subseteq \{- \text{int } K .. \text{int } K\} \implies$ 
 $\text{card } \{s \in T. \text{int } q \text{ dvd } (x - s)\} \leq 2 * K \text{ div } q + 1$ 
proof -
assume  $q: 0 < q$  and  $T: T \subseteq \{- \text{int } K .. \text{int } K\}$ 
have  $fC: \text{finite } \{s \in \{- \text{int } K .. \text{int } K\}. \text{int } q \text{ dvd } (x - s)\}$ 
by (rule finite-subset[of -  $\{- \text{int } K .. \text{int } K\}$ ]) auto
have  $\text{card } \{s \in T. \text{int } q \text{ dvd } (x - s)\} \leq \text{card } \{s \in$ 
 $\{- \text{int } K .. \text{int } K\}. \text{int } q \text{ dvd } (x - s)\}$ 
by (rule card-mono[ $OF\ fC$ ]) (use  $T$  in auto)
also have  $\dots \leq 2 * K \text{ div } q + 1$  by (rule card-filter-dvd-Icc-le[ $OF\ q$ ])
finally show ?thesis .
qed

```

If $x0 \in [-K, K] - T$ lies in the residue class, the count drops by one.

```

lemma card-filter-dvd-le-excl:
 $0 < q \implies T \subseteq \{- \text{int } K .. \text{int } K\} \implies$ 
 $x0 \in \{- \text{int } K .. \text{int } K\} \implies x0 \notin T \implies$ 
 $\text{int } q \text{ dvd } (x - x0) \implies$ 
 $\text{card } \{s \in T. \text{int } q \text{ dvd } (x - s)\} \leq 2 * K \text{ div } q$ 
proof -
assume  $q: 0 < q$  and  $T: T \subseteq \{- \text{int } K .. \text{int } K\}$  and  $x0: x0 \in$ 
 $\{- \text{int } K .. \text{int } K\}$   $x0 \notin T$ 
and  $d: \text{int } q \text{ dvd } (x - x0)$ 
have  $fin: \text{finite } \{s \in T. \text{int } q \text{ dvd } (x - s)\}$ 
by (rule finite-subset[of -  $\{- \text{int } K .. \text{int } K\}$ ]) (use  $T$  in auto)
have  $fC: \text{finite } \{s \in \{- \text{int } K .. \text{int } K\}. \text{int } q \text{ dvd } (x - s)\}$ 
by (rule finite-subset[of -  $\{- \text{int } K .. \text{int } K\}$ ]) auto
have  $\text{card } (\text{insert } x0 \{s \in T. \text{int } q \text{ dvd } (x - s)\}) = \text{Suc } (\text{card } \{s \in T. \text{int } q \text{ dvd}$ 
 $(x - s)\})$ 
using fin  $x0(2)$  by simp
moreover have  $\text{card } (\text{insert } x0 \{s \in T. \text{int } q \text{ dvd } (x - s)\}) \leq \text{card } \{s \in$ 
 $\{- \text{int } K .. \text{int } K\}. \text{int } q \text{ dvd } (x - s)\}$ 
by (rule card-mono[ $OF\ fC$ ]) (use  $T\ x0\ d$  in auto)
moreover have  $\text{card } \{s \in \{- \text{int } K .. \text{int } K\}. \text{int } q \text{ dvd } (x - s)\} \leq 2 * K \text{ div } q$ 
 $+ 1$ 
by (rule card-filter-dvd-Icc-le[ $OF\ q$ ])
ultimately show ?thesis by linarith
qed

```

lemma *lt-pow-nat-log-succ*: $\text{prime } p \implies 0 < n \implies n < p \wedge (\text{nat-log } p \ n + 1)$

proof –

assume p : *prime* p **and** n : $0 < n$

have b : $1 < p$ **using** p **by** (*rule prime-gt-1-nat*)

have $\text{floorlog } p \ n \neq 0$ **using** $b \ n$ **by** (*simp add: floorlog-eq-zero-iff*)

hence $\text{nat-log } p \ n + 1 = \text{floorlog } p \ n$ **by** (*simp add: nat-log-def*)

thus *?thesis* **using** *floorlog-bounds*[*OF* $n \ b$] **by** *simp*

qed

lemma *div-eq-zero-of-gt-nat-log*: $\text{prime } p \implies \text{nat-log } p \ (2 * K) < j \implies 2 * K \text{ div } p \wedge j = 0$

proof –

assume p : *prime* p **and** j : $\text{nat-log } p \ (2 * K) < j$

show *?thesis*

proof (*cases* $K = 0$)

case *False*

have $2 * K < p \wedge (\text{nat-log } p \ (2 * K) + 1)$ **using** *False* **by** (*intro lt-pow-nat-log-succ*[*OF* p]) *simp*

also have $\dots \leq p \wedge j$ **using** j *prime-gt-1-nat*[*OF* p]

by (*intro power-increasing*) *simp-all*

finally show *?thesis* **by** *simp*

qed *simp*

qed

17.5 Valuations of integers as counts

lemma *multiplicity-int-eq-card*:

assumes *prime* p $z \neq 0$ *multiplicity* ($\text{int } p$) $z < B$

shows *multiplicity* ($\text{int } p$) $z = \text{card } \{j \in \{1..<B\}. \text{int } p \wedge j \text{ dvd } z\}$

proof –

have *prime* ($\text{int } p$) **using** *assms*(1) **by** (*simp add: prime-nat-int-transfer*)

hence *pu*: $\neg \text{is-unit } (\text{int } p)$ **by** (*metis not-prime-unit*)

have $\{j \in \{1..<B\}. \text{int } p \wedge j \text{ dvd } z\} = \{1..\text{multiplicity } (\text{int } p) \ z\}$

using *assms*(3) **by** (*auto simp: power-dvd-iff-le-multiplicity*[*OF* *assms*(2) *pu*])

thus *?thesis* **by** *simp*

qed

lemma *sum-multiplicity-le*:

assumes *prime* p *finite* T $\bigwedge s. s \in T \implies x - s \neq 0$

and $\bigwedge s. s \in T \implies \text{multiplicity } (\text{int } p) \ (x - s) < B$

and $\bigwedge j. j \in \{1..<B\} \implies \text{card } \{s \in T. \text{int } p \wedge j \text{ dvd } (x - s)\} \leq c \ j$

shows $(\sum s \in T. \text{multiplicity } (\text{int } p) \ (x - s)) \leq (\sum j \in \{1..<B\}. c \ j)$

proof –

have *cs*: $\text{card } \{y \in A. Q \ y\} = (\sum y \in A. \text{if } Q \ y \text{ then } 1 \text{ else } 0)$ **if** *finite* A

for $A :: 'b \text{ set}$ **and** Q

by (*simp only: card-eq-sum sum.inter-filter*[*OF* *that*])

have $(\sum s \in T. \text{multiplicity } (\text{int } p) \ (x - s))$

$= (\sum s \in T. \sum j \in \{1..<B\}. \text{if } \text{int } p \wedge j \text{ dvd } (x - s) \text{ then } 1 \text{ else } 0)$
proof (rule *sum.cong*[*OF refl*])
fix s **assume** $s: s \in T$
have $\text{multiplicity } (\text{int } p) (x - s) = \text{card } \{j \in \{1..<B\}. \text{int } p \wedge j \text{ dvd } (x - s)\}$
by (rule *multiplicity-int-eq-card*[*OF assms(1) assms(3)*][*OF s*] *assms(4)*][*OF s*])
also have $\dots = (\sum j \in \{1..<B\}. \text{if } \text{int } p \wedge j \text{ dvd } (x - s) \text{ then } 1 \text{ else } 0)$
by (rule *cs*) *simp*
finally show $\text{multiplicity } (\text{int } p) (x - s) = (\sum j \in \{1..<B\}. \text{if } \text{int } p \wedge j \text{ dvd } (x - s) \text{ then } 1 \text{ else } 0) .$
qed
also have $\dots = (\sum j \in \{1..<B\}. \sum s \in T. \text{if } \text{int } p \wedge j \text{ dvd } (x - s) \text{ then } 1 \text{ else } 0)$ **by** (rule *sum.swap*)
also have $\dots = (\sum j \in \{1..<B\}. \text{card } \{s \in T. \text{int } p \wedge j \text{ dvd } (x - s)\})$
by (rule *sum.cong*[*OF refl*]) (rule *cs*[*OF assms(2)*, *symmetric*])
also have $\dots \leq (\sum j \in \{1..<B\}. c\ j)$ **by** (rule *sum-mono*) (rule *assms(5)*)
finally show *?thesis* .
qed

The factorial ratio $(K!)^2 / \prod_{s \in T} (x - s)$, from per-level counts: if at each level j at most $2 \lfloor K/p^j \rfloor + e$ [$j \leq L0$] of the $x - s$ are divisible by p^j , the ratio has $v_p \geq -e L0$, where $L0 = \lfloor \log_p 2K \rfloor$.

lemma *vge-factorial-sq-div-prod*:

assumes p : prime p **and** T : finite T **and** $x: \bigwedge s. s \in T \implies x - s \neq 0$
and $hc: \bigwedge j. 1 \leq j \implies \text{card } \{s \in T. \text{int } p \wedge j \text{ dvd } (x - s)\} \leq 2 * (K \text{ div } p^j) + (\text{if } j \leq \text{nat-log } p (2 * K) \text{ then } e \text{ else } 0)$
shows $vge\ p (\text{fact } K^2 / (\prod_{s \in T} (x - s))) (-\ \text{of-nat } (e * \text{nat-log } p (2 * K)))$
proof –
define $L0$ **where** $L0 = \text{nat-log } p (2 * K)$
define B **where** $B = (\sum s \in T. \text{multiplicity } (\text{int } p) (x - s)) + L0 + K + 1$
have $hB: \bigwedge s. s \in T \implies \text{multiplicity } (\text{int } p) (x - s) < B$
proof –
fix s **assume** $s: s \in T$
have $\text{multiplicity } (\text{int } p) (x - s) \leq (\sum s \in T. \text{multiplicity } (\text{int } p) (x - s))$
by (rule *member-le-sum*[*OF s - T*]) *simp*
thus $\text{multiplicity } (\text{int } p) (x - s) < B$ **by** (*simp add: B-def*)
qed
have $hsum: (\sum s \in T. \text{multiplicity } (\text{int } p) (x - s)) \leq 2 * \text{multiplicity } p (\text{fact } K) + L0 * e$
proof –
define c **where** $c\ j = 2 * (K \text{ div } p^j) + (\text{if } j \leq L0 \text{ then } e \text{ else } 0)$ **for** j
have $B1: 1 \leq B$ **by** (*simp add: B-def*)
have $leg: (\sum j \in \{1..<B\}. K \text{ div } p^j) = \text{multiplicity } p (\text{fact } K)$
proof –
have $\{1..<B\} = \{1..B - 1\}$ **using** $B1$ **by** *auto*
moreover have $K \leq B - 1$ **by** (*simp add: B-def*)
ultimately show *?thesis* **using** *legendre-sum*[*OF p, of K B - 1*] **by** *simp*

```

qed
have cnt: (∑ j ∈ {1..<B}. if j ≤ L0 then e else 0) = L0 * e
proof -
  have (∑ j ∈ {1..<B}. if j ≤ L0 then e else 0) = (∑ j ∈ {j ∈
    {1..<B}. j ≤ L0}. e)
    by (rule sum.inter-filter[symmetric]) simp
  also have {j ∈ {1..<B}. j ≤ L0} = {1..L0} by (auto simp: B-def)
  finally show ?thesis by simp
qed
have (∑ s ∈ T. multiplicity (int p) (x - s)) ≤ (∑ j ∈ {1..<B}. c j)
  by (rule sum-multiplicity-le[OF p T x hB]) (use hc in ⟨auto simp: c-def
L0-def⟩)
also have ... = 2 * (∑ j ∈ {1..<B}. K div p ^ j) + (∑ j ∈ {1..<B}.
  if j ≤ L0 then e else 0)
  by (simp add: c-def sum.distrib sum-distrib-left)
also have ... = 2 * multiplicity p (fact K) + L0 * e by (simp only: leg cnt)
finally show ?thesis .
qed
have prod-ne: (∏ s ∈ T. of-int x - of-int s :: rat) ≠ 0
  using T x by (auto simp: prod-zero-iff)
have vprod: vp p (∏ s ∈ T. of-int x - of-int s) = (∑ s ∈
  T. int (multiplicity (int p) (x - s)))
proof -
  have vp p (∏ s ∈ T. of-int x - of-int s) = (∑ s ∈
    T. vp p (of-int x - of-int s))
    by (rule vp-prod-rat[OF p T]) (use x in auto)
  also have ... = (∑ s ∈ T. int (multiplicity (int p) (x - s)))
    by (rule sum.cong[OF refl]) (simp add: vp-def qmultiplicity-of-int flip: of-int-diff)
  finally show ?thesis .
qed
have vfact: vp p (fact K ^ 2) = 2 * int (multiplicity p (fact K))
proof -
  have vp p (fact K :: rat) = qmultiplicity (int p) (of-nat (fact K)) by (simp
add: vp-def)
  also have ... = int (multiplicity (int p) (int (fact K))) by (rule qmultiplic-
ity-of-nat)
  also have ... = int (multiplicity p (fact K)) by (simp only: multiplicity-int)
  finally have vf: vp p (fact K :: rat) = int (multiplicity p (fact K)) .
  show ?thesis by (simp add: vp-power-rat[OF p] vf)
qed
have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer prime-imp-prime-elem)
have vp p (fact K ^ 2 / (∏ s ∈ T. of-int x - of-int s)) =
  vp p (fact K ^ 2) - vp p (∏ s ∈ T. of-int x - of-int s)
  unfolding vp-def by (rule qmultiplicity-divide[OF pe - prod-ne]) simp
hence vq: vp p (fact K ^ 2 / (∏ s ∈ T. of-int x - of-int s))
  = 2 * int (multiplicity p (fact K)) - (∑ s ∈ T. int (multiplicity (int p) (x
- s)))
  using vprod vfact by simp
have int (∑ s ∈ T. multiplicity (int p) (x - s)) ≤

```

```

    int (2 * multiplicity p (fact K) + L0 * e)
  using hsum by (simp only: of-nat-le-iff)
  hence (∑ s ∈ T. int (multiplicity (int p) (x - s))) ≤
    2 * int (multiplicity p (fact K)) + int L0 * int e
  by simp
  with vq have - (int L0 * int e) ≤ vp p (fact K ^ 2 / (∏ s ∈
    T. of-int x - of-int s)) by linarith
  hence (of-int (- (int L0 * int e)) :: rat) ≤ of-int (vp p (fact K ^ 2 / (∏ s ∈
    T. of-int x - of-int s)))
  by (simp only: of-int-le-iff)
  thus ?thesis unfolding vge-def by (simp add: L0-def mult.commute)
qed

```

17.6 Partial fractions with linear poles

```

lemma degree-div-le-self:
  fixes G M :: rat poly
  assumes M ≠ 0
  shows degree (G div M) ≤ degree G
proof (cases G div M = 0)
  case True
  thus ?thesis by simp
next
  case False
  have dm: degree (M * (G div M)) = degree M + degree (G div M) using assms
  False
  by (rule degree-mult-eq)
  have G mod M = 0 ∨ degree (G mod M) < degree (M * (G div M))
  using degree-mod-less[OF assms, of G] dm by auto
  hence degree (M * (G div M) + G mod M) = degree (M * (G div M))
  proof
    assume G mod M = 0
    thus ?thesis by (simp only: add-0-right)
  next
    assume degree (G mod M) < degree (M * (G div M))
    thus ?thesis by (rule degree-add-eq-left)
  qed
  hence degree G = degree M + degree (G div M) using dm by simp
  thus ?thesis by simp
qed
lemma linPoleProd-eq-qPoleProd: linPoleProd S = qPoleProd (of-int 'S)
  unfolding linPoleProd-def qPoleProd-def by (subst prod.reindex) (auto simp:
  inj-on-def)

```

```

lemma degree-linPoleProd: finite S ⟹ degree (linPoleProd S) = card S
proof -
  assume S: finite S
  have inj-on (of-int :: int ⇒ rat) S by (simp add: inj-on-def)
  thus ?thesis using S by (simp add: linPoleProd-eq-qPoleProd degree-qPoleProd)

```

card-image)
qed

lemma *lead-coeff-linPoleProd*: $\text{finite } S \implies \text{lead-coeff } (\text{linPoleProd } S) = 1$
by (simp add: linPoleProd-eq-qPoleProd qPoleProd-monic)

lemma *linPoleProd-nonzero*: $\text{finite } S \implies \text{linPoleProd } S \neq 0$
using lead-coeff-linPoleProd[of S] by auto

lemma *poly-linPoleProd*: $\text{poly } (\text{linPoleProd } S) y = (\prod_{r \in S} y - \text{of-int } r)$
unfolding linPoleProd-def poly-prod by (simp add: poly-monom)

Lagrange: $P \bmod \prod_{r \in S} (x - r) = \sum_{r \in S} -r \text{ res-}r \prod_{s \in S, s \neq r} (x - s).$

lemma *mod-linPoleProd*:

assumes *finite S*

shows $P \bmod \text{linPoleProd } S = (\sum_{r \in S} [: \text{linResidue } S \ P \ r:] * \text{linPoleProd } (S - \{r\}))$

proof –

define *S'* where $S' = (\text{of-int } 'S :: \text{rat set})$

have *inj*: $\text{inj-on } (\text{of-int } :: \text{int} \Rightarrow \text{rat}) \ X \text{ for } X \text{ by (simp add: inj-on-def)}$

have *fin'*: $\text{finite } S' \text{ using assms by (simp add: S'-def)}$

have *Pi*: $\text{linPoleProd } S = \text{qPoleProd } S' \text{ by (simp add: S'-def linPoleProd-eq-qPoleProd)}$

have *L*: $P = \text{linPoleProd } S * (P \text{ div } \text{linPoleProd } S) + (\sum_{s \in S'} [: \text{qRes } S' \ P \ s:] * \text{qPoleProd } (S' - \{s\}))$

unfolding *Pi* by (rule lagrange-decomp[OF fin'])

have *R*: $(\sum_{s \in S'} [: \text{qRes } S' \ P \ s:] * \text{qPoleProd } (S' - \{s\}))$
 $= (\sum_{r \in S} [: \text{linResidue } S \ P \ r:] * \text{linPoleProd } (S - \{r\}))$

proof –

have $(\sum_{s \in S'} [: \text{qRes } S' \ P \ s:] * \text{qPoleProd } (S' - \{s\}))$
 $= (\sum_{r \in S} [: \text{qRes } S' \ P \ (\text{of-int } r):] * \text{qPoleProd } (S' - \{\text{of-int } r\}))$

unfolding *S'-def* by (subst sum.reindex[OF inj]) simp

also have $\dots = (\sum_{r \in S} [: \text{linResidue } S \ P \ r:] * \text{linPoleProd } (S - \{r\}))$

proof (rule sum.cong[OF refl])

fix *r* assume *r*: $r \in S$

have *e*: $S' - \{\text{of-int } r\} = \text{of-int } ' (S - \{r\}) \text{ by (auto simp: S'-def)}$

have $\text{qRes } S' \ P \ (\text{of-int } r) = \text{linResidue } S \ P \ r$

unfolding *qRes-def* *linResidue-def* *e* by (subst prod.reindex[OF inj]) (simp

add: *o-def*)

moreover have $\text{qPoleProd } (S' - \{\text{of-int } r\}) = \text{linPoleProd } (S - \{r\})$

unfolding *e* by (simp add: linPoleProd-eq-qPoleProd)

ultimately show $[: \text{qRes } S' \ P \ (\text{of-int } r):] * \text{qPoleProd } (S' - \{\text{of-int } r\}) =$
 $[: \text{linResidue } S \ P \ r:] * \text{linPoleProd } (S - \{r\})$

by simp

qed

finally show *?thesis* .

qed

have $\text{linPoleProd } S * (P \text{ div } \text{linPoleProd } S) + P \bmod \text{linPoleProd } S$

$= \text{linPoleProd } S * (P \text{ div } \text{linPoleProd } S) + (\sum_{s \in S'} [: \text{qRes } S' \ P \ s:] * \text{qPoleProd } (S' -$

$S' -$


```

    {s}))
  by (subst mult-div-mod-eq) (rule L)
  hence  $P \bmod \text{linPoleProd } S = (\sum_{s \in S'} [:qRes \ S' \ P \ s:] * qPoleProd \ (S' - \{s\}))$ 
  by (simp only: add-left-cancel)
  thus ?thesis using R by simp
qed

```

Evaluating the partial-fraction decomposition away from the poles.

lemma *poly-div-linPoleProd*:

```

  assumes finite S  $\wedge r. r \in S \implies y - \text{of-int } r \neq 0$ 
  shows poly (P div linPoleProd S) y =
    poly P y / ( $\prod_{r \in S. y - \text{of-int } r} - (\sum_{r \in S. \text{linResidue } S \ P \ r / (y - \text{of-int } r)}$ ))
proof -
  define D where  $D = (\prod_{r \in S. y - \text{of-int } r})$ 
  have D0:  $D \neq 0$  using assms by (simp add: D-def prod-zero-iff)
  have rem: ( $\prod_{s \in S - \{r\}. y - \text{of-int } s} = D / (y - \text{of-int } r)$ ) if  $r: r \in S$  for r
  proof -
    have eq:  $D = (y - \text{of-int } r) * (\prod_{s \in S - \{r\}. y - \text{of-int } s})$  unfolding D-def
    by (rule prod.remove[OF assms(1) r])
    show ?thesis unfolding eq using assms(2)[OF r] by simp
  qed
  have poly P y = poly (linPoleProd S * (P div linPoleProd S) + P mod linPoleProd S) y by simp
  also have ... = D * poly (P div linPoleProd S) y + ( $\sum_{r \in S. \text{linResidue } S \ P \ r * (\prod_{s \in S - \{r\}. y - \text{of-int } s})}$ )
  by (simp add: mod-linPoleProd[OF assms(1)] poly-linPoleProd poly-sum D-def)
  also have ( $\sum_{r \in S. \text{linResidue } S \ P \ r * (\prod_{s \in S - \{r\}. y - \text{of-int } s})} = D * (\sum_{r \in S. \text{linResidue } S \ P \ r / (y - \text{of-int } r)}$ )
  unfolding sum-distrib-left by (rule sum.cong[OF refl]) (simp add: rem)
  finally have poly P y = D * (poly (P div linPoleProd S) y + ( $\sum_{r \in S. \text{linResidue } S \ P \ r / (y - \text{of-int } r)}$ ))
  by (simp add: distrib-left)
  hence poly (P div linPoleProd S) y + ( $\sum_{r \in S. \text{linResidue } S \ P \ r / (y - \text{of-int } r)}$ ) =
    poly P y / D
  using D0 by (simp add: eq-divide-eq mult.commute)
  thus ?thesis unfolding D-def[symmetric] by (simp add: algebra-simps)
qed

```

17.7 Valuations of integers

lemma *vp-int-le-nat-log*:

```

  prime p  $\implies z \neq 0 \implies$ 
    nat |z|  $\leq M \implies vp \ p \ (\text{of-int } z) \leq \text{int } (\text{nat-log } p \ M)$ 
proof -
  assume p: prime p and z:  $z \neq 0$  and M: nat |z|  $\leq M$ 
  let ?m = multiplicity (int p) z
  have int p  $^{\wedge} ?m \text{ dvd } z$  by (rule multiplicity-dvd)

```

hence $|int\ p \wedge ?m| \leq |z|$ **by** (rule dvd-imp-le-int[OF z])
 hence $int\ p \wedge ?m \leq |z|$ **by** simp
 also have $\dots \leq int\ M$ **using** M **by** (simp only: nat-le-iff)
 finally have $int\ p \wedge ?m \leq int\ M$.
 hence $int\ (p \wedge ?m) \leq int\ M$ **by** (simp only: of-nat-power)
 hence $p \wedge (Suc\ ?m - 1) \leq M$ **by** (simp only: of-nat-le-iff diff-Suc-1)
 hence $Suc\ ?m \leq floorlog\ p\ M$ **by** (rule floorlog-geI) (rule prime-gt-1-nat[OF p])
 thus $?thesis$ **by** (simp add: vp-def qmultiplicity-of-int nat-log-def)
qed

lemma lt-vp-of-dvd:

$prime\ p \implies z \neq 0 \implies int\ p \wedge (k + 1)\ dvd\ z \implies$
 $int\ k < vp\ p\ (of-int\ z)$

proof –

assume p : $prime\ p$ and z : $z \neq 0$ and d : $int\ p \wedge (k + 1)\ dvd\ z$
 have $prime\ (int\ p)$ **using** p **by** (simp add: prime-nat-int-transfer)
 hence pu : $\neg is-unit\ (int\ p)$ **by** (metis not-prime-unit)
 have $k + 1 \leq multiplicity\ (int\ p)\ z$
by (subst power-dvd-iff-le-multiplicity[OF z pu, symmetric]) (rule d)
 thus $?thesis$ **by** (simp add: vp-def qmultiplicity-of-int)
qed

lemma vge-inv-int:

$prime\ p \implies z \neq 0 \implies$
 $\neg int\ p \wedge (k + 1)\ dvd\ z \implies vge\ p\ (1 / of-int\ z)\ (\neg of-nat\ k)$

proof –

assume p : $prime\ p$ and z : $z \neq 0$ and d : $\neg int\ p \wedge (k + 1)\ dvd\ z$
 have $prime\ (int\ p)$ **using** p **by** (simp add: prime-nat-int-transfer)
 hence pu : $\neg is-unit\ (int\ p)$ **by** (metis not-prime-unit)
 have $multiplicity\ (int\ p)\ z \leq k$ **using** d $power-dvd-iff-le-multiplicity[OF\ z\ pu,\ of\ k + 1]$
by simp
 thus $?thesis$ **using** z **by** (simp add: vge-def vp-def inverse-eq-divide[symmetric]
 qmultiplicity-inverse qmultiplicity-of-int)
qed

The pole values $H^{\wedge}\{(5)\}$ - m for $m \leq K$ have $v-p \geq -5 \lfloor \log-p\ K \rfloor$.

lemma vge-H5: $prime\ p \implies m \leq K \implies$

$vge\ p\ (H5\ m)\ (\neg 5 * of-nat\ (nat-log\ p\ K))$

proof –

assume p : $prime\ p$ and mK : $m \leq K$
 show $?thesis$ **unfolding** $H5-def$
proof (rule vge-sum[OF p])
 fix v **assume** $v \in \{1..m\}$
 hence $v1$: $1 \leq v$ and vK : $v \leq K$ **using** mK **by** auto
 have nz : $(1 / of-nat\ v :: rat) \neq 0$ **using** $v1$ **by** simp
 have $\neg of-nat\ (nat-log\ p\ v) \leq (of-int\ (vp\ p\ (1 / of-nat\ v)) :: rat)$
using $vge-inv-nat[OF\ p\ v1]\ nz$ **by** (simp add: vge-def)
 moreover have $(of-nat\ (nat-log\ p\ v) :: rat) \leq of-nat\ (nat-log\ p\ K)$ **using** vK

by (simp add: nat-log-mono)
 ultimately show vge p (1 / of-nat v ^ 5) (- 5 * of-nat (nat-log p K))
 unfolding vge-def power-one-over[symmetric] vp-power-rat[OF p nz] by simp
 qed simp
 qed

lemma vpGge-pX: vpGge p pX 0
 by (auto simp: vpGge-iff coeff-monom vge-one vge-zero)

17.8 The product-difference estimate

$v(\prod (a-i + u) - \prod a-i) \geq v(u) - L + \sum v(a-i)$ when every $v(a-i) \leq L < v(u)$.

lemma vge-prod-add-sub-prod:
 assumes p: prime p and I: finite I and u: u ≠ 0
 and a0: $\bigwedge i. i \in I \implies$
 $a\ i \neq 0$ and aL: $\bigwedge i. i \in I \implies \text{of-int } (vp\ p\ (a\ i)) \leq L$
 and Lu: $L < \text{of-int } (vp\ p\ u)$
 shows vge p (($\prod i \in I. a\ i + u$) - ($\prod i \in I. a\ i$))
 (of-int (vp p u) - L + ($\sum i \in I. \text{of-int } (vp\ p\ (a\ i))$))
 using I a0 aL
 proof (induction I rule: finite-induct)
 case empty
 thus ?case by (simp add: vge-zero)
 next
 case (insert b I)
 have IH: vge p (($\prod i \in I. a\ i + u$) - ($\prod i \in I. a\ i$))
 (of-int (vp p u) - L + ($\sum i \in I. \text{of-int } (vp\ p\ (a\ i))$))
 using insert.IH insert.premis by simp
 have e: ($\prod i \in \text{insert } b\ I. a\ i + u$) - ($\prod i \in \text{insert } b\ I. a\ i$) =
 $a\ b * ((\prod i \in I. a\ i + u) - (\prod i \in I. a\ i)) + u * (\prod i \in$
 $I. a\ i + u)$
 using insert.hyps by (simp add: algebra-simps)
 have 1: vge p (a b * (($\prod i \in I. a\ i + u$) - ($\prod i \in I. a\ i$))
 (of-int (vp p u) - L + ($\sum i \in \text{insert } b\ I. \text{of-int } (vp\ p\ (a\ i))$))
 proof -
 have ab: vge p (a b) (of-int (vp p (a b))) by (simp add: vge-def)
 have vge p (a b * (($\prod i \in I. a\ i + u$) - ($\prod i \in I. a\ i$))
 (of-int (vp p (a b)) + (of-int (vp p u) - L + ($\sum i \in I. \text{of-int } (vp\ p\ (a\ i))$))))
 by (rule vge-mul[OF p ab IH])
 thus ?thesis by (simp add: sum.insert[OF insert.hyps] algebra-simps)
 qed
 have 2: vge p (u * ($\prod i \in I. a\ i + u$))
 (of-int (vp p u) - L + ($\sum i \in \text{insert } b\ I. \text{of-int } (vp\ p\ (a\ i))$))
 proof -
 have bI: b ∈ insert b I by simp
 have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer
 prime-imp-prime-elem)
 have lt: vp p (a i) < vp p u if i: i ∈ insert b I for i

```

proof –
  have of-int (vp p (a i)) < (of-int (vp p u) :: rat) using insert.prems(2)[OF
i] Lu
    by linarith
  thus ?thesis by simp
qed
have e: a i + u ≠ 0 ∧ vp p (a i + u) = vp p (a i) if i: i ∈ I for i
  using vp-add-eq-of-lt[OF p insert.prems(1) u lt] i by simp
have pne: (∏ i ∈ I. a i + u) ≠ 0 using e insert.hyps(1)
  by (simp add: prod-zero-iff)
have V1: vp p (u * (∏ i ∈ I. a i + u)) = vp p u + vp p (∏ i ∈ I. a i + u)
  unfolding vp-def by (rule qmultiplicity-mult[OF pe u pne])
have V2: vp p (∏ i ∈ I. a i + u) = (∑ i ∈ I. vp p (a i + u))
  by (rule vp-prod-rat[OF p insert.hyps(1)]) (use e in simp)
have V3: (∑ i ∈ I. vp p (a i + u)) = (∑ i ∈ I. vp p (a i))
  by (rule sum.cong[OF refl]) (simp add: e)
have of-int (vp p (a b)) ≤ L using insert.prems(2)[OF bI] .
hence of-int (vp p u) − L + (∑ i ∈ insert b I. of-int (vp p (a i)))
  ≤ (of-int (vp p (u * (∏ i ∈ I. a i + u))) :: rat)
  unfolding V1 V2 V3 by (simp add: sum.insert[OF insert.hyps])
thus ?thesis by (simp add: vge-def)
qed
show ?case unfolding e by (rule vge-add[OF p 1 2])
qed

```

17.9 Lemma 3.3

```

lemma symPoles-subset: symPoles K ⊆ {− int K..int K}
  by (auto simp: symPoles-def)

```

```

lemma finite-symPoles: finite (symPoles K)
  using symPoles-subset finite-subset by blast

```

```

lemma symPoles-abs-sub-le: r ∈ symPoles K ⇒
  s ∈ symPoles K ⇒ nat |r − s| ≤ 2 * K
  by (auto simp: symPoles-def)

```

```

lemma dIdx-le-symPoles: r ∈ symPoles K ⇒ dIdx r ≤ K
  by (auto simp: symPoles-def dIdx-def)

```

theorem *far-pole-valuation-bound*:

```

  assumes p: prime p and A: IntValuedAt p A and dA: degree A ≤ d
  shows vpGge p (tauX (symPoles K) ([:fact K ^ 2:] * A))
    (− 6 * of-nat (nat-log p (max (2 * K) (d + 1))) − of-nat (multiplicity p
    (24 :: nat)))

```

```

proof –
  define R where R = symPoles K
  define G where G = [:fact K ^ 2:] * A
  define L0 where L0 = nat-log p (2 * K)

```

```

define Ld where Ld = nat-log p (max 1 d)
define L' where L' = nat-log p (max (2 * K) (d + 1))
have L0L: L0 ≤ L' unfolding L0-def L'-def by (rule nat-log-mono) (auto simp:
max-def)
have LdL: Ld ≤ L' unfolding Ld-def L'-def by (rule nat-log-mono) (auto simp:
max-def)
have Ld1: nat-log p (d + 1) ≤ L' unfolding L'-def
by (rule nat-log-mono) (auto simp: max-def)
have LKL: nat-log p K ≤ L' unfolding L'-def by (rule nat-log-mono) (auto
simp: max-def)
have finR: finite R and Rsub: R ⊆ {− int K..int K}
by (simp-all add: R-def finite-symPoles symPoles-subset)
have Geval: ⋀y. poly G y = fact K ^ 2 * poly A y by (simp add: G-def)
— (a) the residues have v-p ≥ −L0
have F0: ⋀r. r ∈ R ⇒
vge p (fact K ^ 2 / (⋀ s ∈ R − {r}. of-int r − of-int s)) (− of-nat L0)
proof −
fix r assume r: r ∈ R
have hc: card {s ∈ R − {r}. int p ^ j dvd (r − s)} ≤ 2 * (K div p ^ j) + (if
j ≤
nat-log p (2 * K) then 1 else 0)
if j1: 1 ≤ j for j
proof −
have q: 0 < p ^ j using prime-gt-0-nat[OF p] by simp
have s1: R − {r} ⊆ {− int K..int K} using Rsub by auto
have s2: r ∈ {− int K..int K} using Rsub r by auto
have s3: r ∉ R − {r} by simp
have s4: int (p ^ j) dvd (r − r) by simp
have card {s ∈ R − {r}. int (p ^ j) dvd (r − s)} ≤ 2 * K div p ^ j
by (rule card-filter-dvd-le-excl[OF q s1 s2 s3 s4])
moreover have 2 * K div p ^ j ≤ 2 * (K div p ^ j) + (if j ≤
nat-log p (2 * K) then 1 else 0)
using two-mul-div-le-nat[OF q, of K] div-eq-zero-of-gt-nat-log[OF p, of K j]
by auto
ultimately show ?thesis by simp
qed
have vge p (fact K ^ 2 / (⋀ s ∈
R − {r}. of-int r − of-int s)) (− of-nat (1 * nat-log p (2 * K)))
by (rule vge-factorial-sq-div-prod[OF p - hc]) (use finR in auto)
thus vge p (fact K ^ 2 / (⋀ s ∈ R − {r}. of-int r − of-int s)) (− of-nat L0)
by (simp add: L0-def)
qed
have res: ⋀r. r ∈ R ⇒ vge p (linResidue R G r) (− of-nat L0)
proof −
fix r assume r: r ∈ R
have e: linResidue R G r = poly A (of-int r) * (fact K ^ 2 / (⋀ s ∈
R − {r}. of-int r − of-int s))
by (simp add: linResidue-def Geval mult.commute)
have vge p (poly A (of-int r) * (fact K ^ 2 / (⋀ s ∈

```

```

      R - {r}. of-int r - of-int s))) (0 + - of-nat L0)
    by (rule vge-mul[OF p IntValuedAt-vge[OF A] F0[OF r]])
    thus vge p (linResidue R G r) (- of-nat L0) unfolding e by simp
  qed
  — (c) the polynomial part  $Q = G \operatorname{div} \prod (x - r)$ , bounded at the  $d + 1$  integers
   $K+1, \dots, K+1+d$ 
  define Q where Q = G div linPoleProd R
  have dQ: degree Q ≤ d
  proof —
    have degree Q ≤ degree G unfolding Q-def
    by (rule degree-div-le-self[OF linPoleProd-nonzero[OF finR]])
    also have degree G ≤ degree A unfolding G-def using degree-smult-le[of fact
  K ^ 2 A]
    by simp
    also note dA
    finally show ?thesis .
  qed
  define a where a = (of-nat (L0 + max L0 Ld) :: rat)
  have a2L0: - (2 * of-nat L0) ≥ - a and aLd: - (of-nat Ld + of-nat L0) ≥ -
  a
  by (simp-all add: a-def)
  have hval:  $\bigwedge i. i \leq d \implies$ 
    vge p (poly Q (of-int (int K + 1) + of-nat i)) (- a)
  proof —
    fix i assume i ≤ d
    define n where n = int K + 1 + int i
    have ncast: of-int (int K + 1) + of-nat i = (of-int n :: rat) by (simp add:
  n-def)
    have nr:  $\bigwedge r. r \in R \implies n - r \neq 0$  using Rsub
    by (auto simp: n-def)
    have nrq:  $\bigwedge r. r \in R \implies \operatorname{of-int} n - \operatorname{of-int} r \neq (0 :: \operatorname{rat})$ 
    using nr by simp
    have Qn: poly Q (of-int n) = fact K ^ 2 * poly A (of-int n) / ( $\prod r \in$ 
    R. of-int n - of-int r)
      - ( $\sum r \in R. \operatorname{linResidue} R G r / (\operatorname{of-int} n - \operatorname{of-int} r)$ )
    by (simp only: Q-def poly-div-linPoleProd[OF finR nrq] Geval)
    — a pole term  $\operatorname{res-r} / (n - r)$  with  $v\text{-}p(n - r) \leq L0$ 
    have hterm:  $\bigwedge r. r \in R \implies$ 
       $\neg \operatorname{int} p \wedge (L0 + 1) \operatorname{dvd} (n - r) \implies$ 
      vge p (linResidue R G r / (of-int n - of-int r)) (- a)
    proof —
      fix r assume r: r ∈ R and nd:  $\neg \operatorname{int} p \wedge (L0 + 1) \operatorname{dvd} (n - r)$ 
      have e: linResidue R G r / (of-int n - of-int r) = linResidue R G r * (1 /
  of-int (n - r))
      by simp
      have vge p (linResidue R G r * (1 / of-int (n - r))) (- of-nat L0 + - of-nat
  L0)
      by (rule vge-mul[OF p res[OF r] vge-inv-int[OF p nr[OF r] nd]])
      hence vge p (linResidue R G r / (of-int n - of-int r)) (- of-nat L0 + -

```

```

of-nat L0)
  unfolding e .
  thus vge p (linResidue R G r / (of-int n - of-int r)) (- a)
  by (rule vge-mono) (use a2L0 in simp)
qed
show vge p (poly Q (of-int (int K + 1) + of-nat i)) (- a)
  unfolding ncast
proof (cases  $\exists r0 \in R. \text{int } p \wedge (L0 + 1) \text{ dvd } (n - r0)$ )
  case True
  — Case B:  $n$  is  $p$ -adically close to exactly one pole  $r0$ 
  then obtain r0 where r0:  $r0 \in R$  and dv:  $\text{int } p \wedge (L0 + 1) \text{ dvd } (n - r0)$ 
by blast
define T where  $T = R - \{r0\}$ 
have Tsub:  $T \subseteq \{- \text{int } K .. \text{int } K\}$  using Rsub by (auto simp: T-def)
have sep:  $\bigwedge s. s \in T \implies \neg \text{int } p \wedge (L0 + 1) \text{ dvd } (r0 - s)$ 
proof -
  fix s assume s:  $s \in T$ 
  have sR:  $s \in R$  and sr:  $s \neq r0$  using s by (auto simp: T-def)
  have ne:  $r0 - s \neq 0$  using sr by simp
  have le:  $\text{nat } |r0 - s| \leq 2 * K$  using symPoles-abs-sub-le[of r0 K s] r0 sR
  by (simp add: R-def)
  have K0:  $0 < 2 * K$  using le ne by (cases  $K = 0$ ) auto
  have lt:  $2 * K < p \wedge (L0 + 1)$  unfolding L0-def by (rule lt-pow-nat-log-succ[OF
p K0])
  show  $\neg \text{int } p \wedge (L0 + 1) \text{ dvd } (r0 - s)$ 
  proof
    assume int p  $\wedge (L0 + 1) \text{ dvd } (r0 - s)$ 
    hence  $|\text{int } p \wedge (L0 + 1)| \leq |r0 - s|$ 
    by (rule dvd-imp-le-int[OF ne])
    hence  $\text{int } p \wedge (L0 + 1) \leq |r0 - s|$  by simp
    also have  $\dots \leq \text{int } (2 * K)$  using le by (simp only: nat-le-iff)
    finally have  $\text{int } p \wedge (L0 + 1) \leq \text{int } (2 * K)$  .
    hence  $\text{int } (p \wedge (L0 + 1)) \leq \text{int } (2 * K)$  by (simp only: of-nat-power)
    hence  $p \wedge (L0 + 1) \leq 2 * K$  by (simp only: of-nat-le-iff)
    thus False using lt by simp
  qed
qed
have P1:  $(\prod s \in T. \text{of-int } r0 - \text{of-int } s :: \text{rat}) \neq 0$ 
  and P2:  $(\prod s \in T. \text{of-int } n - \text{of-int } s :: \text{rat}) \neq 0$ 
  using finR nrq by (auto simp: T-def prod-zero-iff)
have Fn:  $vge p (\text{fact } K \wedge 2 / (\prod s \in T. \text{of-int } n - \text{of-int } s)) (- \text{of-nat } L0)$ 
proof -
  have finT: finite T using finR by (simp add: T-def)
  have hc:  $\text{card } \{s \in T. \text{int } p \wedge j \text{ dvd } (n - s)\} \leq 2 * (K \text{ div } p \wedge j) + (\text{if } j \leq \text{nat-log } p (2 * K) \text{ then } 1 \text{ else } 0)$ 
  if j1:  $1 \leq j$  for j
  proof (cases  $j \leq L0$ )
    case True
    have q:  $0 < p \wedge j$  using prime-gt-0-nat[OF p] by simp

```

```

have int p ^ j dvd int p ^ (L0 + 1) using True by (intro le-imp-power-dvd)
simp
  hence dj: int (p ^ j) dvd (n - r0) using dvd-trans[OF - dv] by simp
  have card {s ∈ T. int (p ^ j) dvd (n - s)} ≤ 2 * K div p ^ j
    by (rule card-filter-dvd-le-excl[OF q Tsub - - dj]) (use r0 Rsub in ‹auto
simp:
  T-def›)
  hence card {s ∈ T. int p ^ j dvd (n - s)} ≤ 2 * K div p ^ j
    by (simp only: of-nat-power)
  also have ... ≤ 2 * (K div p ^ j) + 1 by (rule two-mul-div-le-nat[OF q])
  finally have c: card {s ∈ T. int p ^ j dvd (n - s)} ≤ 2 * (K div p ^ j
+ 1 .
  have jle: j ≤ nat-log p (2 * K) using True by (simp add: L0-def)
  show ?thesis unfolding if-P[OF jle] by (rule c)
next
case False
have {s ∈ T. int p ^ j dvd (n - s)} = {}
proof (rule ccontr)
  assume {s ∈ T. int p ^ j dvd (n - s)} ≠ {}
  then obtain s where s: s ∈ T and ds: int p ^ j dvd (n - s) by blast
  have int p ^ (L0 + 1) dvd int p ^ j using False by (intro le-imp-power-dvd)
simp
  hence int p ^ (L0 + 1) dvd (n - s) using ds by (rule dvd-trans)
  hence int p ^ (L0 + 1) dvd (n - s) - (n - r0) using dv by (rule
dvd-diff)
  moreover have (n - s) - (n - r0) = r0 - s by simp
  ultimately have int p ^ (L0 + 1) dvd (r0 - s) by simp
  thus False using sep[OF s] by simp
qed
  hence c0: card {s ∈ T. int p ^ j dvd (n - s)} = 0 by (simp only:
card.empty)
  show ?thesis unfolding c0 by simp
qed
have vge p (fact K ^ 2 / (∏ s ∈
  T. of-int n - of-int s)) (- of-nat (1 * nat-log p (2 * K)))
  by (rule vge-factorial-sq-div-prod[OF p finT - hc]) (use nr in ‹auto simp:
  T-def›)
  thus ?thesis by (simp add: L0-def)
qed
have PD: vge p (((∏ s ∈ T. of-int r0 - of-int s) - (∏ s ∈
  T. of-int n - of-int s)) /
  ((∏ s ∈
    T. of-int r0 - of-int s) * (of-int n - of-int r0))) (- of-nat L0)
proof -
  have finT: finite T using finR by (simp add: T-def)
  have pe: prime-elem (int p) using p
    by (simp add: prime-nat-int-transfer prime-imp-prime-elem)
  define u where u = (of-int n - of-int r0 :: rat)
  have u0: u ≠ 0 using nrq[OF r0] by (simp add: u-def)

```



```

have a0: (of-int r0 - of-int s :: rat) ≠ 0 if s ∈ T for s using that
  by (auto simp: T-def)
have aL: of-int (vp p (of-int r0 - of-int s :: rat)) ≤ (of-nat L0 :: rat)
  if s: s ∈ T for s
proof -
  have sR: s ∈ R and sr: s ≠ r0 using s by (auto simp: T-def)
  have vp p (of-int (r0 - s)) ≤ int (nat-log p (2 * K))
    by (rule vp-int-le-nat-log[OF p]) (use sr symPoles-abs-sub-le[of r0 K s]
r0 sR in
    ⟨simp-all add: R-def⟩)
  hence vp p (of-int r0 - of-int s) ≤ int L0 by (simp add: L0-def)
  hence (of-int (vp p (of-int r0 - of-int s :: rat)) :: rat) ≤ of-int (int L0)
    by (simp only: of-int-le-iff)
  thus ?thesis by simp
qed
have Lu: (of-nat L0 :: rat) < of-int (vp p u)
proof -
  have int L0 < vp p (of-int (n - r0)) by (rule lt-vp-of-dvd[OF p nr[OF
r0] dv])
  hence int L0 < vp p u by (simp add: u-def)
  hence (of-int (int L0) :: rat) < of-int (vp p u) by (simp only: of-int-less-iff)
  thus ?thesis by simp
qed
have main: vge p ((∏ s ∈ T. (of-int r0 - of-int s) + u) - (∏ s ∈
  T. of-int r0 - of-int s))
  (of-int (vp p u) - of-nat L0 + (∑ s ∈
    T. of-int (vp p (of-int r0 - of-int s))))
  by (rule vge-prod-add-sub-prod[OF p finT u0 a0 aL Lu])
have e1: (∏ s ∈ T. (of-int r0 - of-int s) + u) = (∏ s ∈
  T. of-int n - of-int s) by (simp add: u-def)
define Pr where Pr = (∏ s ∈ T. of-int r0 - of-int s :: rat)
define Pn where Pn = (∏ s ∈ T. of-int n - of-int s :: rat)
have main': vge p (Pn - Pr) (of-int (vp p u) - of-nat L0 + (∑ s ∈
  T. of-int (vp p (of-int r0 - of-int s))))
  using main unfolding e1 Pr-def[symmetric] Pn-def[symmetric] .
have Pr0: Pr ≠ 0 unfolding Pr-def using P1 .
have vPr: vp p Pr = (∑ s ∈ T. vp p (of-int r0 - of-int s)) unfolding
Pr-def
  by (rule vp-prod-rat[OF p finT a0])
have vge p ((Pr - Pn) / (Pr * u)) (- of-nat L0)
  unfolding vge-def
proof
  assume (Pr - Pn) / (Pr * u) ≠ 0
  hence N0: Pn - Pr ≠ 0 by auto
  have N0': Pr - Pn ≠ 0 using N0 by simp
  have m1: vp p (Pn - Pr) = vp p (Pr - Pn)
  proof -
    have Pn - Pr = - (Pr - Pn) by simp
    thus ?thesis by (simp only: vp-def qmultiplicity-minus)

```

```

qed
have A1: of-int (vp p u) - of-nat L0 + (∑ s ∈
  T. of-int (vp p (of-int r0 - of-int s)))
  ≤ (of-int (vp p (Pr - Pn)) :: rat)
using main' N0 m1 by (simp add: vge-def)
have vq: vp p ((Pr - Pn) / (Pr * u)) = vp p (Pr - Pn) - (vp p Pr +
vp p u)
unfolding vp-def by (simp add: qmultiplicity-divide[OF pe N0] qmulti-
plicity-mult[OF pe
  Pr0 u0] Pr0 u0)
have (of-int (vp p ((Pr - Pn) / (Pr * u))) :: rat)
  = of-int (vp p (Pr - Pn)) - ((∑ s ∈
  T. of-int (vp p (of-int r0 - of-int s))) + of-int (vp p u))
unfolding vq vPr by simp
with A1 show - of-nat L0 ≤ (of-int (vp p ((Pr - Pn) / (Pr * u))) ::
rat) by linarith
qed
thus ?thesis unfolding Pr-def Pn-def u-def .
qed
have split: fact K ^ 2 * poly A (of-int n) / (∏ r ∈ R. of-int n - of-int r)
  - (∑ r ∈ R. linResidue R G r / (of-int n - of-int r)) =
  (poly A (of-int n) - poly A (of-int r0)) / (of-int n - of-int r0) *
  (fact K ^ 2 / (∏ s ∈ T. of-int n - of-int s))
  + poly A (of-int r0) * ((fact K ^ 2 / (∏ s ∈ T. of-int n - of-int s)) *
  (((∏ s ∈ T. of-int r0 - of-int s) - (∏ s ∈
  T. of-int n - of-int s)) /
  ((∏ s ∈ T. of-int r0 - of-int s) * (of-int n - of-int r0))))
  - (∑ r ∈ T. linResidue R G r / (of-int n - of-int r))
proof -
have pr: (∏ r ∈ R. of-int n - of-int r) =
  (of-int n - of-int r0) * (∏ s ∈ T. of-int n - of-int s :: rat)
unfolding T-def by (rule prod.remove[OF finR r0])
have sm: (∑ r ∈ R. linResidue R G r / (of-int n - of-int r)) =
  linResidue R G r0 / (of-int n - of-int r0) + (∑ r ∈
  T. linResidue R G r / (of-int n - of-int r))
unfolding T-def by (rule sum.remove[OF finR r0])
have rr: linResidue R G r0 = fact K ^ 2 * poly A (of-int r0) / (∏ s ∈
  T. of-int r0 - of-int s)
by (simp add: linResidue-def Geval T-def)
have w0: (of-int n - of-int r0 :: rat) ≠ 0 by (rule nrq[OF r0])
have gen: c * An / (w * Pn) - (c * Ar / Pr / w + Sg) =
  (An - Ar) / w * (c / Pn) + Ar * ((c / Pn) * ((Pr - Pn) / (Pr * w)))
- Sg
if Pr ≠ 0 Pn ≠ 0 w ≠ 0 for c An Ar Pr Pn w Sg :: rat
using that by (simp add: field-simps)
show ?thesis unfolding pr sm rr by (rule gen[OF P1 P2 w0])
qed
have t1: vge p ((poly A (of-int n) - poly A (of-int r0)) / (of-int n - of-int
r0)) *

```

```

      (fact  $K \wedge 2 / (\prod s \in T. \text{of-int } n - \text{of-int } s)) (- a)$ 
proof –
  have  $dq: n \neq r0$  using  $nr[OF\ r0]$  by  $simp$ 
  have  $vge\ p\ ((poly\ A\ (\text{of-int } n) - poly\ A\ (\text{of-int } r0)) / (\text{of-int } n - \text{of-int } r0))$ 
*
      (fact  $K \wedge 2 / (\prod s \in$ 
         $T. \text{of-int } n - \text{of-int } s)) (- \text{of-nat } (\text{nat-log } p\ (\max\ 1\ d)) + -$ 
 $\text{of-nat } L0)$ 
    by  $(rule\ vge\text{-mul}[OF\ p\ \text{diffQuot-vge}[OF\ p\ A\ dA\ dq]\ Fn])$ 
    thus  $?thesis$  by  $(rule\ vge\text{-mono})\ (use\ aLd\ in\ \langle simp\ add: Ld\text{-def} \rangle)$ 
qed
  have  $t2: vge\ p\ (poly\ A\ (\text{of-int } r0) * ((fact\ K \wedge 2 / (\prod s \in$ 
     $T. \text{of-int } n - \text{of-int } s)) * ((\prod s \in T. \text{of-int } r0 - \text{of-int } s) - (\prod s \in$ 
     $T. \text{of-int } n - \text{of-int } s)) / ((\prod s \in T. \text{of-int } r0 - \text{of-int } s) * (\text{of-int } n - \text{of-int } r0)))) (- a)$ 
    using  $vge\text{-mul}[OF\ p\ \text{IntValuedAt-vge}[OF\ A]\ vge\text{-mul}[OF\ p\ Fn\ PD]]$ 
    by  $(rule\ vge\text{-mono})\ (use\ a2L0\ in\ simp)$ 
  have  $t3: vge\ p\ (\sum r \in T. \text{linResidue } R\ G\ r / (\text{of-int } n - \text{of-int } r)) (- a)$ 
proof  $(rule\ vge\text{-sum}[OF\ p])$ 
  show  $finite\ T$  using  $finR$  by  $(simp\ add: T\text{-def})$ 
  fix  $r$  assume  $r: r \in T$ 
  have  $rR: r \in R$  using  $r$  by  $(simp\ add: T\text{-def})$ 
  have  $\neg int\ p \wedge (L0 + 1)\ dvd\ (n - r)$ 
proof
    assume  $int\ p \wedge (L0 + 1)\ dvd\ (n - r)$ 
    hence  $int\ p \wedge (L0 + 1)\ dvd\ (n - r) - (n - r0)$  using  $dv$  by  $(rule\ dvd\text{-diff})$ 
    moreover have  $(n - r) - (n - r0) = r0 - r$  by  $simp$ 
    ultimately show  $False$  using  $sep[OF\ r]$  by  $simp$ 
qed
  thus  $vge\ p\ (\text{linResidue } R\ G\ r / (\text{of-int } n - \text{of-int } r)) (- a)$  by  $(rule\ hterm[OF\ rR])$ 
qed
  show  $vge\ p\ (poly\ Q\ (\text{of-int } n)) (- a)$ 
  unfolding  $Qn\ split$  by  $(intro\ vge\text{-diff}[OF\ p]\ vge\text{-add}[OF\ p]\ t1\ t2\ t3)$ 
next
  case  $False$ 
  — Case A:  $n$  is  $p$ -adically far from every pole
  have  $Gn: vge\ p\ (fact\ K \wedge 2 / (\prod r \in R. \text{of-int } n - \text{of-int } r)) (- \text{of-nat } (2$ 
*  $L0))$ 
proof –
  have  $hc: card\ \{r \in R. int\ p \wedge j\ dvd\ (n - r)\} \leq 2 * (K\ div\ p \wedge j) + (if\ j \leq$ 
 $\text{nat-log } p\ (2 * K)\ \text{then } 2\ \text{else } 0)$ 
    if  $j1: 1 \leq j$  for  $j$ 
proof  $(cases\ j \leq L0)$ 
  case  $True$ 
  have  $q: 0 < p \wedge j$  using  $prime\text{-gt-0-nat}[OF\ p]$  by  $simp$ 
  have  $card\ \{r \in R. int\ (p \wedge j)\ dvd\ (n - r)\} \leq 2 * K\ div\ p \wedge j + 1$ 

```

```

    by (rule card-filter-dvd-le[OF q Rsub])
  hence card {r ∈ R. int p ^ j dvd (n - r)} ≤ 2 * K div p ^ j + 1
    by (simp only: of-nat-power)
  moreover have 2 * K div p ^ j ≤ 2 * (K div p ^ j) + 1
    by (rule two-mul-div-le-nat[OF q])
  ultimately have c: card {r ∈ R. int p ^ j dvd (n - r)} ≤ 2 * (K div p
^ j) + 2
    by linarith
  have jle: j ≤ nat-log p (2 * K) using True by (simp add: L0-def)
  show ?thesis unfolding if-P[OF jle] by (rule c)
next
case False
have {r ∈ R. int p ^ j dvd (n - r)} = {}
proof (rule ccontr)
  assume {r ∈ R. int p ^ j dvd (n - r)} ≠ {}
  then obtain r where r: r ∈ R and dr: int p ^ j dvd (n - r) by blast
  have int p ^ (L0 + 1) dvd int p ^ j using False by (intro le-imp-power-dvd)
simp
  hence int p ^ (L0 + 1) dvd (n - r) using dr by (rule dvd-trans)
  thus False using ‹∃ r0 ∈
    R. int p ^ (L0 + 1) dvd (n - r0)› r by blast
qed
  hence c0: card {r ∈ R. int p ^ j dvd (n - r)} = 0 by (simp only:
card.empty)
  show ?thesis unfolding c0 by simp
qed
have vge p (fact K ^ 2 / (∏ r ∈
  R. of-int n - of-int r)) (- of-nat (2 * nat-log p (2 * K)))
  by (rule vge-factorial-sq-div-prod[OF p finR - hc]) (use nr in auto)
  thus ?thesis by (simp add: L0-def)
qed
have s1: vge p (fact K ^ 2 * poly A (of-int n) / (∏ r ∈
  R. of-int n - of-int r)) (- a)
proof -
  have e: fact K ^ 2 * poly A (of-int n) / (∏ r ∈ R. of-int n - of-int r)
    = poly A (of-int n) * (fact K ^ 2 / (∏ r ∈ R. of-int n - of-int r))
    by (simp add: mult.commute)
  show ?thesis unfolding e using vge-mul[OF p IntValuedAt-vge[OF A] Gn]
    by (rule vge-mono) (use a2L0 in simp)
qed
have s2: vge p (∑ r ∈ R. linResidue R G r / (of-int n - of-int r)) (- a)
proof (rule vge-sum[OF p finR])
  fix r assume r: r ∈ R
  have ¬ int p ^ (L0 + 1) dvd (n - r) using False r by blast
  thus vge p (linResidue R G r / (of-int n - of-int r)) (- a) by (rule
hterm[OF r])
qed
show vge p (poly Q (of-int n)) (- a)
  unfolding Qn by (intro vge-diff[OF p] s1 s2)

```

```

    qed
  qed
  have Qv: vge p (tau Q) (- a - 4 * of-nat (nat-log p (d + 1)))
    by (rule tau-vge-of-values[OF p dQ hval])
  — assemble the coefficients of  $\tau\text{-}X(G) = [\tau(Q)] + \sum -r [\text{res-}r:] * (H5 (dIdx r) - X)$ 
  have num: - 6 * of-nat L' - of-nat (multiplicity p (24::nat)) ≤
    - a - 4 * (of-nat (nat-log p (d + 1)) :: rat)
  proof —
    have L0 + max L0 Ld ≤ 2 * L' using L0L LdL by simp
    hence (of-nat (L0 + max L0 Ld) :: rat) ≤ of-nat (2 * L') by (simp only:
of-nat-le-iff)
    hence a ≤ 2 * of-nat L' by (simp add: a-def)
    moreover have (of-nat (nat-log p (d + 1)) :: rat) ≤ of-nat L' using Ld1 by
simp
    moreover have (0::rat) ≤ of-nat (multiplicity p (24::nat)) by simp
    ultimately show ?thesis by linarith
  qed
  have pole:  $\bigwedge r. r \in R \implies vpGge\ p\ ([:\text{linResidue}\ R\ G\ r:] * \text{tauPole}\ r)$ 
    (- 6 * of-nat L' - of-nat (multiplicity p (24::nat)))
  proof —
    fix r assume r: r ∈ R
    have rK: r ∈ symPoles K using r by (simp add: R-def)
    have h: vge p (H5 (dIdx r)) (- 5 * of-nat (nat-log p K))
      by (rule vge-H5[OF p dIdx-le-symPoles[OF rK]])
    have x: vpGge p (- pX) (- 5 * of-nat (nat-log p K))
      by (rule vpGge-neg[OF vpGge-mono[OF vpGge-pX]]) simp
    have tp: vpGge p (tauPole r) (- 5 * of-nat (nat-log p K))
    unfolding tauPole-def diff-conv-add-uminus by (rule vpGge-add[OF p vpGge-C[OF
h] x])
    have vpGge p ([:\text{linResidue}\ R\ G\ r:] * tauPole r) (- of-nat L0 + - 5 * of-nat
(nat-log p K))
      by (rule vpGge-mul[OF p vpGge-C[OF res[OF r]] tp])
    moreover have - 6 * of-nat L' - of-nat (multiplicity p (24::nat)) ≤
      - of-nat L0 + - 5 * (of-nat (nat-log p K) :: rat)
    proof —
      have (of-nat L0 :: rat) ≤ of-nat L' using L0L by simp
      moreover have (of-nat (nat-log p K) :: rat) ≤ of-nat L' using LKL by simp
      moreover have (0::rat) ≤ of-nat (multiplicity p (24::nat)) by simp
      ultimately show ?thesis by linarith
    qed
    ultimately show vpGge p ([:\text{linResidue}\ R\ G\ r:] * tauPole r) (- 6 * of-nat L'
—
      of-nat (multiplicity p (24::nat)))
      by (rule vpGge-mono)
    qed
  have vpGge p (tauX R G) (- 6 * of-nat L' - of-nat (multiplicity p (24::nat)))
    unfolding tauX-def Q-def[symmetric]
    by (intro vpGge-add[OF p] vpGge-C vge-mono[OF Qv num] vpGge-sum[OF p

```

```

    finR] pole)
  thus ?thesis by (simp add: R-def G-def L'-def)
qed

end

theory Zeta5-Small-Primes
  imports Zeta5-Far-Poles
begin

```

18 The small-prime bound (3.12)

Section 3.3 of the paper: the basis polynomials $f_i(-x^2)$ are integer-valued, the determinant identity (3.11), and the small-prime bound (3.12) derived from Lemma 3.3. Lean theory *SmallPrimes*.

18.1 Section 3.3 Integer-valued basis polynomials

A takes integer values at integers.

definition *IntEval* :: *rat poly* $\Rightarrow$ *bool* **where**
IntEval $A \iff (\forall x :: \text{int}. \exists z :: \text{int}. \text{poly } A (\text{of-int } x) = \text{of-int } z)$

lemma *IntEval-IntValuedAt*: *prime p* $\implies$ *IntEval* $A \implies$ *IntValuedAt p A*

proof –

```

  assume prime p and A: IntEval A
  show ?thesis unfolding IntValuedAt-def vp-def
  proof (intro allI impI)
    fix x :: int
    obtain z :: int where z: poly A (of-int x) = of-int z using A unfolding
    IntEval-def by blast
    show 0 ≤ qmultiplicity (int p) (poly A (of-int x)) unfolding z qmultiplic-
    ity-of-int
    by simp
  qed
qed

```

lemma *IntEval-mult*: *IntEval* $A \implies$ *IntEval* $B \implies$ *IntEval* $(A * B)$

proof –

```

  assume A: IntEval A and B: IntEval B
  show ?thesis unfolding IntEval-def
  proof
    fix x :: int
    obtain a b :: int where poly A (of-int x) = of-int a poly B (of-int x) = of-int b
    using A B unfolding IntEval-def by blast
    thus  $\exists z :: \text{int}. \text{poly } (A * B) (\text{of-int } x) = \text{of-int } z$  by (intro exI[of - a * b]) simp
  qed
qed

```

```

lemma IntEval-power: IntEval A  $\implies$  IntEval (A  $\wedge$  n)
proof (induction n)
  case 0
    have IntEval 1 unfolding IntEval-def by (auto intro!: exI[of - 1])
    thus ?case by simp
next
  case (Suc n)
    have IntEval (A  $\wedge$  n) using Suc by simp
    thus ?case unfolding power-Suc by (rule IntEval-mult[OF Suc.prems])
qed

lemma IntEval-add: IntEval A  $\implies$  IntEval B  $\implies$  IntEval (A + B)
proof -
  assume A: IntEval A and B: IntEval B
  show ?thesis unfolding IntEval-def
  proof
    fix x :: int
    obtain a b :: int where poly A (of-int x) = of-int a poly B (of-int x) = of-int b
    using A B unfolding IntEval-def by blast
    thus  $\exists z :: \text{int. } \text{poly } (A + B) (\text{of-int } x) = \text{of-int } z$  by (intro exI[of - a + b]) simp
  qed
qed

lemma IntEval-C-int: IntEval [:of-int z:]
  by (simp add: IntEval-def)

lemma IntEval-pX: IntEval pX
  unfolding IntEval-def by (simp add: poly-monom)

binom(s x + m, k) is integer-valued (s, m  $\in \mathbb{Z}$ ).

lemma IntEval-bp-pcompose: IntEval (pcompose (bp k) [:of-int m, of-int s:])
  unfolding IntEval-def
proof
  fix x :: int
  have e: poly [:of-int m, of-int s:] (of-int x) = (of-int (m + x * s) :: rat) by simp
  obtain z :: int where poly (bp k) (of-int (m + x * s)) = of-int z using bp-eval-int
  by blast
  thus  $\exists z :: \text{int. } \text{poly } (\text{pcompose } (\text{bp } k) [: \text{of-int } m, \text{of-int } s: ]) (\text{of-int } x) = \text{of-int } z$ 
    unfolding poly-pcompose e by blast
qed

lemma poly-bp: poly (bp k) y = ( $\prod_{i < k. y - \text{of-nat } i}$ ) / fact k
  unfolding bp-def by (simp add: poly-prod poly-monom)

 $\prod_{j=1}^N (j^2 - x^2) = \prod_{k < N} (N - k - x)(x + N - k).$ 

lemma prod-Icc-sq-sub:
  ( $\prod_{j=1}^N. - (x^2) + \text{of-nat } j^2$ ) = ( $\prod_{k < N. (\text{of-nat } N - x - \text{of-nat } k) * (x + \text{of-nat } N - \text{of-nat } k) :: \text{rat}$ )

```

proof –
have $(\prod j = 1..N. - (x \wedge 2) + \text{of-nat } j \wedge 2) = (\prod k < N. - (x \wedge 2) + \text{of-nat } (N - k) \wedge 2)$
by $(\text{rule prod.reindex-bij-witness}[\text{where } i = \lambda k. N - k \text{ and } j = \lambda j. N - j]) \text{ auto}$
also have $\dots = (\prod k < N. (\text{of-nat } N - x - \text{of-nat } k) * (x + \text{of-nat } N - \text{of-nat } k))$
by $(\text{intro prod.cong refl}) (\text{simp add: of-nat-diff power2-eq-square algebra-simps})$
finally show *?thesis* .
qed

$$D-N(-x \wedge 2)/(N!) \wedge 2 = \text{binom}(N - x, N) \text{binom}(N + x, N).$$

lemma *Dm-pcompose-eq*:

$$\text{Polynomial.smult } (1 / \text{fact } N \wedge 2) (\text{pcompose } (Dm \ N) (- (pX \wedge 2))) = \text{pcompose } (bp \ N) [: \text{of-nat } N, - 1:] * \text{pcompose } (bp \ N) [: \text{of-nat } N, 1:]$$

proof –
have $l: \text{poly } (Dm \ N) \ y = (\prod j = 1..N. y + \text{of-nat } j \wedge 2) \text{ for } y$
unfolding *Dm-def poly-prod* **by** $(\text{rule prod.cong}) (\text{simp-all add: poly-monom})$
have $e: \text{poly } (\text{Polynomial.smult } (1 / \text{fact } N \wedge 2) (\text{pcompose } (Dm \ N) (- (pX \wedge 2)))) \ x =$
 $\text{poly } (\text{pcompose } (bp \ N) [: \text{of-nat } N, - 1:] * \text{pcompose } (bp \ N) [: \text{of-nat } N, 1:]) \ x$
for $x :: \text{rat}$
proof –
have $r: (\prod k < N. \text{of-nat } N + x - \text{of-nat } k) = (\prod k < N. x + \text{of-nat } N - \text{of-nat } k)$
by $(\text{simp add: add.commute})$
have $\text{poly } (\text{Polynomial.smult } (1 / \text{fact } N \wedge 2) (\text{pcompose } (Dm \ N) (- (pX \wedge 2)))) \ x =$
 $1 / \text{fact } N \wedge 2 * \text{poly } (Dm \ N) (- (x \wedge 2))$
by $(\text{simp add: poly-pcompose poly-monom})$
also have $\dots = 1 / \text{fact } N \wedge 2 * (\prod k < N. (\text{of-nat } N - x - \text{of-nat } k) * (x + \text{of-nat } N - \text{of-nat } k))$
unfolding *l prod-Icc-sq-sub* ..
also have $\dots = (\prod k < N. \text{of-nat } N - x - \text{of-nat } k) / \text{fact } N * ((\prod k < N. x + \text{of-nat } N - \text{of-nat } k) / \text{fact } N)$
by $(\text{simp add: prod.distrib power2-eq-square})$
also have $\dots = \text{poly } (bp \ N) (\text{of-nat } N - x) * \text{poly } (bp \ N) (\text{of-nat } N + x)$
unfolding *poly-bp r* ..
also have $\dots = \text{poly } (\text{pcompose } (bp \ N) [: \text{of-nat } N, - 1:] * \text{pcompose } (bp \ N) [: \text{of-nat } N, 1:]) \ x$
by $(\text{simp add: poly-pcompose})$
finally show *?thesis* .
qed
show *?thesis* **by** $(\text{rule poly-eq-of-eval-nat}) (\text{rule } e)$
qed

lemma *IntEval-Dm-pcompose*: $\text{IntEval } (\text{Polynomial.smult } (1 / \text{fact } N \wedge 2) (\text{pcompose } (Dm \ N) (- (pX \wedge 2))))$

unfolding *Dm-pcompose-eq*
by (*intro IntEval-mult IntEval-bp-pcompose*[*of N int N - 1*,
simplified] *IntEval-bp-pcompose*[*of N int N 1*, *simplified*])

lemma *E1-eq*:

$$\left(\prod_{k < 2 * m + 2} x + (of\text{-}nat\ m + 1) - of\text{-}nat\ k\right) = (x + of\text{-}nat\ m + 1) * x$$

$$* \left(\prod_{j = 1..m} x^2 - of\text{-}nat\ j^2 :: rat\right)$$

proof (*induction m*)
case 0
show ?*case* **by** (*simp add: numeral-2-eq-2 algebra-simps*)
next
case (*Suc m*)
have *e*: $2 * Suc\ m + 2 = Suc\ (Suc\ (2 * m + 2))$ **by** *simp*
have *mid*: $\left(\prod_{i < 2 * m + 2} x + (of\text{-}nat\ (Suc\ m) + 1) - of\text{-}nat\ (Suc\ i)\right) =$
 $\left(\prod_{k < 2 * m + 2} x + (of\text{-}nat\ m + 1) - of\text{-}nat\ k\right)$
by (*rule prod.cong[OF refl]*) (*simp add: algebra-simps*)
show ?*case*
by (*subst e, subst prod.lessThan-Suc, subst prod.lessThan-Suc-shift,*
unfold mid Suc.IH prod.cl-ivl-Suc)
(simp add: algebra-simps power2-eq-square)
qed

lemma *E2-eq*:

$$\left(\prod_{k < 2 * m + 2} x + of\text{-}nat\ m - of\text{-}nat\ k\right) = x * (x - of\text{-}nat\ m - 1) * \left(\prod_{j = 1..m} x^2 - of\text{-}nat\ j^2 :: rat\right)$$

proof (*induction m*)
case 0
show ?*case* **by** (*simp add: numeral-2-eq-2 algebra-simps*)
next
case (*Suc m*)
have *e*: $2 * Suc\ m + 2 = Suc\ (Suc\ (2 * m + 2))$ **by** *simp*
have *mid*: $\left(\prod_{i < 2 * m + 2} x + of\text{-}nat\ (Suc\ m) - of\text{-}nat\ (Suc\ i)\right) =$
 $\left(\prod_{k < 2 * m + 2} x + of\text{-}nat\ m - of\text{-}nat\ k\right)$
by (*rule prod.cong[OF refl]*) (*simp add: algebra-simps*)
show ?*case*
by (*subst e, subst prod.lessThan-Suc, subst prod.lessThan-Suc-shift,*
unfold mid Suc.IH prod.cl-ivl-Suc)
(simp add: algebra-simps power2-eq-square)
qed

$q\text{-}\{m+1\}(-x^2) = binom(x + m + 1, 2m + 2) + binom(x + m, 2m + 2).$

lemma *qBasis-pcompose-eq*:

$$pcompose\ (qBasis\ (Suc\ m))\ (-\ (pX^2)) =$$

$$pcompose\ (bp\ (2 * m + 2))\ [:of\text{-}nat\ m + 1, 1:] + pcompose\ (bp\ (2 * m + 2))$$

$$[:of\text{-}nat\ m, 1:]$$

proof –

```

have l: poly (Dm m) y = (∏ j = 1..m. y + of-nat j ^ 2) for y
  unfolding Dm-def poly-prod by (rule prod.cong) (simp-all add: poly-monom)
have e: poly (pcompose (qBasis (Suc m)) (-(pX ^ 2))) x =
  poly (pcompose (bp (2 * m + 2)) [:of-nat m + 1, 1:] + pcompose (bp (2 * m
+ 2)) [:of-nat m,
  1:]) x for x :: rat
proof -
  define F :: rat where F = fact (2 * Suc m)
  define P :: rat where P = (∏ j = 1..m. x ^ 2 - of-nat j ^ 2)
  have F2: (fact (2 * m + 2) :: rat) = F
  proof -
    have 2 * m + 2 = 2 * Suc m by simp
    thus ?thesis unfolding F-def by simp
  qed
  have q: poly (qBasis (Suc m)) y = (-1) ^ Suc m * 2 / F * y * poly (Dm m)
y for y
  unfolding F-def by (simp add: qBasis-def poly-monom)
  have neg: (∏ j = 1..m. -(x ^ 2) + of-nat j ^ 2) = (-1) ^ m * P
  proof -
    have (∏ j = 1..m. -(x ^ 2) + of-nat j ^ 2) = (∏ j =
      1..m. (-1) * (x ^ 2 - of-nat j ^ 2))
    by (rule prod.cong) simp-all
    also have ... = (-1) ^ m * P unfolding P-def by (subst prod.distrib) simp
    finally show ?thesis .
  qed
  have z: poly (-(pX ^ 2)) x = -(x ^ 2) by (simp add: poly-monom)
  have y1: poly [:of-nat m + 1, 1:] x = x + (of-nat m + 1) by simp
  have y2: poly [:of-nat m, 1:] x = x + of-nat m by simp
  have key: (x + of-nat m + 1) * x * P + x * (x - of-nat m - 1) * P = 2 * x
^ 2 * P
  by (simp add: algebra-simps power2-eq-square)
  have L: (-1) ^ Suc m * 2 / F * (-(x ^ 2)) * ((-1) ^ m * P) = 2 * x ^ 2
* P / F
  by (cases even m) (simp-all add: neg-one-even-power neg-one-odd-power)
  have poly (pcompose (qBasis (Suc m)) (-(pX ^ 2))) x =
    (-1) ^ Suc m * 2 / F * (-(x ^ 2)) * ((-1) ^ m * P)
  unfolding poly-pcompose z q l neg ..
  also have ... = 2 * x ^ 2 * P / F by (rule L)
  also have ... = (x + of-nat m + 1) * x * P / F + x * (x - of-nat m - 1) *
P / F
  unfolding add-divide-distrib[symmetric] key ..
  also have ... = poly (pcompose (bp (2 * m + 2)) [:of-nat m + 1,
    1:] + pcompose (bp (2 * m + 2)) [:of-nat m, 1:]) x
  unfolding poly-add poly-pcompose y1 y2 poly-bp E1-eq E2-eq P-def[symmetric]
F2 ..
  finally show ?thesis .
qed
show ?thesis by (rule poly-eq-of-eval-nat) (rule e)
qed

```

```

lemma IntEval-qBasis: IntEval (pcompose (qBasis i) (− (pX ^ 2)))
proof (cases i)
  case 0
    have IntEval 1 unfolding IntEval-def by (auto intro!: exI[of - 1])
    thus ?thesis using 0 by (simp add: qBasis-def pcompose-1)
  next
    case (Suc m)
    have a: IntEval (pcompose (bp (2 * m + 2)) [:of-nat m + 1, 1:])
    using IntEval-bp-pcompose[of 2 * m + 2 int m + 1 1] by simp
    have b: IntEval (pcompose (bp (2 * m + 2)) [:of-nat m, 1:])
    using IntEval-bp-pcompose[of 2 * m + 2 int m 1] by simp
    show ?thesis unfolding Suc qBasis-pcompose-eq by (rule IntEval-add[OF a b])
  qed

lemma IntEval-fBasis: IntEval (pcompose (fBasis K i) (− (pX ^ 2)))
proof −
  have e: pcompose (fBasis K i) (− (pX ^ 2)) =
    (Polynomial.smult (1 / fact (Nof K) ^ 2) (pcompose (Dm (Nof K)) (− (pX
    ^ 2)))) ^ 3 *
    pcompose (qBasis i) (− (pX ^ 2))
    unfolding fBasis-def pcompose-mult pcompose-power-left by (simp add: pcom-
pose-smult)
  show ?thesis unfolding e by (intro IntEval-mult IntEval-power IntEval-Dm-pcompose
IntEval-qBasis)
qed

f-i(−x^2) is integer-valued (binomial identities  $q-i(-x^2) = C(x+i, 2i) + C(x+i-1, 2i)$ ,  $D-N(-x^2)/(N!)^2 = C(N-x, N) C(N+x, N)$ ).

theorem fBasis-intValued: prime p  $\implies$ 
  IntValuedAt p (pcompose (fBasis K i) (− (pX ^ 2)))
  by (rule IntEval-IntValuedAt, assumption, rule IntEval-fBasis)

```

18.2 The determinant identity (3.11)

```

lemma degree-Dm: degree (Dm m) = m
proof −
  have degree (Dm m) = ( $\sum j = 1..m. \text{degree } (pX + [:of-nat j^2:])$ )
  unfolding Dm-def by (rule degree-prod-eq-sum-degree) (use pX-plus-const-nonzero
  in blast)
  also have ... = ( $\sum j = 1..m. 1$ ) by (rule sum.cong[OF refl]) (rule degree-pX-plus-const)
  also have ... = m by simp
  finally show ?thesis .
qed

lemma lead-coeff-Dm: lead-coeff (Dm m) = 1
proof −
  have l: lead-coeff (pX + [:of-nat j ^ 2:]) = 1 for j :: nat
  unfolding degree-pX-plus-const by (simp add: coeff-monom)

```

show *?thesis* **unfolding** *Dm-def lead-coeff-prod l* **by** *simp*
qed

lemma *Dm-nonzero*: *Dm m* $\neq 0$
using *lead-coeff-Dm[of m]* **by** *auto*

The leading coefficient $(-1)^i 2 / (2i)!$ of *q-i* (and 1 for $i = 0$).

definition *qLead* :: *nat* $\Rightarrow$ *rat* **where**
qLead i = (if $i = 0$ then 1 else $(-1)^i 2 / \text{fact } (2 * i)$)

lemma *degree-qBasis-le*: *degree (qBasis i)* $\leq i$

proof (*cases i*)

case 0

thus *?thesis* **by** (*simp add: qBasis-def*)

next

case (*Suc m*)

have *q: qBasis i* = $[:(-1)^i 2 / \text{fact } (2 * i):] * pX * Dm m$ **unfolding**
qBasis-def **using** *Suc*

by *simp*

have *degree* ($[:(-1)^i 2 / \text{fact } (2 * i):] * pX * Dm m$) $\leq$
degree ($[:(-1)^i 2 / \text{fact } (2 * i):] * pX$) + *degree* (*Dm m*) **by** (*rule*
degree-mult-le)

moreover **have** *degree* ($[:(-1)^i 2 / \text{fact } (2 * i):] * pX$) ≤ 1

by (*simp add: degree-monom-le*)

ultimately show *?thesis* **unfolding** *q* **using** *degree-Dm[of m]* *Suc* **by** *linarith*
qed

lemma *coeff-qBasis-self*: *Polynomial.coeff (qBasis i) i* = *qLead i*

proof (*cases i*)

case 0

thus *?thesis* **by** (*simp add: qBasis-def qLead-def*)

next

case (*Suc m*)

have *Polynomial.coeff (Dm m) m* = 1 **using** *lead-coeff-Dm[of m]* **unfolding**
degree-Dm .

hence *Polynomial.coeff (Polynomial.monom a (Suc 0) * Dm m) (Suc m)* = *a*
for *a*

by (*simp add: coeff-monom-mult*)

with *Suc* **show** *?thesis* **unfolding** *qBasis-def qLead-def* **by** *simp*
qed

lemma *qBasis-eq-sum*:

$i < h \implies qBasis i = (\sum k <$

$h. \text{Polynomial.smult } (\text{Polynomial.coeff } (qBasis i) k) (pX \wedge k))$

proof –

assume *ih*: $i < h$

have *d*: *degree (qBasis i)* $\leq h - 1$ **using** *degree-qBasis-le[of i]* *ih* **by** *linarith*

have *e*: $\{..h - 1\} = \{..<h\}$ **using** *ih* **by** *auto*

have *qBasis i* = $(\sum k \leq h - 1. \text{Polynomial.monom } (\text{Polynomial.coeff } (qBasis$

```

i) k) k)
  by (rule sym[OF poly-as-sum-of-monoms'[OF d]])
  also have ... = ( $\sum k < h. \text{Polynomial.monom } (\text{Polynomial.coeff } (qBasis i) k)$ 
k)
  unfolding e ..
  also have ... = ( $\sum k < h. \text{Polynomial.smult } (\text{Polynomial.coeff } (qBasis i) k)$ 
( $pX \wedge k$ ))
  by (rule sum.cong[OF refl]) (unfold pX-pow-monom smult-monom, simp)
  finally show ?thesis .
qed

```

The coefficient matrix of the q - i , lower triangular.

definition $Qc :: nat \Rightarrow rat\ mat$ **where**

$Qc\ h = \text{Matrix.mat } h\ h\ (\lambda(i, k). \text{Polynomial.coeff } (qBasis i) k)$

lemma $Qc\text{-carrier}$: $Qc\ h \in \text{carrier-mat } h\ h$

by (simp add: $Qc\text{-def}$)

lemma $det\text{-}Qc$: $\text{Determinant.det } (Qc\ h) = (\prod i < h. qLead\ i)$

proof –

have $\text{Determinant.det } (Qc\ h) = \text{prod-list } (\text{diag-mat } (Qc\ h))$

proof (rule $det\text{-lower-triangular}$ [OF - $Qc\text{-carrier}$])

fix $i\ j$ **assume** ij : $i < j$ **and** jh : $j < h$

have $\text{degree } (qBasis\ i) < j$ **using** $\text{degree-}qBasis\text{-le}$ [of i] ij **by** linarith

thus $Qc\ h\ \$\$ (i, j) = 0$ **using** $ij\ jh$ **by** (simp add: $Qc\text{-def coeff-eq-0}$)

qed

also have ... = ($\prod i = 0..<dim\text{-row } (Qc\ h). Qc\ h\ \$\$ (i, i)$)

by (rule $\text{prod-list-diag-prod}$)

also have ... = ($\prod i \in \{..<h\}. qLead\ i$)

by (rule prod.cong) (auto simp: $Qc\text{-def coeff-}qBasis\text{-self}$)

finally show ?thesis .

qed

lemma $qLead\text{-sq-prod}$: ($\prod i < h. qLead\ i$) $\wedge 2 = 4 \wedge (h - 1) / (\prod i =$
 $1..h - 1. \text{fact } (2 * i) \wedge 2)$

proof (cases h)

case 0

thus ?thesis **by** simp

next

case ($Suc\ m$)

have $sq: qLead\ (Suc\ i) \wedge 2 = 4 / \text{fact } (2 * Suc\ i) \wedge 2$ **for** i

by (cases even i) (simp-all add: $qLead\text{-def power-divide neg-one-even-power neg-one-odd-power}$)

have r : ($\prod i < m. \text{fact } (2 * Suc\ i) \wedge 2$) = ($\prod i = 1..m. \text{fact } (2 * i) \wedge 2 :: rat$)
using $\text{prod.atLeast1-atMost-eq}$ [of $\lambda i. \text{fact } (2 * i) \wedge 2 :: rat\ m$] **by** simp

have ($\prod i < h. qLead\ i$) $\wedge 2 = (\prod i < m. qLead\ (Suc\ i)) \wedge 2$

unfolding $Suc\ \text{prod.lessThan-Suc-shift}$ **by** (simp add: $qLead\text{-def}$)

also have ... = ($\prod i < m. qLead\ (Suc\ i) \wedge 2$) **by** (rule $\text{prod-power-distrib}$)

also have ... = ($\prod i < m. 4 / \text{fact } (2 * Suc\ i) \wedge 2$) **unfolding** sq ..

also have $\dots = 4^m / (\prod i < m. \text{fact } (2 * \text{Suc } i) ^ 2)$ **by** (*simp add: prod-dividef*)
also have $\dots = 4^m / (\prod i = 1..m. \text{fact } (2 * i) ^ 2)$ **unfolding** *r*..
finally show *?thesis* **using** *Suc* **by** *simp*
qed

lemma *fBasis-mult*:

$fBasis\ K\ i * fBasis\ K\ j = \text{Polynomial.smult } ((1 / \text{fact } (\text{Nof } K) ^ 2) ^ 6) (Dm$
 $(\text{Nof } K) ^ 6 * (qBasis\ i * qBasis\ j))$

proof –

define *c* :: *rat* **where** $c = 1 / \text{fact } (\text{Nof } K) ^ 2$

define *D* **where** $D = Dm\ (\text{Nof } K)$

have *e1*: $c ^ 6 = c ^ 3 * c ^ 3$ **and** *e2*: $D ^ 6 = D ^ 3 * D ^ 3$

by (*simp-all add: power-add[symmetric]*)

show *?thesis* **unfolding** *fBasis-def c-def[symmetric] D-def[symmetric]* *e1 e2*

by (*simp add: smult-power mult-ac*)

qed

lemma *muX-fBasis*:

assumes $i < \text{hof } K\ j < \text{hof } K$

shows $\text{muX } \{1..K\} (fBasis\ K\ i * fBasis\ K\ j) =$

$\text{Polynomial.smult } ((1 / \text{fact } (\text{Nof } K) ^ 2) ^ 6)$

$(\sum k < \text{hof } K. \sum l <$

$\text{hof } K. [:Qc\ (\text{hof } K) \$\$ (i, k) * Qc\ (\text{hof } K) \$\$ (j, l):] * GK\ K \$\$ (k, l))$

proof –

let *?h* = $\text{hof } K$ **and** *?c* = $(1 / \text{fact } (\text{Nof } K) ^ 2) ^ 6$:: *rat* **and** *?D* = $Dm\ (\text{Nof } K) ^ 6$

let *?Q* = $Qc\ (\text{hof } K)$

have *sm*: $\text{muX } S\ (\text{Polynomial.smult } a\ P) = \text{Polynomial.smult } a\ (\text{muX } S\ P)$ **for** $S\ a\ P$

using *muX-C-mul[of S a P]* **by** *simp*

have *qe*: $qBasis\ a = (\sum k < ?h. \text{Polynomial.smult } (?Q \$\$ (a, k)) (pX ^ k))$ **if** $a < ?h$ **for** *a*

proof –

have $qBasis\ a = (\sum k < ?h. \text{Polynomial.smult } (\text{Polynomial.coeff } (qBasis\ a)\ k) (pX ^ k))$

by (*rule qBasis-eq-sum[OF that]*)

also have $\dots = (\sum k < ?h. \text{Polynomial.smult } (?Q \$\$ (a, k)) (pX ^ k))$

using *that* **by** (*intro sum.cong refl*) (*simp add: Qc-def*)

finally show *?thesis* .

qed

have *pq0*: $qBasis\ i * qBasis\ j = (\sum k < ?h. \sum l < ?h.$

$\text{Polynomial.smult } (?Q \$\$ (i, k)) (pX ^ k) * \text{Polynomial.smult } (?Q \$\$ (j, l)) (pX ^ l))$

unfolding *qe[OF assms(1)] qe[OF assms(2)]* **by** (*rule sum-product*)

have *pq*: $?D * (qBasis\ i * qBasis\ j) =$

$(\sum k < ?h. \sum l < ?h. \text{Polynomial.smult } (?Q \$\$ (i, k) * ?Q \$\$ (j, l)) (?D * pX ^ (k + l)))$

unfolding *pq0 sum-distrib-left*
by (*intro sum.cong refl*) (*simp add: power-add mult-ac*)
have $\text{muX } \{1..K\} (fBasis\ K\ i * fBasis\ K\ j) =$
 $Polynomial.smult\ ?c\ (\text{muX } \{1..K\} (?D * (qBasis\ i * qBasis\ j)))$
unfolding *fBasis-mult sm ..*
also have $\dots = Polynomial.smult\ ?c\ (\sum k < ?h. \sum l <$
 $?h. [:?Q\ \$\$ (i, k) * ?Q\ \$\$ (j, l):] * GK\ K\ \$\$ (k, l))$
unfolding *pq muX-sum sm*
by (*intro arg-cong[where f = Polynomial.smult ?c] sum.cong refl*) (*simp add:*
GK-def)
finally show *?thesis .*
qed

lemma *SK-eq-lead:*
 $SK\ K = (fact\ K \wedge 2 * (1 / fact\ (Nof\ K) \wedge 2) \wedge 6) \wedge hof\ K * (\prod i < hof\ K.$
 $qLead\ i) \wedge 2$
unfolding *SK-def qLead-sq-prod*
by (*simp add: power-mult-distrib power-one-over power-divide power-mult[symmetric]*
mult-ac)

The matrix of (3.11) is $c \cdot Q\ G\ Q^T$ with $c = (K!) \wedge 2 / (N!) \wedge \{12\}$.

lemma *fBasis-matrix-eq:*
 $Matrix.mat\ (hof\ K)\ (hof\ K)\ (\lambda(i,$
 $j). [:fact\ K \wedge 2:] * \text{muX } \{1..K\} (fBasis\ K\ i * fBasis\ K\ j)) =$
 $[:fact\ K \wedge 2 * (1 / fact\ (Nof\ K) \wedge 2) \wedge 6:] \cdot_m$
 $(map\ mat\ (\lambda q. [:q:]) (Qc\ (hof\ K)) * GK\ K * transpose\ mat\ (map\ mat\ (\lambda q. [:q:]) (Qc\ (hof\ K))))$
proof –
let $?h = hof\ K$ **and** $?c = fact\ K \wedge 2 * (1 / fact\ (Nof\ K) \wedge 2) \wedge 6 :: rat$
let $?Q = map\ mat\ (\lambda q. [:q:]) (Qc\ (hof\ K))$
have $dQ: dim\ row\ (Qc\ n) = n\ dim\ col\ (Qc\ n) = n$ **for** n **by** (*simp-all add:*
Qc-def)
have $dG: dim\ row\ (GK\ K) = ?h\ dim\ col\ (GK\ K) = ?h$ **by** (*simp-all add: GK-def*)
have $entry: [:fact\ K \wedge 2:] * \text{muX } \{1..K\} (fBasis\ K\ i * fBasis\ K\ j) =$
 $[:?c:] * (?Q * GK\ K * transpose\ mat\ ?Q)\ \$\$ (i, j)$
if $ih: i < ?h$ **and** $jh: j < ?h$ **for** $i\ j$
proof –
have $m: (?Q * GK\ K * transpose\ mat\ ?Q)\ \$\$ (i, j) =$
 $(\sum l < ?h. (\sum k < ?h. [:Qc\ ?h\ \$\$ (i, k):] * GK\ K\ \$\$ (k, l)) * [:Qc\ ?h\ \$\$$
 $(j,$
 $l):])$
using $ih\ jh$ **by** (*simp add: scalar-prod-def dQ dG atLeast0LessThan*)
have $[:?c:] * (\sum l < ?h. (\sum k <$
 $?h. [:Qc\ ?h\ \$\$ (i, k):] * GK\ K\ \$\$ (k, l)) * [:Qc\ ?h\ \$\$ (j, l):]) =$
 $(\sum l < ?h. \sum k < ?h. [:?c:] * ([:Qc\ ?h\ \$\$ (i, k):] * GK\ K\ \$\$ (k,$
 $l) * [:Qc\ ?h\ \$\$ (j, l):]))$
by (*simp add: sum-distrib-left sum-distrib-right smult-sum-right*)
also have $\dots = (\sum k < ?h. \sum l <$
 $?h. [:?c:] * ([:Qc\ ?h\ \$\$ (i, k):] * GK\ K\ \$\$ (k, l) * [:Qc\ ?h\ \$\$ (j, l):]))$

```

    by (rule sum.swap)
  also have ... = [:fact K ^ 2:] * muX {1..K} (fBasis K i * fBasis K j)
    unfolding muX-fBasis[OF ih jh] by (simp add: smult-sum-right mult-ac)
  finally show ?thesis unfolding m by (rule sym)
qed
show ?thesis
proof (rule eq-matI)
  fix i j
  assume i: i < dim-row ([:?c:] ·m (?Q * GK K * transpose-mat ?Q))
    and j: j < dim-col ([:?c:] ·m (?Q * GK K * transpose-mat ?Q))
  have ih: i < ?h and jh: j < ?h using i j by (simp-all add: dQ dG)
  have a: i < dim-row (?Q * GK K * transpose-mat ?Q) j <
    dim-col (?Q * GK K * transpose-mat ?Q)
    using ih jh by (simp-all add: dQ dG)
  show Matrix.mat ?h ?h (λ(i,
    j). [:fact K ^ 2:] * muX {1..K} (fBasis K i * fBasis K j)) $$ (i, j) =
    ([:?c:] ·m (?Q * GK K * transpose-mat ?Q)) $$ (i, j)
    unfolding index-mat(1)[OF ih jh] index-smult-mat(1)[OF a] prod.case by
(rule entry[OF ih jh])
  qed (simp-all add: dQ dG)
qed

```

lemma *comm-ring-hom-pCons*: *comm-ring-hom* (λq :: 'a :: comm-ring-1. [:q:])
 by *unfold-locales simp-all*

lemma *det-map-mat-C*:
 $M \in \text{carrier-mat } n \implies$
 $\text{Determinant.det } (\text{map-mat } (\lambda q. [:q:]) M) = [: \text{Determinant.det } M:]$
 by (rule *comm-ring-hom.hom-det*[OF *comm-ring-hom-pCons*])

(3.11): $F \cdot K = \det[(K!)^{\wedge 2} \mu\text{-}X(f\text{-}i \text{ } f\text{-}j / D \cdot K)]$.

theorem *F-eq-det-fBasis*:
 assumes $40 \text{ dvd } K$
 shows $FK K = \text{Determinant.det } (\text{Matrix.mat } (\text{hof } K) (\text{hof } K)$
 $(\lambda(i, j). [:fact K ^ 2:] * \mu X \{1..K\} (fBasis K i * fBasis K j)))$

proof –
 define h where $h = \text{hof } K$
 define $c :: \text{rat}$ where $c = \text{fact } K ^ 2 * (1 / \text{fact } (\text{Nof } K) ^ 2) ^ 6$
 define Qp where $Qp = \text{map-mat } (\lambda q. [:q:]) (Qc \text{ } h)$
 have $Qpc: Qp \in \text{carrier-mat } h \text{ } h$ and $Gc: GK K \in \text{carrier-mat } h \text{ } h$
 by (simp-all add: *Qp-def Qc-def GK-def h-def*)
 have $\text{Determinant.det } (\text{Matrix.mat } h \text{ } h (\lambda(i,$
 $j). [:fact K ^ 2:] * \mu X \{1..K\} (fBasis K i * fBasis K j)))$
 $= [:c:] ^ h * (\text{Determinant.det } Qp * \text{Determinant.det } (GK K) * \text{Determinant.det } Qp)$
proof –
 have $e: \text{Matrix.mat } h \text{ } h (\lambda(i,$
 $j). [:fact K ^ 2:] * \mu X \{1..K\} (fBasis K i * fBasis K j)) =$
 $[:c:] \cdot_m (Qp * GK K * \text{transpose-mat } Qp)$


```

    unfolding h-def c-def Qp-def by (rule fBasis-matrix-eq)
    have Tc: transpose-mat Qp ∈ carrier-mat h h using Qpc by simp
    have QGc: Qp * GK K ∈ carrier-mat h h using Qpc Gc by (rule mult-carrier-mat)
    have dc: dim-col (Qp * GK K * transpose-mat Qp) = h using Qpc by simp
    show ?thesis unfolding e det-smult dc det-mult[OF QGc Tc] det-mult[OF Qpc
Gc] det-transpose[OF
    Qpc] ..
qed
also have ... = [:c ^ h * (∏ i < h. qLead i) ^ 2:] * DeltaK K
    unfolding Qp-def det-map-mat-C[OF Qc-carrier] det-Qc DeltaK-def const-poly-power
    by (simp add: power2-eq-square mult-ac)
also have ... = FK K
    unfolding FK-def SK-eq-lead c-def h-def ..
finally show ?thesis unfolding h-def ..
qed

```

18.3 The small-prime bound (3.12)

```

lemma degree-fBasis-le: degree (fBasis K i) ≤ 3 * Nof K + i
proof -
  have degree ([:1 / fact (Nof K) ^ 2:] * Dm (Nof K)) ≤ Nof K
    using degree-mult-le[of [:1 / fact (Nof K) ^ 2:] Dm (Nof K)] degree-Dm[of Nof
K] by simp
  with degree-power-le[of [:1 / fact (Nof K) ^ 2:] * Dm (Nof K) 3]
  have degree (([:1 / fact (Nof K) ^ 2:] * Dm (Nof K)) ^ 3) ≤ 3 * Nof K by
linarith
  moreover have degree (([:1 / fact (Nof K) ^ 2:] * Dm (Nof K)) ^ 3 * qBasis
i) ≤
    degree (([:1 / fact (Nof K) ^ 2:] * Dm (Nof K)) ^ 3) + degree (qBasis i)
    by (rule degree-mult-le)
  ultimately show ?thesis unfolding fBasis-def using degree-qBasis-le[of i] by
linarith
qed

```

```

lemma signedPoles-Icc: signedPoles {1..K} = symPoles K
unfolding signedPoles-def symPoles-def
proof
  show int ' {1..K} ∪ (λj. - int j) ' {1..K} ⊆ {r. - int K ≤ r ∧
r ≤ int K ∧ r ≠ 0} by auto
next
  show {r. - int K ≤ r ∧ r ≤ int K ∧
r ≠ 0} ⊆ int ' {1..K} ∪ (λj. - int j) ' {1..K}
proof
  fix r assume r: r ∈ {r. - int K ≤ r ∧ r ≤ int K ∧ r ≠ 0}
  show r ∈ int ' {1..K} ∪ (λj. - int j) ' {1..K}
proof (cases 0 < r)
  case True
  hence r = int (nat r) nat r ∈ {1..K} using r by auto
  thus ?thesis by blast

```

```

next
  case False
  hence  $r = - \text{int} (\text{nat} (- r)) \text{ nat} (- r) \in \{1..K\}$  using r by auto
  thus ?thesis by blast
qed
qed
qed

```

Each entry of the matrix in (3.11) satisfies the Lemma 3.3 bound.

lemma *fBasis-entry-vge*:

```

  assumes p: prime p and K:  $40 \text{ dvd } K$  and i:  $i < \text{hof } K$  and j:  $j < \text{hof } K$ 
  shows  $\text{vpGge } p \text{ } ([:\text{fact } K \wedge 2:] * \text{muX } \{1..K\} (\text{fBasis } K \text{ } i * \text{fBasis } K \text{ } j))$ 
     $(- 6 * \text{of-nat} (\text{nat-log } p (5 * K)) - \text{of-nat} (\text{multiplicity } p (24::\text{nat})))$ 
proof -
  obtain n where Kn:  $K = 40 * n$  using K by (elim dvdE)
  have n1:  $1 \leq n$  using i by (auto simp: Kn hof-def)
  have hN:  $\text{Nof } K = 3 * n$  and hh:  $\text{hof } K = 37 * n$  by (simp-all add: Kn Nof-def hof-def)
  define A where  $A = [:(-1) \wedge \text{card } \{1..K\}:] * pX \wedge 5 * \text{pcompose} (\text{fBasis } K \text{ } i * \text{fBasis } K \text{ } j) (- (pX \wedge 2))$ 
  have pb:  $[:\text{fact } K \wedge 2:] * \text{muX } \{1..K\} (\text{fBasis } K \text{ } i * \text{fBasis } K \text{ } j) = \text{tauX } (\text{symPoles } K) ([:\text{fact } K \wedge 2:] * A)$ 
  proof -
    have  $\text{muX } \{1..K\} (\text{fBasis } K \text{ } i * \text{fBasis } K \text{ } j) = \text{tauX } (\text{symPoles } K) A$ 
    unfolding A-def signedPoles-Icc[symmetric] by (rule pullback) auto
    thus ?thesis unfolding tauX-C-mul by simp
  qed
  have Aint: IntValuedAt p A
  proof -
    have c: IntEval  $[:(-1) \wedge \text{card } \{1..K\}: :: \text{rat}]$  using IntEval-C-int[of  $(-1) \wedge \text{card } \{1..K\}$ ]
    by simp
    have IntEval A unfolding A-def pcompose-mult
    by (intro IntEval-mult c IntEval-power IntEval-pX IntEval-fBasis)
    thus ?thesis by (rule IntEval-IntValuedAt[OF p])
  qed
  have dA:  $\text{degree } A \leq 5 + 184 * n$ 
  proof -
    have d1:  $\text{degree} (\text{fBasis } K \text{ } i * \text{fBasis } K \text{ } j) \leq 92 * n$ 
    using degree-mult-le[of fBasis K i fBasis K j] degree-fBasis-le[of K i]
    degree-fBasis-le[of K j] i j hN hh
    by linarith
    have d2:  $\text{degree} (- (pX \wedge 2) :: \text{rat poly}) = 2$  unfolding degree-minus pX-pow-monom
    by (rule degree-monom-eq) simp
    have  $\text{degree} (\text{pcompose} (\text{fBasis } K \text{ } i * \text{fBasis } K \text{ } j) (- (pX \wedge 2))) \leq 184 * n$ 
    unfolding degree-pcompose d2 using d1 by linarith
    moreover have  $\text{degree} ([:(-1) \wedge \text{card } \{1..K\}:] * pX \wedge 5 :: \text{rat poly}) \leq 5$ 
    unfolding pX-pow-monom by (simp add: degree-monom-le)
    ultimately show ?thesis unfolding A-def

```

```

using degree-mult-le[of [:(-1) ^ card {1..K}]:] * pX ^ 5 pcompose (fBasis K
i *
  fBasis K j) (- (pX ^ 2))]
by linarith
qed
have log: nat-log p (max (2 * K) (5 + 184 * n + 1)) ≤ nat-log p (5 * K)
by (rule nat-log-mono) (use n1 in ⟨simp add: Kn⟩)
show ?thesis
unfolding pb
by (rule vpGge-mono[OF far-pole-valuation-bound[OF p Aint dA]]) (use log in
⟨simp add:
  of-nat-le-iff⟩)
qed

(3.12):  $v\text{-}p^{\wedge}G(F\text{-}K) \geq -6h \lfloor \log\text{-}p(5K) \rfloor - h \text{ } v\text{-}p(24)$  for every prime  $p$ .

theorem vpG-F-ge:
  assumes p: prime p and K: 40 dvd K
  shows vpGge p (FK K) (- 6 * of-nat (hof K) * of-nat (nat-log p (5 * K))
    - of-nat (hof K) * of-nat (multiplicity p (24::nat)))

proof -
  define h where h = hof K
  define M where M = Matrix.mat h h (λ(i,
    j). [fact K ^ 2:] * muX {1..K} (fBasis K i * fBasis K j))
  define b where b = (- 6 * of-nat (nat-log p (5 * K)) - of-nat (multiplicity p
    (24::nat))) :: rat)
  have Mc: M ∈ carrier-mat h h by (simp add: M-def)
  have entry: ∧ i j. i < h ⇒ j < h ⇒ vpGge p (M $$ (i, j)) b
  proof -
    fix i j assume ih: i < h and jh: j < h
    have e: M $$ (i, j) = [fact K ^ 2:] * muX {1..K} (fBasis K i * fBasis K j)
    unfolding M-def using ih jh by (subst index-mat) simp-all
    show vpGge p (M $$ (i, j)) b
    unfolding e b-def by (rule fBasis-entry-vge[OF p K]) (use ih jh in ⟨simp-all
add:
      h-def⟩)
  qed
  have vpGge p (Determinant.det M) (∑ i = 0..h. b)
  proof (rule vpGge-det-of-terms[OF p Mc])
    fix σ assume σ permutes {0..h}
    thus vpGge p (∏ i = 0..h. M $$ (i, σ i)) (∑ i = 0..h. b)
    by (intro vpGge-prod[OF p] entry) (auto simp: permutes-in-image)
  qed
  moreover have FK K = Determinant.det M unfolding M-def h-def by (rule
F-eq-det-fBasis[OF K])
  moreover have (∑ i = 0..h. b) = - 6 * of-nat (hof K) * of-nat (nat-log p
    (5 * K))
    - of-nat (hof K) * of-nat (multiplicity p (24::nat))
    by (simp add: b-def h-def algebra-simps)
  ultimately show ?thesis by simp

```

qed

end

theory *Zeta5-Truncated-Expansions*

imports *Zeta5-Far-Poles HOL-Computational-Algebra.Polynomial-FPS*

begin

19 The truncated p -adic layer and Lemma 3.1

Section 3.1 of the paper (Lean theory *PadicTau*). Lean works in $\mathbb{Q}_p$ with convergent series; here every truncated series is a polynomial, so the difference identity (3.3) for a far pole comes with an explicit error polynomial whose coefficients are divisible by p^{N+1} . The near residues and the analytic part (handled with formal power series) are shown to be integral, and Lemma 3.1 is proved in this truncated rational form (*tauX-far-pole-valuation*).

19.1 Truncated expansions of $1/(z - s)$ as polynomials

$\text{farPoly } s \ N = -\sum_{d < N} s^{-(d+1)} z^d$, the degree $< N$ truncation of $1/(z - s)$.

definition $\text{farPoly} :: \text{rat} \Rightarrow \text{nat} \Rightarrow \text{rat poly}$ **where**

$\text{farPoly } s \ N = -(\sum_{d < N} \text{Polynomial.smult } (\text{inverse } s^{(d+1)}) (pX^{(d)}))$

lemma *coeff-farPoly*:

$\text{Polynomial.coeff } (\text{farPoly } s \ N) \ d = (\text{if } d < N \text{ then } -(\text{inverse } s^{(d+1)}) \text{ else } 0)$

unfolding *farPoly-def pX-pow-monom* **by** (*simp add: coeff-sum coeff-monom if-distrib* [**where** $f =$

$\lambda x. - * x$] *cong: if-cong*)

lemma *degree-farPoly-lt*: $0 < N \implies \text{degree } (\text{farPoly } s \ N) < N$

proof –

assume $N: 0 < N$

have $\text{degree } (\text{farPoly } s \ N) \leq N - 1$

by (*rule degree-le*) (*use N in <auto simp: coeff-farPoly>*)

thus *?thesis* **using** N **by** *linarith*

qed

lemma *tau-farPoly*: $\text{tau } (\text{farPoly } s \ N) = -(\sum_{d < N} \text{kappa } d * \text{inverse } s^{(d+1)})$

unfolding *farPoly-def tau-uminus tau-sum tau-smult tau-X-pow-eq-kappa* **by** (*simp add: mult.commute*)

The truncated series (3.5) is τ of the truncated expansion.

lemma *tauAnPole-tr-eq-tau*: $\text{tauAnPole-tr } p \ s \ M = \text{tau } (\text{farPoly } s \ (M + 3))$

```

proof –
  have  $c: \bigwedge k. (Suc (Suc (Suc k)) \text{ choose } 3) * 6 = Suc (Suc (Suc k)) * Suc (Suc k) * Suc k$ 
  proof –
    fix  $k :: nat$ 
    show  $(Suc (Suc (Suc k)) \text{ choose } 3) * 6 = Suc (Suc (Suc k)) * Suc (Suc k) * Suc k$ 
    using choose-falling3[of k 0] by (simp add: numeral-3-eq-3)
  qed
  have  $kap: \bigwedge k. \kappa (Suc (Suc (Suc k))) = of\_nat (Suc (Suc (Suc k)) \text{ choose } 3) * bernoulli\ k / 4$ 
  proof –
    fix  $k :: nat$ 
    have  $e: (of\_nat (Suc (Suc (Suc k)) * Suc (Suc k) * Suc k) :: rat) = of\_nat (Suc (Suc (Suc k)) \text{ choose } 3) * 6$ 
    unfolding c[of k, symmetric] by simp
    show  $\kappa (Suc (Suc (Suc k))) = of\_nat (Suc (Suc (Suc k)) \text{ choose } 3) * bernoulli\ k / 4$ 
    unfolding kappa-Suc3 e by simp
  qed
  have  $pw: \bigwedge k. s \text{ powi } (- (int\ k + 4)) = inverse\ s ^ (k + 4)$ 
  proof –
    fix  $k :: nat$ 
    have  $- (int\ k + 4) = - int\ (k + 4)$  by simp
    thus  $s \text{ powi } (- (int\ k + 4)) = inverse\ s ^ (k + 4)$ 
    by (simp only: power-int-minus power-int-of-nat power-inverse)
  qed
  have  $M3: M + 3 = Suc (Suc (Suc M))$  by simp
  have  $\tau (farPoly\ s\ (M + 3)) = - (\sum k < M. \kappa (Suc (Suc (Suc k))) * inverse\ s ^ (Suc (Suc (Suc k)) + 1))$ 
  unfolding tau-farPoly M3 sum.lessThan-Suc-shift
  by (simp add: kappa-0 kappa-Suc-0 kappa-Suc-Suc-0)
  also have  $(\sum k < M. \kappa (Suc (Suc (Suc k))) * inverse\ s ^ (Suc (Suc (Suc k)) + 1)) =$ 

$$(1 / 4) * (\sum k < M. of\_nat ((k + 3) \text{ choose } 3) * bernoulli\ k * s \text{ powi } (- (int\ k + 4)))$$

  unfolding sum-distrib-left
  proof (rule sum.cong[OF refl])
    fix  $k :: nat$ 
    have  $k3: k + 3 = Suc (Suc (Suc k)) \text{ Suc } (Suc (Suc k)) + 1 = k + 4$  by simp-all
    show  $\kappa (Suc (Suc (Suc k))) * inverse\ s ^ (Suc (Suc (Suc k)) + 1) = 1 / 4 * (of\_nat ((k + 3) \text{ choose } 3) * bernoulli\ k * s \text{ powi } (- (int\ k + 4)))$ 
    unfolding kap pw k3 by simp
  qed
  finally show ?thesis unfolding tauAnPole-tr-def by simp
qed

```

lemma *tauAnPole-tr-eq-sum*: $\text{tauAnPole-tr } p \ s \ M = - (\sum d < M + 3. \text{ kappa } d * \text{inverse } s ^{(d + 1)})$
by (*simp only*: *tauAnPole-tr-eq-tau tau-farPoly*)

$(z - s) \text{ farPoly } s \ N = 1 - s^{\{-N\}} z^N.$

lemma *farPoly-mul*:
 $s \neq 0 \implies (pX - [:s:]) * \text{farPoly } s \ N = 1 - \text{Polynomial.smult } (\text{inverse } s ^ N) (pX ^ N)$

proof –
assume *s*: $s \neq 0$
show *?thesis*
proof (*induction N*)
case 0
show *?case* **by** (*simp add*: *farPoly-def*)
next
case (*Suc N*)
have *fs*: $\text{farPoly } s \ (\text{Suc } N) = \text{farPoly } s \ N - \text{Polynomial.smult } (\text{inverse } s ^ (N + 1)) (pX ^ N)$
by (*simp add*: *farPoly-def*)
have *m*: $(pX - [:s:]) * \text{Polynomial.smult } (\text{inverse } s ^ (N + 1)) (pX ^ N) = \text{Polynomial.smult } (\text{inverse } s ^ (N + 1)) (pX ^ \text{Suc } N) - \text{Polynomial.smult } (\text{inverse } s ^ N) (pX ^ N)$
proof –
have *e*: $s * \text{inverse } s ^ (N + 1) = \text{inverse } s ^ N$ **using** *s* **by** (*simp add*: *field-simps*)
have $(pX - [:s:]) * \text{Polynomial.smult } (\text{inverse } s ^ (N + 1)) (pX ^ N) = \text{Polynomial.smult } (\text{inverse } s ^ (N + 1)) (pX * pX ^ N) - \text{Polynomial.smult } (s * \text{inverse } s ^ (N + 1)) (pX ^ N)$
by (*simp add*: *algebra-simps smult-diff-right*)
thus *?thesis* **unfolding** *e* **by** *simp*
qed
show *?case* **unfolding** *fs right-diff-distrib Suc.IH m* **by** (*simp add*: *algebra-simps*)
qed
qed

The shifted-pole correction $\text{farPoly } s \ N \ (z + 1) - \text{farPoly } (s - 1) \ N$, as an explicit polynomial whose coefficients are divisible by $p^{\{N+1\}}$ when $v\text{-}p(s) < 0$.

definition *hPoly* :: $\text{rat} \Rightarrow \text{nat} \Rightarrow \text{rat poly}$ **where**
 $\text{hPoly } s \ N = \text{Polynomial.smult } (\text{inverse } (s * (s - 1)))$
 $(\sum i < N. (\text{Polynomial.smult } (\text{inverse } (s - 1)) \ pX) ^ i * (\text{Polynomial.smult } (\text{inverse } s) (pX + 1)) ^ (N - 1 - i))$

lemma *vp-sub-one-lt*: $\text{prime } p \implies v\text{-}p \ p \ s < 0 \implies s - 1 \neq 0 \wedge v\text{-}p \ (s - 1) < 0$
proof –
assume *p*: *prime p* **and** *s*: $v\text{-}p \ p \ s < 0$

```

have s0:  $s \neq 0$  using s by (auto simp: vp-def)
have v1:  $vp\ p\ (-\ 1) = 0$  by (simp add: vp-def)
have  $s + -\ 1 \neq 0 \wedge vp\ p\ (s + -\ 1) = vp\ p\ s$ 
  by (rule vp-add-eq-of-lt[OF p s0]) (use s v1 in auto)
thus ?thesis using s by simp
qed

```

```

lemma vpGge-pow: prime p  $\implies$  vpGge p A a  $\implies$ 
  vpGge p (A  $^{\wedge}$  n) (of-nat n * a)
proof -
  assume p: prime p and A: vpGge p A a
  show ?thesis
  proof (induction n)
    case 0 show ?case using vpGge-one by simp
  next
    case (Suc n)
    have vpGge p (A * A  $^{\wedge}$  n) (a + of-nat n * a) by (rule vpGge-mul[OF p A
  Suc.IH])
    thus ?case by (simp add: algebra-simps)
  qed
qed

```

```

lemma vge-inverse-of-vp-lt: prime p  $\implies$  vp p s < 0  $\implies$ 
  vge p (inverse s) 1
  by (simp add: vge-def vp-def qmultiplicity-inverse)

```

```

lemma smult-as-C-mult: Polynomial.smult c P = [:c:] * P
  by simp

```

```

lemma vpGge-hPoly:
  assumes prime p vp p s < 0
  shows vpGge p (hPoly s N) (of-nat (N + 1))
proof -
  obtain v1: vp p (s - 1) < 0 using vp-sub-one-lt[OF assms] by blast
  have is0: vge p (inverse s) 1 by (rule vge-inverse-of-vp-lt[OF assms])
  have is1: vge p (inverse (s - 1)) 1 by (rule vge-inverse-of-vp-lt[OF assms(1)
  v1])
  have c: vge p (inverse (s * (s - 1))) (1 + 1)
    unfolding inverse-mult-distrib by (rule vge-mul[OF assms(1) is0 is1])
  define u where u = Polynomial.smult (inverse (s - 1)) pX
  define w where w = Polynomial.smult (inverse s) (pX + 1)
  have u: vpGge p u (1 + 0)
    unfolding u-def smult-as-C-mult by (rule vpGge-mul[OF assms(1) vpGge-C[OF
  is1] vpGge-pX])
  have w: vpGge p w (1 + 0)
    unfolding w-def smult-as-C-mult by (rule vpGge-mul[OF assms(1) vpGge-C[OF
  is0] vpGge-add[OF
  assms(1) vpGge-pX vpGge-one]])
  have sum: vpGge p ( $\sum\ i < N. u^{\wedge} i * w^{\wedge} (N - 1 - i)$ ) (of-nat N - 1)

```

proof (rule vpGge-sum[OF assms(1) finite-lessThan])
fix i **assume** $i: i \in \{..<N\}$
have $vpGge\ p\ (u \wedge^i * w \wedge^{(N-1-i)})\ (of\text{-}nat\ i * (1 + 0) + of\text{-}nat\ (N - 1 - i) * (1 + 0))$
by (rule vpGge-mul[OF assms(1) vpGge-pow[OF assms(1) u] vpGge-pow[OF assms(1) w]])
moreover have $of\text{-}nat\ i * (1 + 0) + of\text{-}nat\ (N - 1 - i) * (1 + 0) = (of\text{-}nat\ N - 1 :: rat)$
using i **by** (simp add: of-nat-diff)
ultimately show $vpGge\ p\ (u \wedge^i * w \wedge^{(N-1-i)})\ (of\text{-}nat\ N - 1)$ **by** simp
qed
have $vpGge\ p\ ([:inverse\ (s * (s - 1)):] * (\sum i < N. u \wedge^i * w \wedge^{(N-1-i)}))\ (1 + 1 + (of\text{-}nat\ N - 1))$
by (rule vpGge-mul[OF assms(1) vpGge-C[OF c] sum])
hence $vpGge\ p\ (Polynomial.smult\ (inverse\ (s * (s - 1)))\ (\sum i < N. u \wedge^i * w \wedge^{(N-1-i)}))\ (of\text{-}nat\ (N + 1))$
unfolding $smult\text{-}as\text{-}C\text{-}mult[of - \sum i < N. u \wedge^i * w \wedge^{(N-1-i)}]$
by (rule vpGge-mono) simp
thus ?thesis **unfolding** hPoly-def u-def w-def .
qed

lemma farPoly-shift:

assumes $s \neq 0\ s - 1 \neq 0$
shows $pcompose\ (farPoly\ s\ N)\ (pX + 1) - farPoly\ (s - 1)\ N = hPoly\ s\ N$
proof -
have $nz: pX - [:s - 1:] \neq 0$ **by** (rule pX-minus-const-nonzero)
have $lhs: (pX - [:s - 1:]) * (pcompose\ (farPoly\ s\ N)\ (pX + 1) - farPoly\ (s - 1)\ N) =$

$$Polynomial.smult\ (inverse\ (s - 1) \wedge N)\ (pX \wedge N) -$$

$$Polynomial.smult\ (inverse\ s \wedge N)\ ((pX + 1) \wedge N)$$

proof -
have $c1: pcompose\ (pX - [:s:])\ (pX + 1) = pX - [:s - 1:]$
unfolding $pcompose\text{-}diff\ pcompose\text{-}pX\ pcompose\text{-}const$ **by** (simp add: one-pCons algebra-simps)
have $c2: pcompose\ (1 - Polynomial.smult\ (inverse\ s \wedge N)\ (pX \wedge N))\ (pX + 1) =$

$$1 - Polynomial.smult\ (inverse\ s \wedge N)\ ((pX + 1) \wedge N)$$

unfolding $pcompose\text{-}diff\ pcompose\text{-}1\ pcompose\text{-}smult\ pcompose\text{-}power\text{-}left\ pcompose\text{-}pX$..
have $A: (pX - [:s - 1:]) * pcompose\ (farPoly\ s\ N)\ (pX + 1) =$

$$1 - Polynomial.smult\ (inverse\ s \wedge N)\ ((pX + 1) \wedge N)$$

using $arg\text{-}cong[OF\ farPoly\text{-}mul[OF\ assms(1),\ of\ N],\ of\ \lambda P. pcompose\ P\ (pX + 1)]$
unfolding $pcompose\text{-}mult\ c1\ c2$.
have $B: (pX - [:s - 1:]) * farPoly\ (s - 1)\ N =$

$$1 - Polynomial.smult\ (inverse\ (s - 1) \wedge N)\ (pX \wedge N)$$

by (rule farPoly-mul[OF assms(2)])
show ?thesis **unfolding** right-diff-distrib $A\ B$ **by** simp
qed


```

have rhs: (pX - [:s - 1:]) * hPoly s N =
  Polynomial.smult (inverse (s - 1) ^ N) (pX ^ N) -
  Polynomial.smult (inverse s ^ N) ((pX + 1) ^ N)
proof -
  define u where u = Polynomial.smult (inverse (s - 1)) pX
  define w where w = Polynomial.smult (inverse s) (pX + 1)
  have X: pX = [:0, 1:] by (simp add: monom-Suc)
  have uw: u - w = Polynomial.smult (inverse (s * (s - 1))) (pX - [:s - 1:])
    unfolding pX-minus-const unfolding u-def w-def X using assms
    by (simp add: one-pCons field-simps)
  have tel: u ^ N - w ^ N = (u - w) * (∑ i < N. u ^ i * w ^ (N - 1 - i))
    using power-diff-sumr2[of u N w] by (simp add: mult.commute)
  have e1: (pX - [:s - 1:]) * hPoly s N = (u - w) * (∑ i < N. u ^ i * w ^
(N - 1 - i))
    unfolding hPoly-def uw u-def[symmetric] w-def[symmetric] by simp
  have uN: u ^ N = Polynomial.smult (inverse (s - 1) ^ N) (pX ^ N) unfolding
u-def
    by (rule smult-power)
  have wN: w ^ N = Polynomial.smult (inverse s ^ N) ((pX + 1) ^ N) unfolding
w-def
    by (rule smult-power)
  show ?thesis unfolding e1 tel[symmetric] uN wN ..
qed
show ?thesis using lhs rhs nz by (metis mult-cancel-left)
qed

```

```

lemma vge-kappa-m4: prime p ⟹ vge p (kappa d) (- 4)
proof -
  assume p: prime p
  have m24: vp p 24 ≤ 3
  proof (cases p = 2)
    case False
    thus ?thesis using vp-24[OF p False] by simp
  next
    case True
    have multiplicity (2::int) 24 = 3 by (rule multiplicity-eqI) simp-all
    thus ?thesis unfolding True vp-def using qmultiplicity-of-int[of 2 24] by simp
  qed
  have vp p (inverse 24) = - vp p 24 unfolding vp-def by (rule qmultiplic-
ity-inverse)
  hence i24: vge p (inverse 24) (- 3) unfolding vge-def using m24 by simp
  have B: vge p (bernoulli (d - 3)) (- 1)
    using vp-bernoulli-ge[OF p, of d - 3] unfolding vge-def
    by (metis of-int-le-iff of-int-minus of-int-1)
  have vge p (of-int (int d * (int d - 1) * (int d - 2)) * bernoulli (d - 3) *
inverse 24) (0 + -
  1 + - 3)
    by (intro vge-mul[OF p] vge-of-int[OF p] B i24)
  note v = this

```

have e : $\text{kappa } d = \text{of-int } (\text{int } d * (\text{int } d - 1) * (\text{int } d - 2)) * \text{bernoulli } (d - 3)$
 $* \text{inverse } 24$
unfolding kappa-eq **by** $(\text{rule } \text{divide-inverse})$
show $?thesis$ **unfolding** e **by** $(\text{rule } \text{vge-mono}[OF\ v]) \text{ simp}$
qed

lemma vge-tau-of-vpGge : $\text{prime } p \implies \text{vpGge } p\ P\ b \implies$
 $\text{vge } p\ (\text{tau } P)\ (b - 4)$
proof –
assume p : $\text{prime } p$ **and** P : $\text{vpGge } p\ P\ b$
have $\text{vge } p\ (\text{tauK } P)\ (b + - 4)$
unfolding tauK-def **by** $(\text{intro } \text{vge-sum}[OF\ p] \text{ finite-atMost } \text{vge-mul}[OF\ p]$
 $\text{vge-kappa-m4}[OF\ p]) \text{ (use}$
 $P \text{ in } \langle \text{simp add: vpGge-iff} \rangle)$
thus $?thesis$ **by** $(\text{simp add: tauK-eq-tau})$
qed

lemma coeff-farPoly-4 : $5 \leq N \implies$
 $\text{Polynomial.coeff } (\text{farPoly } s\ N)\ 4 = - (s \text{ powi } (- 5))$
proof –
assume $5 \leq N$
hence $\text{Polynomial.coeff } (\text{farPoly } s\ N)\ 4 = - (\text{inverse } s \wedge 5)$ **by** $(\text{simp add:}$
 $\text{coeff-farPoly})$
also have $\text{inverse } s \wedge 5 = s \text{ powi } (- 5)$ **by** $(\text{simp add: power-int-minus power-inverse})$
finally show $?thesis$.
qed

The difference identity (3.3) for a far pole, with explicit error: the discrepancy equals $\tau(h)$ and has valuation $\geq M$.

lemma $\text{tauAnPole-tr-sub-eq}$:
assumes p : $\text{prime } p$ **and** s : $\text{vp } p\ s < 0$ **and** $s0$: $s \neq 0$ **and** $M2$: $2 \leq M$
shows $\text{vge } p\ (\text{tauAnPole-tr } p\ s\ M - \text{tauAnPole-tr } p\ (s - 1)\ M - s \text{ powi } (- 5))$
 $(\text{of-nat } M)$
proof –
define N **where** $N = M + 3$
have $s1$: $s - 1 \neq 0$ **using** $\text{vp-sub-one-lt}[OF\ p\ s]$ **by** simp
have $e1$: $\text{tauAnPole-tr } p\ s\ M - \text{tauAnPole-tr } p\ (s - 1)\ M - s \text{ powi } (- 5) =$
 $\text{tau } (\text{farPoly } s\ N) - \text{tau } (\text{farPoly } (s - 1)\ N) - s \text{ powi } (- 5)$
by $(\text{simp only: tauAnPole-tr-eq-tau } N\text{-def})$
have $e2$: $\text{tau } (\text{pcompose } (\text{farPoly } s\ N)\ (pX + 1)) - \text{tau } (\text{farPoly } s\ N) = - (s$
 $\text{powi } (- 5))$
using $\text{tau-fwd}[of\ \text{farPoly } s\ N] \text{ coeff-farPoly-4}[of\ N\ s]\ M2$ **by** $(\text{simp add: tau-diff}$
 $N\text{-def})$
have $e3$: $\text{tau } (\text{pcompose } (\text{farPoly } s\ N)\ (pX + 1)) - \text{tau } (\text{farPoly } (s - 1)\ N) =$
 $\text{tau } (h\text{Poly } s\ N)$
using $\text{arg-cong}[OF\ \text{farPoly-shift}[OF\ s0\ s1, of\ N], of\ \text{tau}]$ **by** $(\text{simp add: tau-diff})$
have v : $\text{vge } p\ (\text{tau } (h\text{Poly } s\ N))\ (\text{of-nat } (N + 1) - 4)$
by $(\text{rule } \text{vge-tau-of-vpGge}[OF\ p\ \text{vpGge-hPoly}[OF\ p\ s]])$
have $\text{tauAnPole-tr } p\ s\ M - \text{tauAnPole-tr } p\ (s - 1)\ M - s \text{ powi } (- 5) = \text{tau}$

```

(hPoly s N)
  using e1 e2 e3 by linarith
  thus ?thesis using v by (simp add: N-def)
qed

```

The difference identity (3.3) for a far pole: $\tau^{\sim\{an\}}(1/(z-s)) - \tau^{\sim\{an\}}(1/(z-(s-1))) = s^{\sim\{-5\}}$ holds in $\mathbb{Q}\text{-}p$; for the partial sums it holds modulo any prescribed power of p once M is large (Lean: *tauAnPole-sub*).

lemma *tauAnPole-tr-sub*:

```

  assumes prime p vp p s < 0 s ≠ 0
  shows ∃ M0. ∀ M ≥ M0. vge p (tauAnPole-tr p s M - tauAnPole-tr p (s - 1)
M -
  s powi (- 5)) B

```

proof

```

  show ∀ M ≥ nat ⌈B⌉ + 2. vge p (tauAnPole-tr p s M - tauAnPole-tr p (s -
  1) M - s powi (- 5)) B
  proof (intro allI impI)
    fix M assume M: nat ⌈B⌉ + 2 ≤ M
    have v: vge p (tauAnPole-tr p s M - tauAnPole-tr p (s - 1) M - s powi (-
  5)) (of-nat M)
      by (rule tauAnPole-tr-sub-eq[OF assms]) (use M in simp)
    have nat ⌈B⌉ ≤ M using M by simp
    hence ⌈B⌉ ≤ int M by (simp add: nat-le-iff)
    hence (of-int ⌈B⌉ :: rat) ≤ of-nat M
      by (metis of-int-le-iff of-int-of-nat-eq)
    hence B ≤ of-nat M using le-of-int-ceiling[of B] by linarith
    thus vge p (tauAnPole-tr p s M - tauAnPole-tr p (s - 1) M - s powi (- 5))
  B
    by (rule vge-mono[OF v])
  qed
qed

```

19.2 The near residues

lemma *vp-of-int-nonneg*: $\text{prime } p \implies \text{of-int } r \neq (0 :: \text{rat}) \implies 0 \leq \text{vp } p (\text{of-int } r)$
 by (simp add: vp-def qmultiplicity-of-int)

lemma *far-ne-int*: $\text{prime } p \implies \text{vp } p s = -1 \implies s \neq \text{of-int } r$

proof –

```

  assume p: prime p and v: vp p s = - 1
  show ?thesis
  proof
    assume e: s = of-int r
    show False
  proof (cases (of-int r :: rat) = 0)
    case True thus False using v e by (simp add: vp-def)
  next
    case False thus False using vp-of-int-nonneg[OF p False] v e by simp
  end

```

qed
 qed
 qed

lemma *far-disj*: $\text{prime } p \implies (\bigwedge s. s \in F \implies$
 $\text{vp } p \ s = -1) \implies \text{of-int } T \cap F = \{\}$
proof –
assume p : *prime* p **and** h : $\bigwedge s. s \in F \implies \text{vp } p \ s = -1$
show *?thesis* **using** *far-ne-int*[*OF* p h] **by** *blast*
 qed

An integer minus a far pole has valuation exactly -1 .

lemma *vp-int-sub-far*:
 $\text{prime } p \implies \text{vp } p \ s = -1 \implies \text{of-int } r - s \neq 0 \wedge$
 $\text{vp } p \ (\text{of-int } r - s) = -1$
proof –
assume p : *prime* p **and** v : $\text{vp } p \ s = -1$
have $s0$: $s \neq 0$ **using** v **by** (*auto simp: vp-def*)
have vm : $\text{vp } p \ (-s) = -1$ **using** v **by** (*simp add: vp-def*)
show *?thesis*
proof (*cases* ($\text{of-int } r :: \text{rat}$) = 0)
case *True* **thus** *?thesis* **using** $s0$ vm **by** *simp*
next
case *False*
have $-s + \text{of-int } r \neq 0 \wedge \text{vp } p \ (-s + \text{of-int } r) = \text{vp } p \ (-s)$
by (*rule vp-add-eq-of-lt*[*OF* p]) (*use* $s0$ vm *vp-of-int-nonneg*[*OF* p *False*] *False*
in *auto*)
thus *?thesis* **using** vm **by** (*simp add: algebra-simps*)
 qed
 qed

lemma *vp-int-sub-near*:
 $\text{prime } p \implies \neg \text{int } p \ \text{dvd } (r - r') \implies$
 $\text{of-int } r - \text{of-int } r' \neq (0 :: \text{rat}) \wedge \text{vp } p \ (\text{of-int } r - \text{of-int } r') = 0$
proof –
assume p : *prime* p **and** nd : $\neg \text{int } p \ \text{dvd } (r - r')$
have $r - r' \neq 0$ **using** nd **by** *auto*
moreover **have** $\text{multiplicity } (\text{int } p) \ (r - r') = 0$ **by** (*rule not-dvd-imp-multiplicity-0*[*OF* nd])
ultimately show *?thesis* **unfolding** *of-int-diff*[*symmetric*] *vp-def*
by (*simp add: qmultiplicity-of-int del: of-int-diff*)
 qed

lemma *vp-neg-p-powi*: $\text{prime } p \implies \text{vp } p \ ((-\text{of-nat } p) \ \text{powi } (-\text{int } n)) = -\text{int } n$
proof –
assume p : *prime* p
have pe : *prime-elem* ($\text{int } p$) **using** p **by** (*simp add: prime-nat-int-transfer*)
have ne : $(-\text{of-nat } p :: \text{rat}) \neq 0$ **using** *prime-gt-0-nat*[*OF* p] **by** *simp*
have $\text{vp } p \ (-\text{of-nat } p) = 1$ **using** pe

by (simp add: vp-def qmultiplicity-of-nat multiplicity-prime)
 with vp-powi-rat[OF p ne, of - int n] show ?thesis by simp
 qed

The near residues are integral (Lean: *pt-near-res-vge*): poles T distinct modulo p , far poles F with $v\text{-}p(s) = -1$, numerator $(-p)^{\wedge\{-|F|\}}$ U with U integral.

lemma *near-res-vge*:

assumes p : prime p and T : finite T and F : finite F
 and hT : $\bigwedge r s. r \in T \implies s \in T \implies$
 $r \neq s \implies \neg \text{int } p \text{ dvd } (r - s)$
 and hF : $\bigwedge s. s \in F \implies$
 $vp \text{ } p \text{ } s = -1$ and U : $vpGge \text{ } p \text{ } U \text{ } 0$ and r : $r \in T$
 shows $vge \text{ } p \text{ } (qRes \text{ } (of\text{-}int \text{ } 'T \cup F) \text{ } ([:(- \text{ } of\text{-}nat \text{ } p) \text{ } powi \text{ } (- \text{ } int \text{ } (card \text{ } F))]:] * U) \text{ } (of\text{-}int \text{ } r)) \text{ } 0$
proof -
 define $Tq :: \text{rat set}$ where $Tq = of\text{-}int \text{ } 'T$
 define c where $c = ((- \text{ } of\text{-}nat \text{ } p) \text{ } powi \text{ } (- \text{ } int \text{ } (card \text{ } F)) :: \text{rat})$
 have $disj$: $Tq \cap F = \{\}$ **unfolding** $Tq\text{-}def$ **by** (rule *far-disj*[OF $p \text{ } hF$])
 have $of\text{-}int \text{ } r \in Tq$ **using** r **by** (simp add: $Tq\text{-}def$)
 with $disj$ have $erase$: $(Tq \cup F) - \{of\text{-}int \text{ } r\} = (Tq - \{of\text{-}int \text{ } r\}) \cup F$ **by** *auto*
 have $near$: $\bigwedge x. x \in Tq - \{of\text{-}int \text{ } r\} \implies of\text{-}int \text{ } r - x \neq 0 \wedge$
 $vp \text{ } p \text{ } (of\text{-}int \text{ } r - x) = 0$
proof -
 fix x assume $x \in Tq - \{of\text{-}int \text{ } r\}$
 then obtain r' where r' : $r' \in T \text{ } x = of\text{-}int \text{ } r' \text{ } r' \neq r$ **unfolding** $Tq\text{-}def$ **by**
auto
 show $of\text{-}int \text{ } r - x \neq 0 \wedge vp \text{ } p \text{ } (of\text{-}int \text{ } r - x) = 0$
unfolding $r'(2)$ **by** (rule *vp-int-sub-near*[OF $p \text{ } hT$ [OF $r \text{ } r'(1)$]]) (use r' in
auto)
 qed
 have far : $\bigwedge s. s \in F \implies of\text{-}int \text{ } r - s \neq 0 \wedge$
 $vp \text{ } p \text{ } (of\text{-}int \text{ } r - s) = -1$
using *vp-int-sub-far*[OF $p \text{ } hF$] .
 have den : $(\prod x \in (Tq \cup F) - \{of\text{-}int \text{ } r\}. of\text{-}int \text{ } r - x) =$
 $(\prod x \in Tq - \{of\text{-}int \text{ } r\}. of\text{-}int \text{ } r - x) * (\prod s \in F. of\text{-}int \text{ } r - s)$
unfolding $erase$ **by** (rule *prod.union-disjoint*) (use $disj \text{ } T \text{ } F$ in $\langle auto \text{ } simp:$
 $Tq\text{-}def \rangle$)
 have $den\text{-}ne$: $(\prod x \in Tq - \{of\text{-}int \text{ } r\}. of\text{-}int \text{ } r - x) * (\prod s \in$
 $F. of\text{-}int \text{ } r - s) \neq 0$
proof -
 have $finite \text{ } (Tq - \{of\text{-}int \text{ } r\})$ **using** T **by** (simp add: $Tq\text{-}def$)
 hence $(\prod x \in Tq - \{of\text{-}int \text{ } r\}. of\text{-}int \text{ } r - x) \neq 0$ **using** $near$
by (simp add: *prod-zero-iff*)
 moreover have $(\prod s \in F. of\text{-}int \text{ } r - s) \neq 0$ **using** $F \text{ } far$
by (simp add: *prod-zero-iff*)
 ultimately show ?thesis **by** *simp*
 qed
 have $den\text{-}v$: $vp \text{ } p \text{ } ((\prod x \in Tq - \{of\text{-}int \text{ } r\}. of\text{-}int \text{ } r - x) * (\prod s \in$

```

     $F. \text{ of-int } r - s)) = - \text{ int } (\text{card } F)$ 
  proof -
    have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer)
    have fT: finite (Tq - {of-int r}) using T by (simp add: Tq-def)
    have A0:  $(\prod x \in Tq - \{ \text{of-int } r \}. \text{ of-int } r - x) \neq 0$  and B0:  $(\prod s \in F. \text{ of-int } r - s) \neq 0$ 
    using den-ne by auto
    have vp p  $(\prod x \in Tq - \{ \text{of-int } r \}. \text{ of-int } r - x) = (\sum x \in Tq - \{ \text{of-int } r \}. vp p (\text{ of-int } r - x))$ 
    by (rule vp-prod-rat[OF p fT]) (use near in blast)
    also have ... = 0 by (rule sum.neutral) (use near in blast)
    finally have vA:  $vp p (\prod x \in Tq - \{ \text{of-int } r \}. \text{ of-int } r - x) = 0$  .
    have vp p  $(\prod s \in F. \text{ of-int } r - s) = (\sum s \in F. vp p (\text{ of-int } r - s))$ 
    by (rule vp-prod-rat[OF p F]) (use far in blast)
    also have ... =  $(\sum s \in F. - 1)$  by (rule sum.cong[OF refl]) (use far in blast)
    finally have vB:  $vp p (\prod s \in F. \text{ of-int } r - s) = - \text{ int } (\text{card } F)$  by simp
    have vp p  $((\prod x \in Tq - \{ \text{of-int } r \}. \text{ of-int } r - x) * (\prod s \in F. \text{ of-int } r - s)) =$ 
       $vp p (\prod x \in Tq - \{ \text{of-int } r \}. \text{ of-int } r - x) + vp p (\prod s \in F. \text{ of-int } r - s)$ 
    using A0 B0 pe by (simp add: vp-def qmultiplicity-mult)
    thus ?thesis using vA vB by simp
  qed
  have num:  $vge p (\text{poly } ([c:] * U) (\text{ of-int } r)) (- \text{ of-nat } (\text{card } F))$ 
  proof -
    have vc:  $vge p c (- \text{ of-nat } (\text{card } F))$  unfolding c-def vge-def
    using vp-neg-p-powi[OF p, of card F] by simp
    have vge p  $(c * \text{poly } U (\text{ of-int } r)) (- \text{ of-nat } (\text{card } F) + 0)$ 
    by (rule vge-mul[OF p vc vge-poly-eval[OF p U vge-of-int[OF p]]])
    thus ?thesis by simp
  qed
  show ?thesis
    unfolding qRes-def Tq-def[symmetric] c-def[symmetric] den
  proof -
    have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer)
    define n where  $n = \text{poly } ([c:] * U) (\text{ of-int } r)$ 
    define D where  $D = (\prod x \in Tq - \{ \text{of-int } r \}. \text{ of-int } r - x) * (\prod s \in F. \text{ of-int } r - s)$ 
    have D0:  $D \neq 0$  unfolding D-def by (rule den-ne)
    have vD:  $vp p D = - \text{ int } (\text{card } F)$  unfolding D-def by (rule den-v)
    have vn:  $vge p n (- \text{ of-nat } (\text{card } F))$  unfolding n-def by (rule num)
    have vge p  $(n / D) 0$ 
    proof (cases n = 0)
      case True thus ?thesis by (simp add: vge-def)
    next
      case False
      have  $vp p (n / D) = vp p n - vp p D$  unfolding vp-def using False D0 pe
      by (simp add: qmultiplicity-divide)
      thus ?thesis using vn vD False D0 by (simp add: vge-def)
    end
  end

```

qed
thus $vge\ p\ (poly\ ([:c:] * U)\ (of-int\ r) / ((\prod x \in$
 $Tq - \{of-int\ r\}. of-int\ r - x) * (\prod s \in F. of-int\ r - s)))\ 0$
unfolding $n-def\ D-def$.
qed
qed

19.3 The analytic part

$-1/(X - s) = \sum s^{\wedge\{-(d+1)\}} X^{\wedge d}$ as a formal power series.

definition $farSeries :: rat \Rightarrow rat\ fps$ **where**
 $farSeries\ s = Abs-fps\ (\lambda d. inverse\ s^{\wedge}(d + 1))$

lemma $X-sub-C-mul-farSeries: s \neq 0 \implies$
 $fps-of-poly\ (pX - [:s:]) * farSeries\ s = -\ 1$

proof –

assume $s: s \neq 0$

show $?thesis$

proof $(rule\ fps-ext)$

fix n

show $(fps-of-poly\ (pX - [:s:]) * farSeries\ s)\ \$\ n = (-\ 1 :: rat\ fps)\ \$\ n$

proof $(cases\ n)$

case 0 **thus** $?thesis$ **using** s **unfolding** $pX-minus-const$ **by** $(simp\ add:$
 $farSeries-def\ fps-mult-nth)$

next

case $(Suc\ m)$

have $(fps-of-poly\ (pX - [:s:]) * farSeries\ s)\ \$\ n =$

$- s * inverse\ s^{\wedge}(Suc\ m + 1) + (\sum i =$
 $0..m. Polynomial.coeff\ [:1:] i * inverse\ s^{\wedge}(m - i + 1))$

unfolding $Suc\ pX-minus-const\ fps-mult-nth\ sum.atLeast0-atMost-Suc-shift$
by $(simp\ add: farSeries-def)$

also have $\dots = 0$ **using** s **by** $(simp\ add: atLeast0AtMost\ coeff-pCons$
 $sum.atMost-shift$
 $field-simps)$

finally show $?thesis$ **using** Suc **by** $simp$

qed

qed

qed

The Taylor coefficients of $Q + \sum -\{s \in F\}\ \varrho-s/(z - s): \Pi-F \cdot f = \Pi-F\ Q + \sum \varrho-s\ \Pi-\{F - s\}$.

lemma $qPoleProd-mul-fCoeff:$

assumes $finite\ F \wedge s. s \in F \implies s \neq 0$

shows $fps-of-poly\ (qPoleProd\ F) * Abs-fps\ (fCoeff\ Q\ F\ \varrho) =$

$fps-of-poly\ (qPoleProd\ F * Q + (\sum s \in F. [: \varrho\ s:] * qPoleProd\ (F - \{s\})))$

proof –

have $A: Abs-fps\ (fCoeff\ Q\ F\ \varrho) = fps-of-poly\ Q - (\sum s \in$

$F. fps-const\ (\varrho\ s) * farSeries\ s)$

```

  by (rule fps-ext) (simp add: fCoeff-def farSeries-def fps-sum-nth)
have B:  $\bigwedge s. s \in F \implies$ 
  fps-of-poly (qPoleProd F) * farSeries s = - fps-of-poly (qPoleProd (F - {s}))
proof -
  fix s assume sF:  $s \in F$ 
  have qPoleProd F = (pX - [:s:]) * qPoleProd (F - {s}) unfolding qPoleProd-def
  using assms(1) sF by (rule prod.remove)
  hence fps-of-poly (qPoleProd F) * farSeries s =
    fps-of-poly (qPoleProd (F - {s})) * (fps-of-poly (pX - [:s:]) * farSeries s)
  by (simp only: fps-of-poly-mult ac-simps)
  also have ... = - fps-of-poly (qPoleProd (F - {s}))
  using X-sub-C-mul-farSeries[OF assms(2)[OF sF]] by simp
  finally show fps-of-poly (qPoleProd F) * farSeries s = - fps-of-poly (qPoleProd
(F - {s})) .
qed
have fps-of-poly (qPoleProd F) * Abs-fps (fCoeff Q F  $\varrho$ ) =
  fps-of-poly (qPoleProd F * Q) - ( $\sum s \in$ 
    F. fps-const ( $\varrho$  s) * (fps-of-poly (qPoleProd F) * farSeries s))
  unfolding A by (simp add: fps-of-poly-mult right-diff-distrib sum-distrib-left
ac-simps)
  also have ( $\sum s \in F$ . fps-const ( $\varrho$  s) * (fps-of-poly (qPoleProd F) * farSeries s))
  =
    ( $\sum s \in F$ . - (fps-const ( $\varrho$  s) * fps-of-poly (qPoleProd (F - {s}))))
  by (rule sum.cong[OF refl]) (simp add: B)
  also have fps-of-poly (qPoleProd F * Q) - ( $\sum s \in$ 
    F. - (fps-const ( $\varrho$  s) * fps-of-poly (qPoleProd (F - {s})))) =
    fps-of-poly (qPoleProd F * Q + ( $\sum s \in F$ . [: $\varrho$  s:] * qPoleProd (F -
{s})))
  by (simp add: sum-negf fps-of-poly-add fps-of-poly-sum fps-of-poly-mult fps-of-poly-smult)
  finally show ?thesis .
qed

```

If $\Psi f = N$ as power series, with $\Psi(0)$ a unit, the other coefficients of Ψ in $p\mathbb{Z}$ - p , N integral and $\equiv 0 \pmod{p}$ beyond degree $p+1$, then so is f .

lemma series-vge:

```

assumes p: prime p and P0: Polynomial.coeff  $\Psi$  0  $\neq$ 
  0 and P0v:  $\forall p. p \mid \text{Polynomial.coeff } \Psi \ 0 \implies$ 
  and P1:  $\bigwedge n. 1 \leq n \implies \text{vge } p \ (\text{Polynomial.coeff } \Psi \ n) \ 1$ 
  and N0:  $\forall p. p \mid N \implies$  and N1:  $\bigwedge d. p + 1 < d \implies$ 
     $\text{vge } p \ (\text{Polynomial.coeff } N \ d) \ 1$ 
  and mul: fps-of-poly  $\Psi$  * Abs-fps f = fps-of-poly N
shows  $\text{vge } p \ (f \ d) \ 0 \wedge (p + 1 < d \implies \text{vge } p \ (f \ d) \ 1)$ 
proof (induction d rule: less-induct)
case (less d)
have fd:  $f \ d = \text{inverse } (\text{Polynomial.coeff } \Psi \ 0) *
  (\text{Polynomial.coeff } N \ d - (\sum i <
    d. \text{Polynomial.coeff } \Psi \ (i + 1) * f \ (d - (i + 1))))$ 
proof -
  have Polynomial.coeff N d = (fps-of-poly  $\Psi$  * Abs-fps f)  $\$$  d unfolding mul

```



```

by simp
also have ... = (∑ i ≤ d. Polynomial.coeff Ψ i * f (d - i))
  by (simp add: fps-mult-nth atLeast0AtMost)
also have ... = Polynomial.coeff Ψ 0 * f d + (∑ i <
  d. Polynomial.coeff Ψ (Suc i) * f (d - Suc i))
  by (simp add: sum.atMost-shift)
finally have e: Polynomial.coeff Ψ 0 * f d = Polynomial.coeff N d - (∑ i <
  d. Polynomial.coeff Ψ (Suc i) * f (d - Suc i))
  by simp
show ?thesis using P0 by (simp add: e[symmetric] mult.assoc[symmetric])
qed
have S: vge p (∑ i < d. Polynomial.coeff Ψ (i + 1) * f (d - (i + 1))) 1
proof (rule vge-sum[OF p finite-lessThan])
  fix i assume i: i ∈ {..

```

The shape of $\Psi = \prod (p \ s - p \ z)$: unit constant term, other coefficients in $p\mathbb{Z}-p$.

definition $PsiShape :: nat \Rightarrow rat \ poly \Rightarrow bool$ **where**

```

PsiShape p P  $\longleftrightarrow$  Polynomial.coeff P 0  $\neq$  0  $\wedge$ 
  vp p (Polynomial.coeff P 0) = 0  $\wedge$ 
  ( $\forall n. 1 \leq n \longrightarrow vge \ p \ (Polynomial.coeff \ P \ n) \ 1$ )

```

lemma $PsiShape-one$: $PsiShape \ p \ 1$

by (auto simp: PsiShape-def vp-def vge-def one-pCons coeff-pCons split: nat.splits)

lemma $PsiShape-vpGge$: $prime \ p \implies PsiShape \ p \ P \implies vpGge \ p \ P \ 0$

proof –

assume $PsiShape \ p \ P$

hence a : $Polynomial.coeff \ P \ 0 \neq 0 \wedge vp \ p \ (Polynomial.coeff \ P \ 0) = 0 \wedge n. 1 \leq n \implies vge \ p \ (Polynomial.coeff \ P \ n) \ 1$

unfolding $PsiShape-def$ **by** auto

show ?thesis **unfolding** $vpGge-iff$

```

proof
  fix  $i$  show  $vge\ p\ (Polynomial.coeff\ P\ i)\ 0$ 
  proof (cases  $i$ )
    case  $0$  thus ?thesis using  $a$  by (simp add: vge-def)
  next
    case (Suc  $j$ ) thus ?thesis by (intro vge-mono[OF a(3)][of i]] simp-all
  qed
qed
qed

lemma PsiShape-mult:  $prime\ p \implies PsiShape\ p\ P \implies$ 
   $PsiShape\ p\ Q \implies PsiShape\ p\ (P * Q)$ 
proof –
  assume  $p$ : prime  $p$  and  $P$ : PsiShape  $p\ P$  and  $Q$ : PsiShape  $p\ Q$ 
  have  $pe$ : prime-elem (int  $p$ ) using  $p$  by (simp add: prime-nat-int-transfer)
  have  $P0$ : vpGge  $p\ P\ 0$  by (rule PsiShape-vpGge[OF p P])
  have  $Q0$ : vpGge  $p\ Q\ 0$  by (rule PsiShape-vpGge[OF p Q])
  from  $P$  have  $a$ :  $Polynomial.coeff\ P\ 0 \neq 0\ vp\ p\ (Polynomial.coeff\ P\ 0) =$ 
     $0 \wedge n. 1 \leq n \implies vge\ p\ (Polynomial.coeff\ P\ n)\ 1$ 
  unfolding PsiShape-def by auto
  from  $Q$  have  $b$ :  $Polynomial.coeff\ Q\ 0 \neq 0\ vp\ p\ (Polynomial.coeff\ Q\ 0) =$ 
     $0 \wedge n. 1 \leq n \implies vge\ p\ (Polynomial.coeff\ Q\ n)\ 1$ 
  unfolding PsiShape-def by auto
  have  $c0$ :  $Polynomial.coeff\ (P * Q)\ 0 = Polynomial.coeff\ P\ 0 * Polynomial.coeff\ Q\ 0$ 
  by (simp add: coeff-mult)
  have  $hi$ :  $\bigwedge n. 1 \leq n \implies vge\ p\ (Polynomial.coeff\ (P * Q)\ n)\ 1$ 
  proof –
    fix  $n :: nat$  assume  $n$ :  $1 \leq n$ 
    show  $vge\ p\ (Polynomial.coeff\ (P * Q)\ n)\ 1$  unfolding coeff-mult
    proof (rule vge-sum[OF p finite-atMost])
      fix  $i$  assume  $i$ :  $i \in \{..n\}$ 
      show  $vge\ p\ (Polynomial.coeff\ P\ i * Polynomial.coeff\ Q\ (n - i))\ 1$ 
      proof (cases  $i = 0$ )
        case True
          have  $q$ :  $vge\ p\ (Polynomial.coeff\ Q\ (n - i))\ 1$  by (rule b(3)) (use True n in simp)
          have  $vge\ p\ (Polynomial.coeff\ P\ i * Polynomial.coeff\ Q\ (n - i))\ (0 + 1)$ 
            by (rule vge-mul[OF p - q]) (use P0 in <simp add: vpGge-iff>)
          thus ?thesis by simp
        case False
          have  $q$ :  $vge\ p\ (Polynomial.coeff\ P\ i)\ 1$  by (rule a(3)) (use False in simp)
          have  $vge\ p\ (Polynomial.coeff\ P\ i * Polynomial.coeff\ Q\ (n - i))\ (1 + 0)$ 
            by (rule vge-mul[OF p q]) (use Q0 in <simp add: vpGge-iff>)
          thus ?thesis by simp
      qed
    qed
  qed

```

show *?thesis unfolding PsiShape-def using a b c0 hi pe by (simp add: vp-def qmultiplicity-mult)*

qed

lemma *PsiShape-prod:*

*prime p $\implies$ finite I $\implies$ ($\bigwedge i. i \in I \implies$
*PsiShape p (f i) $\implies$ PsiShape p ($\prod i \in I. f i$)**

proof –

assume *p: prime p and fin: finite I and h: $\bigwedge i. i \in I \implies$
*PsiShape p (f i)**

show *?thesis using fin h*

proof (*induction I rule: finite-induct*)

case empty thus *?case by (simp add: PsiShape-one)*

next

case (*insert x F*)

have *PsiShape p (f x) using insert.premys by simp*

moreover have *PsiShape p ($\prod i \in F. f i$) using insert.premys by (intro insert.IH) simp*

ultimately show *?case using insert.hyps by (simp add: PsiShape-mult[OF p])*

qed

qed

A factor $p \mid s - p \mid z$ with $v\text{-}p(s) = -1$.

lemma *PsiShape-factor: prime p $\implies$ vp p s = - 1 $\implies$*

*PsiShape p [:of-nat p * s, - of-nat p:]*

proof –

assume *p: prime p and v: vp p s = - 1*

have *pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer)*

have *s0: s $\neq$ 0 using v by (auto simp: vp-def)*

have *p0: (of-nat p :: rat) $\neq$ 0 using prime-gt-0-nat[OF p] by simp*

have *vpp: vp p (of-nat p) = 1 using pe*

by (*simp add: vp-def qmultiplicity-of-nat multiplicity-prime*)

have *c0: vp p (of-nat p * s) = 0*

proof –

have *vp p (of-nat p * s) = vp p (of-nat p) + vp p s using pe p0 s0*

by (*simp add: vp-def qmultiplicity-mult*)

thus *?thesis using vpp v by simp*

qed

have *hi: $\bigwedge n. 1 \leq n \implies$*

*vge p (Polynomial.coeff [:of-nat p * s, - of-nat p:] n) 1*

proof –

fix *n :: nat assume n: 1 $\leq$ n*

show *vge p (Polynomial.coeff [:of-nat p * s, - of-nat p:] n) 1*

proof (*cases n = 1*)

case True thus *?thesis using vge-neg[OF vge-p-self[OF p]] by simp*

next

case False

define *k where k = n - 2*

```

    have nk: n = Suc (Suc k) unfolding k-def using n False by arith
    show ?thesis unfolding nk by (simp add: vge-zero)
  qed
qed
show ?thesis unfolding PsiShape-def using p0 s0 c0 hi by simp
qed

```

Division by an integral monic polynomial preserves " $\equiv 0 \pmod{p}$ " beyond degree $p+1$.

```

lemma degree-div-le-rat: degree (x div y) ≤ degree (x :: rat poly)
proof (cases y = 0 ∨ x div y = 0)
  case True thus ?thesis by auto
next
  case False
  hence y: y ≠ 0 and q: x div y ≠ 0 by auto
  have dq: degree (y * (x div y)) = degree y + degree (x div y) using y q by (rule
degree-mult-eq)
  have degree (y * (x div y) + x mod y) = degree (y * (x div y))
  proof (cases x mod y = 0)
    case True thus ?thesis by simp
  next
    case False
    have degree (x mod y) < degree y using degree-mod-less[OF y, of x] False by
blast
    hence degree (x mod y) < degree (y * (x div y)) using dq by linarith
    thus ?thesis by (rule degree-add-eq-left)
  qed
  hence degree x = degree (y * (x div y)) by (simp only: mult-div-mod-eq)
  thus ?thesis using dq by simp
qed

```

lemma *div-lead1-high*:

```

assumes p: prime p and R0: vpGge p R 0 and R1:  $\bigwedge d. p + 1 < d \implies$ 
  vge p (Polynomial.coeff R d) 1
  and Q1: lead-coeff Q = 1 and Q0: vpGge p Q 0 and d: p + 1 < d
shows vge p (Polynomial.coeff (R div Q) d) 1
proof -
  define Rlo where Rlo = ( $\sum i < p + 2. \text{Polynomial.monom } (\text{Polynomial.coeff } R i) i$ )
  define Rhi where Rhi = Polynomial.smult (1 / of-nat p) (R - Rlo)
  have coeff-Rlo:  $\bigwedge i. \text{Polynomial.coeff } Rlo i = (\text{if } i < p + 2 \text{ then } \text{Polynomial.coeff } R i \text{ else } 0)$ 
  unfolding Rlo-def by (simp add: coeff-sum coeff-monom)
  have split: R = Rlo + Polynomial.smult (of-nat p) Rhi
  proof -
    have (of-nat p :: rat) ≠ 0 using prime-gt-0-nat[OF p] by simp
    thus ?thesis unfolding Rhi-def by simp
  qed
  have Rhi0: vpGge p Rhi 0

```

```

    unfolding vpGge-iff
  proof
    fix i
    show vge p (Polynomial.coeff Rhi i) 0
  proof (cases i < p + 2)
    case True thus ?thesis by (simp add: Rhi-def coeff-Rlo vge-zero)
  next
    case False
    hence Polynomial.coeff Rhi i = 1 / of-nat p * Polynomial.coeff R i
      by (simp add: Rhi-def coeff-Rlo)
    moreover have vge p (1 / of-nat p * Polynomial.coeff R i) (- 1 + 1)
      by (rule vge-mul[OF p vge-inv-p[OF p] R1]) (use False in simp)
    ultimately show ?thesis by simp
  qed
  qed
  have deg: degree (Rlo div Q) ≤ p + 1
  proof -
    have degree Rlo ≤ p + 1 by (rule degree-le) (simp add: coeff-Rlo)
    thus ?thesis using degree-div-le-rat[of Rlo Q] by linarith
  qed
  have R div Q = Rlo div Q + Polynomial.smult (of-nat p) (Rhi div Q)
    by (subst split) (simp add: poly-div-add-left div-smult-left)
  hence Polynomial.coeff (R div Q) d = of-nat p * Polynomial.coeff (Rhi div Q)
d
    using deg d by (simp add: coeff-eq-0)
  thus ?thesis
    using vge-mul[OF p vge-p-self[OF p] vpGge-div-monic[OF p Rhi0 Q1 Q0,
unfolded vpGge-iff,
    rule-format, of d]] by simp
  qed

lemma card-le-prime-of-distinct-mod:
  assumes prime p finite T  $\bigwedge r s. r \in T \implies s \in T \implies$ 
     $r \neq s \implies \neg \text{int } p \text{ dvd } (r - s)$ 
  shows card T ≤ p
  proof -
    have pp: 0 < int p using prime-gt-0-nat[OF assms(1)] by simp
    have inj: inj-on (λr. nat (r mod int p)) T
    proof (rule inj-onI)
      fix r s assume r: r ∈ T and s: s ∈ T and e: nat (r mod int p) = nat (s mod
int p)
      have r mod int p = s mod int p using e pp by (simp add: eq-nat-nat-iff
pos-mod-sign)
      hence int p dvd (r - s) by (simp add: mod-eq-dvd-iff)
      thus r = s using assms(3)[OF r s] by blast
    qed
    have card T = card ((λr. nat (r mod int p)) ‘ T) by (rule card-image[OF inj,
symmetric])
    also have ... ≤ card {..
```

```

    by (rule card-mono) (use pp in ⟨auto simp: nat-less-iff pos-mod-sign
      pos-mod-bound⟩)
  finally show ?thesis by simp
qed

lemma vpGge-qPoleProd-int:
  prime p  $\implies$  finite A  $\implies$  A  $\subseteq$  of-int ' T  $\implies$ 
    vpGge p (qPoleProd A) 0
proof -
  assume p: prime p and A: finite A and sub: A  $\subseteq$  of-int ' T
  have vpGge p ( $\prod$  s  $\in$  A. pX - [:s:]) ( $\sum$  s  $\in$  A. 0)
  proof (rule vpGge-prod[OF p A])
    fix s assume s  $\in$  A
    then obtain r where s: s = of-int r using sub by auto
    show vpGge p (pX - [:s:]) 0 unfolding s diff-conv-add-uminus
    by (rule vpGge-add[OF p vpGge-pX vpGge-neg[OF vpGge-C[OF vge-of-int[OF
      p]]]])
  qed
  thus ?thesis by (simp add: qPoleProd-def)
qed

```

The analytic part is integral (Lean: *pt-far-vge*): every κ -d f -d is p -integral, f -d the Taylor coefficients of $Q + \sum -\{s \in F\} \varrho \cdot s / (z - s)$.

```

lemma far-vge:
  assumes p: prime p and p7: 7  $\leq$  p and T: finite T and F: finite F
  and hT:  $\bigwedge r$  s. r  $\in$  T  $\implies$  s  $\in$  T  $\implies$ 
    r  $\neq$  s  $\implies$   $\neg$  int p dvd (r - s)
  and hF:  $\bigwedge s$ . s  $\in$  F  $\implies$  vp p s = - 1
  and U0: vpGge p U 0 and U1:  $\bigwedge d$ . p + 1 < d  $\implies$ 
    vge p (Polynomial.coeff U d) 1
  shows vge p (kappa d * fCoeff (([:(- of-nat p) powi (- int (card F)):] *
    U) div qPoleProd (of-int ' T  $\cup$  F))
    F (qRes (of-int ' T  $\cup$  F) ([:(- of-nat p) powi (- int (card F)):] *
      U)) d) 0
proof -
  define Tq :: rat set where Tq = of-int ' T
  define S where S = Tq  $\cup$  F
  define c where c = ((- of-nat p) powi (- int (card F)) :: rat)
  define U' where U' = [:c:] * U
  define Q where Q = U' div qPoleProd S
  define ϱ where ϱ = qRes S U'
  have finT: finite Tq and finS: finite S by (simp-all add: Tq-def S-def T F)
  have disj: Tq  $\cap$  F = {} unfolding Tq-def by (rule far-disj[OF p hF])
  have F0:  $\bigwedge s$ . s  $\in$  F  $\implies$  s  $\neq$  0
  proof -
    fix s assume s  $\in$  F
    hence vp p s = - 1 by (rule hF)
    thus s  $\neq$  0 by (auto simp: vp-def)
  qed

```

```

have p0: (of-nat p :: rat) ≠ 0 using p by (simp add: prime-gt-0-nat)
— the Lagrange decomposition, split into near and far poles
define Mq where Mq = qPoleProd F * Q + (∑ s ∈ F. [:ρ s:] * qPoleProd (F
- {s}))
define W0 where W0 = (∑ r ∈ Tq. [:ρ r:] * qPoleProd (Tq - {r}))
have Lag: U' = qPoleProd Tq * Mq + qPoleProd F * W0
proof -
  have qU: ∧ A B. finite A ⇒ finite B ⇒
    A ∩ B = {} ⇒ qPoleProd (A ∪ B) = qPoleProd A * qPoleProd B
  unfolding qPoleProd-def by (rule prod.union-disjoint)
  have L: U' = qPoleProd S * Q + (∑ s ∈ S. [:ρ s:] * qPoleProd (S - {s}))
  unfolding Q-def ρ-def by (rule lagrange-decomp[OF finS])
  have P: qPoleProd S = qPoleProd Tq * qPoleProd F unfolding S-def by (rule
qU[OF finT F disj])
  have sumS: (∑ s ∈ S. [:ρ s:] * qPoleProd (S - {s})) =
    (∑ s ∈ Tq. [:ρ s:] * qPoleProd (S - {s})) + (∑ s ∈
    F. [:ρ s:] * qPoleProd (S - {s}))
  unfolding S-def by (rule sum.union-disjoint[OF finT F disj])
  have sT: (∑ s ∈ Tq. [:ρ s:] * qPoleProd (S - {s})) = qPoleProd F * W0
  proof -
    have (∑ s ∈ Tq. [:ρ s:] * qPoleProd (S - {s})) = (∑ r ∈
    Tq. qPoleProd F * ([:ρ r:] * qPoleProd (Tq - {r})))
    proof (rule sum.cong[OF refl])
      fix r assume r: r ∈ Tq
      have S - {r} = (Tq - {r}) ∪ F using r disj unfolding S-def by auto
      hence qPoleProd (S - {r}) = qPoleProd (Tq - {r}) * qPoleProd F
      using qU[of Tq - {r} F] finT F disj by auto
      thus [:ρ r:] * qPoleProd (S - {r}) =
        qPoleProd F * ([:ρ r:] * qPoleProd (Tq - {r})) by (simp add: ac-simps)
    qed
    thus ?thesis unfolding W0-def by (simp add: sum-distrib-left)
  qed
  have sF: (∑ s ∈ F. [:ρ s:] * qPoleProd (S - {s})) =
    qPoleProd Tq * (∑ s ∈ F. [:ρ s:] * qPoleProd (F - {s}))
  proof -
    have (∑ s ∈ F. [:ρ s:] * qPoleProd (S - {s})) = (∑ s ∈
    F. qPoleProd Tq * ([:ρ s:] * qPoleProd (F - {s})))
    proof (rule sum.cong[OF refl])
      fix s assume s: s ∈ F
      have S - {s} = Tq ∪ (F - {s}) using s disj unfolding S-def by auto
      hence qPoleProd (S - {s}) = qPoleProd Tq * qPoleProd (F - {s})
      using qU[of Tq F - {s}] finT F disj by auto
      thus [:ρ s:] * qPoleProd (S - {s}) =
        qPoleProd Tq * ([:ρ s:] * qPoleProd (F - {s})) by (simp add: ac-simps)
    qed
    thus ?thesis by (simp add: sum-distrib-left)
  qed
  show ?thesis unfolding Mq-def by (subst L) (simp only: P sumS sT sF
algebra-simps)

```

```

qed
— clearing the constant c
define kap where kap = ((- of-nat p) ^ card F :: rat)
have kc: kap * c = 1
  unfolding kap-def c-def power-int-minus power-int-of-nat using p0 by simp
define Psi where Psi = (∏ s ∈ F. [:of-nat p * s, - of-nat p:])
have Psi-eq: Psi = [:kap:] * qPoleProd F
proof -
  have f: ∧s. [:of-nat p * s, - of-nat p:] = [: - of-nat p:] * (pX - [:s:])
    unfolding pX-minus-const by simp
  have Psi = (∏ s ∈ F. [: - of-nat p:] * (pX - [:s:])) unfolding Psi-def f ..
  also have ... = [: - of-nat p:] ^ card F * qPoleProd F
    by (simp only: prod.distrib prod-constant qPoleProd-def)
  also have ... = [:kap:] * qPoleProd F by (simp only: kap-def const-poly-power)
  finally show ?thesis .
qed
define Nq where Nq = [:kap:] * Mq
have Ueq: U = qPoleProd Tq * Nq + Psi * W0
proof -
  have U = [:kap:] * U' unfolding U'-def mult.assoc[symmetric] using kc
    by (simp add: mult.commute[of c kap])
  also have ... = qPoleProd Tq * Nq + Psi * W0 unfolding Lag Nq-def Psi-eq
    by (simp add: algebra-simps)
  finally show ?thesis .
qed
have Ndiv: Nq = (U - Psi * W0) div qPoleProd Tq
proof -
  have nz: qPoleProd Tq ≠ 0 using qPoleProd-monic[OF finT] by auto
  have e: U - Psi * W0 = qPoleProd Tq * Nq using Ueq by simp
  show ?thesis unfolding e using nz by simp
qed
— integrality
have ρT: ∧r. r ∈ Tq ⇒ vge p (ρ r) 0
proof -
  fix r assume r ∈ Tq
  then obtain r' where r': r' ∈ T r = of-int r' unfolding Tq-def by auto
  show vge p (ρ r) 0 unfolding r'(2) ρ-def S-def U'-def c-def Tq-def
    by (rule near-res-vge[OF p T F hT hF U0 r'(1)])
qed
have W0i: vpGge p W0 0
  unfolding W0-def
proof (rule vpGge-sum[OF p finT])
  fix r assume r: r ∈ Tq
  have vpGge p ([:ρ r:] * qPoleProd (Tq - {r})) (0 + 0)
    by (rule vpGge-mul[OF p vpGge-C[OF ρT[OF r]] vpGge-qPoleProd-int[OF
p]]) (use finT in
  ⟨auto simp: Tq-def⟩)
  thus vpGge p ([:ρ r:] * qPoleProd (Tq - {r})) 0 by simp
qed

```



```

have shape: PsiShape p Psi
  unfolding Psi-def by (rule PsiShape-prod[OF p F]) (rule PsiShape-factor[OF
p hF])
have Psii: vpGge p Psi 0 by (rule PsiShape-vpGge[OF p shape])
have Ri: vpGge p (U - Psi * W0) 0
proof -
  have vpGge p (Psi * W0) 0 using vpGge-mul[OF p Psii W0i] by simp
  thus ?thesis unfolding diff-conv-add-uminus by (rule vpGge-add[OF p U0
vpGge-neg])
qed
have W0deg:  $\bigwedge j. p \leq j \implies \text{Polynomial.coeff } W0 \ j = 0$ 
proof -
  fix j assume j:  $p \leq j$ 
  have cT:  $\text{card } Tq \leq p$  unfolding Tq-def
    using card-image-le[OF T, of of-int :: int  $\Rightarrow$  rat]
    card-le-prime-of-distinct-mod[OF p T] hT by fastforce
  have z:  $\bigwedge r. r \in Tq \implies \text{Polynomial.coeff } ([:_\rho \ r:] * qPoleProd (Tq - \{r\})) \ j = 0$ 
  proof -
    fix r assume r:  $r \in Tq$ 
    have degree  $([:_\rho \ r:] * qPoleProd (Tq - \{r\})) \leq \text{degree } (qPoleProd (Tq - \{r\}))$ 
    {r}))
    by (simp add: degree-smult-le)
  also have ... =  $\text{card } (Tq - \{r\})$  using finT by (simp add: degree-qPoleProd)
  finally have dg:  $\text{degree } ([:_\rho \ r:] * qPoleProd (Tq - \{r\})) < p$ 
    using card-Diff1-less[OF finT r] cT by linarith
  show  $\text{Polynomial.coeff } ([:_\rho \ r:] * qPoleProd (Tq - \{r\})) \ j = 0$ 
    by (rule coeff-eq-0) (use dg j in linarith)
  qed
show  $\text{Polynomial.coeff } W0 \ j = 0$  unfolding W0-def coeff-sum
  by (rule sum.neutral) (use z in blast)
qed
have R1:  $\bigwedge d. p + 1 < d \implies \text{vge } p (\text{Polynomial.coeff } (U - Psi * W0) \ d) \ 1$ 
proof -
  fix d assume d:  $p + 1 < d$ 
  have sh:  $\bigwedge n. 1 \leq n \implies \text{vge } p (\text{Polynomial.coeff } Psi \ n) \ 1$  using shape
    unfolding PsiShape-def by blast
  have vge p ( $\text{Polynomial.coeff } (Psi * W0) \ d$ ) 1 unfolding coeff-mult
  proof (rule vge-sum[OF p finite-atMost])
    fix i assume i:  $i \in \{..d\}$ 
    show  $\text{vge } p (\text{Polynomial.coeff } Psi \ i * \text{Polynomial.coeff } W0 \ (d - i)) \ 1$ 
    proof (cases i = 0)
      case True
        have  $\text{Polynomial.coeff } W0 \ (d - i) = 0$  using True d by (intro W0deg)
      case False
        thus ?thesis by (simp add: vge-zero)
    next
      case False
        have q:  $\text{vge } p (\text{Polynomial.coeff } Psi \ i) \ 1$  by (rule sh) (use False in simp)

```

```

      have vge p (Polynomial.coeff Psi i * Polynomial.coeff W0 (d - i)) (1 + 0)
        by (rule vge-mul[OF p q]) (use W0i in ‹simp add: vpGge-iff›)
      thus ?thesis by simp
    qed
  qed
  thus vge p (Polynomial.coeff (U - Psi * W0) d) 1 unfolding coeff-diff
    by (rule vge-diff[OF p U1[OF d]])
  qed
  have TqT: Tq ⊆ of-int ‘ T by (simp add: Tq-def)
  have Ni: vpGge p Nq 0
    unfolding Ndiv by (rule vpGge-div-monic[OF p Ri qPoleProd-monic[OF finT]
vpGge-qPoleProd-int[OF
p finT TqT]])
  have N1: ∧ d. p + 1 < d ⇒ vge p (Polynomial.coeff Nq d) 1
    unfolding Ndiv by (rule div-lead1-high[OF p Ri R1 qPoleProd-monic[OF finT]
vpGge-qPoleProd-int[OF p finT TqT]])
  — the power series identity
  have mul: fps-of-poly Psi * Abs-fps (fCoeff Q F ϱ) = fps-of-poly Nq
    unfolding Psi-eq Nq-def Mq-def fps-of-poly-mult mult.assoc
    by (subst qPoleProd-mul-fCoeff[OF F F0]) simp-all
  have f: vge p (fCoeff Q F ϱ d) 0 ∧ (p + 1 < d ⇒
    vge p (fCoeff Q F ϱ d) 1)
    using shape unfolding PsiShape-def by (intro series-vge[OF p - - - Ni N1
mul]) blast+
  — the factor κ-d
  have vge p (kappa d * fCoeff Q F ϱ d) 0
  proof (cases p + 1 < d)
    case True
      have k: vge p (kappa d) (- 1)
        using kappa-padicVal-ge[OF p, of d] p7 by (auto simp: vge-def)
      show ?thesis using vge-mul[OF p k conjunct2[OF f, rule-format, OF True]]
    by simp
  next
    case False
      have k: vge p (kappa d) 0
        using kappa-integral[OF p p7, of d] False by (auto simp: vge-def)
      show ?thesis using vge-mul[OF p k conjunct1[OF f]] by simp
  qed
  thus ?thesis by (simp add: Q-def ϱ-def U'-def S-def Tq-def c-def)
  qed

```

19.4 Lemma 3.1 with far poles, truncated rational form

lemma *tauPoleP-tr-near*: $\text{tauPoleP-tr } p \ Y \ (\text{of-int } r) \ M = [:H5 \ (dIdx \ r) :] - Y$
 by (simp add: tauPoleP-tr-def)

lemma *far-denom-ne-1*: $\text{prime } p \Rightarrow \text{vp } p \ s = - 1 \Rightarrow$
 $\text{snd } (\text{quotient-of } s) \neq 1$

proof —

```

assume  $p$ : prime  $p$  and  $v$ :  $vp\ p\ s = -\ 1$ 
show ?thesis
proof
  assume  $d1$ :  $snd\ (quotient-of\ s) = 1$ 
  obtain  $n\ d$  where  $q$ :  $quotient-of\ s = (n, d)$  by (cases quotient-of  $s$ )
  hence  $s = of-int\ n / of-int\ d$  by (rule quotient-of-div)
  with  $d1\ q$  have  $s = of-int\ n$  by simp
  thus False using far-ne-int[ $OF\ p\ v, of\ n$ ] by simp
qed
qed

```

```

lemma tauPoleP-tr-far: prime  $p \implies vp\ p\ s = -\ 1 \implies$ 
  tauPoleP-tr  $p\ Y\ s\ M = [:tauAnPole-tr\ p\ s\ M:]$ 
by (simp add: tauPoleP-tr-def far-denom-ne-1)

```

```

lemma vge-H5-small: prime  $p \implies n < p \implies vge\ p\ (H5\ n)\ 0$ 
proof -
  assume  $p$ : prime  $p$  and  $n$ :  $n < p$ 
  show ?thesis unfolding H5-def
  proof (rule vge-sum[ $OF\ p\ finite-atLeastAtMost$ ])
    fix  $v$  assume  $v$ :  $v \in \{1..n\}$ 
    have  $nd$ :  $\neg\ p\ dvd\ v$  using  $v\ n$  by (auto dest: dvd-imp-le)
    have  $vge\ p\ (inverse\ (of-nat\ v) ^ 5)\ 0$  by (rule vge-pow[ $OF\ p\ vge-inverse-unit[OF\ p\ nd]$ ])
    thus  $vge\ p\ (1 / of-nat\ v ^ 5)\ 0$  by (simp add: inverse-eq-divide power-one-over)
  qed
qed

```

Lemma 3.1 with far poles, rational form (Lean: lemma-3-1-rat): with Y integral, near poles T distinct modulo p and $d(r) < p$, far poles F with $v\text{-}p(s) = -1$, numerator $(-p) \sim \{-|F|\}$ U with U integral and $\equiv 0 \pmod{p}$ beyond degree $p+1$, the truncated $\tau \sim \{ext\}$ - Y is integral. The truncation must cover the polynomial part: $\deg U \leq M$.

```

lemma sum-const-poly:  $(\sum\ i \in I. [f\ i:]) = [: \sum\ i \in I. f\ i:]$ 
by (induction  $I$  rule: infinite-finite-induct) simp-all

```

theorem tauX-far-pole-valuation:

```

assumes  $p$ : prime  $p$  and  $p7$ :  $7 \leq p$  and  $T$ : finite  $T$  and  $F$ : finite  $F$ 
and  $hT$ :  $\bigwedge r\ s. r \in T \implies s \in T \implies$ 
   $r \neq s \implies \neg\ int\ p\ dvd\ (r - s)$  and  $hd$ :  $\bigwedge r. r \in$ 
   $T \implies dIdx\ r < p$ 
and  $hF$ :  $\bigwedge s. s \in F \implies vp\ p\ s = -\ 1$ 
and  $Y$ :  $vpGge\ p\ Y\ 0$ 
and  $U0$ :  $vpGge\ p\ U\ 0$  and  $U1$ :  $\bigwedge d. p + 1 < d \implies$ 
   $vge\ p\ (Polynomial.coeff\ U\ d)\ 1$ 
and  $dU$ :  $\deg\ U \leq M$ 
shows  $vpGge\ p\ (tauExt-tr\ p\ Y\ ([:(-\ of-nat\ p)\ powi\ (-\ int\ (card\ F)):] *$ 
   $U)\ (of-int\ 'T \cup F)\ M)\ 0$ 
proof -

```

```

define  $Tq :: \text{rat set}$  where  $Tq = \text{of-int } T$ 
define  $S$  where  $S = Tq \cup F$ 
define  $c$  where  $c = ((- \text{ of-nat } p) \text{ powi } (- \text{ int } (\text{card } F))) :: \text{rat}$ 
define  $U'$  where  $U' = [:c:] * U$ 
define  $Q$  where  $Q = U' \text{ div } q\text{PoleProd } S$ 
define  $\varrho$  where  $\varrho = q\text{Res } S \ U'$ 
have  $\text{disj}: Tq \cap F = \{\}$  unfolding  $Tq\text{-def}$  by  $(\text{rule far-disj}[OF \ p \ hF])$ 
have  $\text{inj}: \text{inj-on } (\text{of-int} :: \text{int} \Rightarrow \text{rat}) \ T$  by  $(\text{simp add: inj-on-def})$ 
have  $dQ: \text{degree } Q < M + 3$ 
proof -
  have  $\text{degree } Q \leq \text{degree } U'$  unfolding  $Q\text{-def}$  by  $(\text{rule degree-div-le-rat})$ 
  also have  $\dots \leq \text{degree } U$  unfolding  $U'\text{-def}$  by  $(\text{simp add: degree-smult-le})$ 
  finally show  $?thesis$  using  $dU$  by  $\text{linarith}$ 
qed
have  $\text{Ext}: \text{tauExt-tr } p \ Y \ U' \ S \ M =$ 
   $[:\text{tauK } Q:] + (\sum r \in T. [: \varrho (\text{of-int } r):] * ([:H5 \ (dIdx \ r):] - Y)) +$ 
   $(\sum s \in F. [: \varrho \ s * \text{tauAnPole-tr } p \ s \ M:])$ 
proof -
  have  $\text{finTq}: \text{finite } Tq$  using  $T$  by  $(\text{simp add: } Tq\text{-def})$ 
  have  $e1: \text{tauExt-tr } p \ Y \ U' \ S \ M = [: \text{tauK } Q:] + (\sum s \in$ 
     $S. [: \varrho \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M)$ 
  unfolding  $\text{tauExt-tr-def } Q\text{-def } \varrho\text{-def} \dots$ 
  have  $e2: (\sum s \in S. [: \varrho \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M) =$ 
     $(\sum s \in Tq. [: \varrho \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M) + (\sum s \in$ 
     $F. [: \varrho \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M)$ 
  unfolding  $S\text{-def}$  by  $(\text{rule sum.union-disjoint}[OF \ \text{finTq } F \ \text{disj}])$ 
  have  $e3: (\sum s \in Tq. [: \varrho \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M) = (\sum r \in$ 
     $T. [: \varrho (\text{of-int } r):] * ([:H5 \ (dIdx \ r):] - Y))$ 
  unfolding  $Tq\text{-def } \text{sum.reindex}[OF \ \text{inj}]$  by  $(\text{simp add: tauPoleP-tr-near})$ 
  have  $e4: (\sum s \in F. [: \varrho \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M) = (\sum s \in$ 
     $F. [: \varrho \ s * \text{tauAnPole-tr } p \ s \ M:])$ 
  by  $(\text{rule sum.cong}[OF \ \text{refl}])$   $(\text{simp add: tauPoleP-tr-far}[OF \ p \ hF])$ 
  show  $?thesis$  unfolding  $e1 \ e2 \ e3 \ e4$  by  $(\text{simp only: add.assoc})$ 
qed
have  $\text{analytic}: \text{tauK } Q + (\sum s \in F. \varrho \ s * \text{tauAnPole-tr } p \ s \ M) = (\sum d <$ 
   $M + 3. \text{kappa } d * \text{fCoeff } Q \ F \ \varrho \ d)$ 
proof -
  have  $t: \text{tauK } Q = (\sum d < M + 3. \text{Polynomial.coef} \ Q \ d * \text{kappa } d)$ 
proof -
  have  $\text{tauK } Q = (\sum d \in \{.. \text{degree } Q\}. \text{Polynomial.coef} \ Q \ d * \text{kappa } d)$ 
  by  $(\text{simp add: tauK-def})$ 
  also have  $\dots = (\sum d \in \{.. < M + 3\}. \text{Polynomial.coef} \ Q \ d * \text{kappa } d)$ 
  by  $(\text{rule sum.mono-neutral-left})$   $(\text{use } dQ \text{ in } \langle \text{auto simp: coef-eq-0} \rangle)$ 
  finally show  $?thesis$  .
qed
have  $(\sum s \in F. \varrho \ s * \text{tauAnPole-tr } p \ s \ M) = - (\sum s \in F. \sum d <$ 
   $M + 3. \text{kappa } d * (\varrho \ s * \text{inverse } s \wedge (d + 1)))$ 
  unfolding  $\text{tauAnPole-tr-eq-sum}$  by  $(\text{simp add: sum-distrib-left sum-negf}$ 
   $\text{mult-ac})$ 

```

```

    also have (∑ s ∈ F. ∑ d < M + 3. kappa d * (ρ s * inverse s ^ (d + 1)))
  =
    (∑ d < M + 3. ∑ s ∈ F. kappa d * (ρ s * inverse s ^ (d + 1)))
    by (rule sum.swap)
  finally have a: (∑ s ∈ F. ρ s * tauAnPole-tr p s M) = - (∑ d <
    M + 3. ∑ s ∈ F. kappa d * (ρ s * inverse s ^ (d + 1))) .
  show ?thesis unfolding t a fCoeff-def
    by (simp add: right-diff-distrib sum-subtractf sum-distrib-left mult-ac)
  qed
  have an-int: vge p (∑ d < M + 3. kappa d * fCoeff Q F ρ d) 0
    unfolding Q-def ρ-def U'-def S-def Tq-def c-def
    by (intro vge-sum[OF p] finite-lessThan) (blast intro: far-vge[OF p p7 T F hT
hF U0 U1])
  have ρT: ⋀r. r ∈ T ⇒ vge p (ρ (of-int r)) 0
    unfolding ρ-def U'-def S-def Tq-def c-def by (blast intro: near-res-vge[OF p
T F hT hF U0])
  have near-int: ⋀r. r ∈ T ⇒
    vpGge p ([:ρ (of-int r):] * ([:H5 (dIdx r):] - Y)) 0
  proof -
    fix r assume r: r ∈ T
    have h: vpGge p ([:H5 (dIdx r):] - Y) 0 unfolding diff-conv-add-uminus
    by (rule vpGge-add[OF p vpGge-C[OF vge-H5-small[OF p hd[OF r]]] vpGge-neg[OF
Y]])
    show vpGge p ([:ρ (of-int r):] * ([:H5 (dIdx r):] - Y)) 0
      using vpGge-mul[OF p vpGge-C[OF ρT[OF r]] h] by simp
  qed
  have const: [:tauK Q:] + (∑ s ∈ F. [:ρ s * tauAnPole-tr p s M:]) = [:∑ d <
    M + 3. kappa d * fCoeff Q F ρ d:]
    unfolding sum-const-poly analytic[symmetric] by simp
  have e: [:tauK Q:] + (∑ r ∈ T. [:ρ (of-int r):] * ([:H5 (dIdx r):] - Y)) +
    (∑ s ∈ F. [:ρ s * tauAnPole-tr p s M:]) =
    [:∑ d < M + 3. kappa d * fCoeff Q F ρ d:] + (∑ r ∈
    T. [:ρ (of-int r):] * ([:H5 (dIdx r):] - Y))
    unfolding const[symmetric] by (simp only: ac-simps)
  have vpGge p (tauExt-tr p Y U' S M) 0
    unfolding Ext e by (rule vpGge-add[OF p vpGge-C[OF an-int] vpGge-sum[OF
p T near-int]])
  thus ?thesis by (simp add: U'-def S-def Tq-def c-def)
  qed
end

theory Zeta5-Pole-Distribution
  imports Zeta5-Truncated-Expansions
begin

```

20 The distribution formula (3.7) and Lemma 3.2

Section 3.2 of the paper (Lean theory *Distribution*): a Raabe-type multiplication formula for τ on polynomials, the sum over the residue classes $a < p$ of the truncated pole values (by induction on the pole, with the truncation error controlled), compatibility of partial fractions with the substitution $x = a + p z$, and Lemma 3.2 (*tauX-distribution-formula*).

20.1 Raabe's multiplication formula

lemma *Lfun-uminus*: $Lfun (- P) = - Lfun P$
using *Lfun-smult[of - 1 P]* **by** *simp*

lemma *Lfun-sub*: $Lfun (P - Q) = Lfun P - Lfun Q$
by (*simp only: diff-conv-add-uminus Lfun-add Lfun-uminus*)

lemma *Lfun-zero*: $Lfun 0 = 0$
using *Lfun-smult[of 0 0]* **by** *simp*

lemma *Lfun-sum*: $Lfun (\sum i \in I. f i) = (\sum i \in I. Lfun (f i))$
by (*induction I rule: infinite-finite-induct*) (*simp-all add: Lfun-add Lfun-zero*)

lemma *sum-const-poly*: $(\sum i \in I. [:f i:]) = [: \sum i \in I. f i:]$
by (*induction I rule: infinite-finite-induct*) *simp-all*

lemma *Lfun-pX-pow*: $Lfun (pX \wedge n) = bernoulli n$
by (*simp only: pX-pow-monom Lfun-monom mult-1*)

lemma *Lfun-fwd-monom*:
 $Lfun (pcompose (Polynomial.monom c n) (pX + 1) - Polynomial.monom c n)$
 $= (if n =$
 $1 then c else 0)$

proof –

have *m*: $Polynomial.monom c n = Polynomial.smult c (pX \wedge n)$
unfolding *pX-pow-monom* **by** (*simp add: smult-monom*)
have $Lfun (pcompose (Polynomial.monom c n) (pX + 1) - Polynomial.monom c n)$
 $= c * Lfun (pcompose (pX \wedge n) (pX + 1) - pX \wedge n)$
unfolding *m pcompose-smult smult-diff-right[symmetric]* *Lfun-smult ..*
also have $Lfun (pcompose (pX \wedge n) (pX + 1) - pX \wedge n) = (\sum k <$
 $n. of-nat (n choose k) * bernoulli k)$
unfolding *pX-shift-pow-diff Lfun-sum Lfun-monom ..*
finally show *?thesis* **unfolding** *sum-binomial-bernoulli-lt* **by** *simp*
qed

$L(f(x+1) - f(x)) = f'(0)$: the defining recursion of the Bernoulli numbers.

lemma *Lfun-fwd*: $Lfun (pcompose f (pX + 1) - f) = poly (pderiv f) 0$
proof –

```

define  $D$  where  $D = \text{degree } f$ 
define  $M$  where  $M\ n = \text{Polynomial.monom } (\text{Polynomial.coeff } f\ n)\ n$  for  $n$ 
have  $e$ :  $(\sum n \leq D. M\ n) = f$  unfolding  $D\text{-def } M\text{-def}$  by  $(\text{rule poly-as-sum-of-monoms})$ 
have  $Lfun\ (pcompose\ f\ (pX + 1) - f) = Lfun\ (pcompose\ (\sum n \leq D. M\ n)\ (pX + 1) - (\sum n \leq D. M\ n))$ 
by  $(\text{simp only: } e)$ 
also have  $\dots = Lfun\ (\sum n \leq D. (pcompose\ (M\ n)\ (pX + 1) - M\ n))$ 
by  $(\text{simp only: } pcompose\text{-sum sum-subtractf})$ 
also have  $\dots = (\sum n \leq D. \text{if } n = 1 \text{ then } \text{Polynomial.coeff } f\ n \text{ else } 0)$ 
by  $(\text{simp only: } Lfun\text{-sum } M\text{-def } Lfun\text{-fwd-monom})$ 
also have  $\dots = \text{Polynomial.coeff } f\ 1$ 
by  $(\text{cases } 1 \leq D)\ (\text{auto simp: } D\text{-def coeff-eq-0})$ 
also have  $\dots = \text{poly } (pderiv\ f)\ 0$ 
by  $(\text{simp add: poly-0-coeff-0 coeff-pderiv})$ 
finally show  $?thesis$  .
qed

```

```

lemma  $pderiv\text{-}pcompose\text{-affine}$ :
 $pderiv\ (pcompose\ Q\ ([:a:] + [:c:] * pX)) =$ 
 $\text{Polynomial.smult } c\ (pcompose\ (pderiv\ Q)\ ([:a:] + [:c:] * pX))$ 
proof –
have  $d$ :  $pderiv\ ([:a:] + [:c:] * pX) = [:c:]$ 
by  $(\text{simp add: } pderiv\text{-add } pderiv\text{-smult } pderiv\text{-monom } pderiv\text{-pCons})$ 
show  $?thesis$  unfolding  $pderiv\text{-}pcompose\ d$  by  $(\text{simp add: mult.commute})$ 
qed

```

```

lemma  $Lfun\text{-raabe}$ :
 $(\sum a < p. Lfun\ (pcompose\ f\ ([:of\text{-nat } a:] + [:of\text{-nat } p:] * pX))) = of\text{-nat } p * Lfun\ f$ 
proof –
obtain  $g$  where  $g$ :  $pcompose\ g\ (pX + 1) - g = f$  using  $\text{exists-antidiff}$  by  $\text{blast}$ 
define  $G$  where  $G = pcompose\ g\ ([:of\text{-nat } p:] * pX)$ 
have  $e$ :  $\bigwedge a. pcompose\ f\ ([:of\text{-nat } a:] + [:of\text{-nat } p:] * pX) =$ 
 $pcompose\ g\ ([:of\text{-nat } (Suc\ a):] + [:of\text{-nat } p:] * pX) - pcompose\ g\ ([:of\text{-nat } a:] +$ 
 $[:of\text{-nat } p:] * pX)$ 
proof –
fix  $a :: nat$ 
have  $r$ :  $pcompose\ (pX + 1)\ ([:of\text{-nat } a:] + [:of\text{-nat } p:] * pX) =$ 
 $[:of\text{-nat } (Suc\ a):] + [:of\text{-nat } p:] * pX$ 
unfolding  $pcompose\text{-add } pcompose\text{-}pX\ pcompose\text{-}1$  by  $(\text{simp add: one-pCons})$ 
show  $pcompose\ f\ ([:of\text{-nat } a:] + [:of\text{-nat } p:] * pX) =$ 
 $pcompose\ g\ ([:of\text{-nat } (Suc\ a):] + [:of\text{-nat } p:] * pX) - pcompose\ g\ ([:of\text{-nat } a:] +$ 
 $[:of\text{-nat } p:] * pX)$ 
unfolding  $g[\text{symmetric}]\ pcompose\text{-diff } pcompose\text{-assoc}[\text{symmetric}]\ r$  ..
qed
have  $tel$ :  $(\sum a < p. Lfun\ (pcompose\ f\ ([:of\text{-nat } a:] + [:of\text{-nat } p:] * pX))) =$ 
 $Lfun\ (pcompose\ g\ ([:of\text{-nat } p:] + [:of\text{-nat } p:] * pX)) - Lfun\ (pcompose\ g$ 

```

```

([:0:] +
  [:of-nat p:] * pX))
  unfolding e Lfun-sub
  using sum-lessThan-telescope[of  $\lambda a. Lfun (pcompose\ g\ ([:of-nat\ a:] + [:of-nat\ p:] * pX))\ p]$  by simp
  have h1: pcompose g ([:of-nat p:] + [:of-nat p:] * pX) = pcompose G (pX + 1)
  and h2: pcompose g ([:0:] + [:of-nat p:] * pX) = G
  proof -
    have c: pcompose ([:of-nat p:] * pX) (pX + 1) = [:of-nat p:] + [:of-nat p:] * pX
    unfolding pcompose-mult pcompose-pX pcompose-const by (simp add: algebra-simps)
    show pcompose g ([:of-nat p:] + [:of-nat p:] * pX) = pcompose G (pX + 1)
    unfolding G-def pcompose-assoc[symmetric] c ..
    show pcompose g ([:0:] + [:of-nat p:] * pX) = G
    unfolding G-def by simp
  qed
  have Lfun (pcompose G (pX + 1)) - Lfun G = poly (pderiv G) 0
  by (simp only: Lfun-sub[symmetric] Lfun-fwd)
  also have ... = of-nat p * poly (pderiv g) 0
  unfolding G-def using pderiv-pcompose-affine[of g 0 of-nat p]
  by (simp add: poly-pcompose poly-monom)
  also have poly (pderiv g) 0 = Lfun f
  by (simp only: g[symmetric] Lfun-fwd)
  finally show ?thesis using tel h1 h2 by simp
qed

```

lemma tau-raabe:

```

( $\sum a < p. \tau (pcompose\ Q\ ([:of-nat\ a:] + [:of-nat\ p:] * pX))$ ) = of-nat p ^ 4 *
tau Q
  proof -
    have e:  $\bigwedge a. \tau (pcompose\ Q\ ([:of-nat\ a:] + [:of-nat\ p:] * pX)) =$ 
      of-nat p ^ 3 * Lfun (pcompose ((pderiv ^ 3) Q) ([:of-nat a:] + [:of-nat
p:] * pX)) /
      24
    proof -
      fix a :: nat
      have d: (pderiv ^ 3) (pcompose Q ([:of-nat a:] + [:of-nat p:] * pX)) =
        Polynomial.smult (of-nat p ^ 3) (pcompose ((pderiv ^ 3) Q) ([:of-nat a:]
+ [:of-nat p:] *
pX))
      by (simp only: numeral-3-eq-3 funpow.simps o-apply id-apply pderiv-pcompose-affine
pderiv-smult smult-smult)
      (simp add: mult-ac)
      show tau (pcompose Q ([:of-nat a:] + [:of-nat p:] * pX)) =
        of-nat p ^ 3 * Lfun (pcompose ((pderiv ^ 3) Q) ([:of-nat a:] + [:of-nat
p:] * pX)) /
        24
    qed
  qed

```



```

    unfolding tau-def d Lfun-smult ..
  qed
  have (∑ a < p. tau (pcompose Q ([of-nat a:] + [of-nat p:] * pX))) =
    of-nat p ^ 3 * (∑ a < p. Lfun (pcompose ((pderiv ~ 3) Q) ([of-nat a:] +
      [of-nat p:] * pX))) / 24
    by (simp only: e sum-divide-distrib[symmetric] sum-distrib-left[symmetric])
  also have ... = of-nat p ^ 3 * (of-nat p * Lfun ((pderiv ~ 3) Q)) / 24
    by (simp only: Lfun-raabe)
  also have ... = of-nat p ^ 4 * tau Q
    by (simp add: tau-def power4-eq-xxxx power3-eq-cube)
  finally show ?thesis .
  qed

```

20.2 Harmonic increments and the pole sums

lemma *H5-dIdx-sub*: $H5 \ (dIdx \ n) - H5 \ (dIdx \ (n - 1)) = (if \ n = 0 \ then \ 0 \ else \ 1 / of-int \ n ^ 5)$

proof –

consider $n < 0 \mid n = 0 \mid n > 0$ **by** *linarith*

thus *?thesis*

proof *cases*

case 1

define j **where** $j = nat \ (-n)$

have $j0: j \neq 0$ **using** 1 **by** (*simp add: j-def*)

have $jn: of-nat \ j = (- \ of-int \ n :: rat)$ **using** 1 **by** (*simp add: j-def*)

have $d: dIdx \ n = j - 1 \ dIdx \ (n - 1) = j$ **using** 1

by (*auto simp: dIdx-def j-def nat-diff-distrib*)

have $h: H5 \ j = H5 \ (j - 1) + 1 / of-nat \ j ^ 5$ **by** (*rule H5-Suc-pred[OF j0]*)

have $i: 1 / of-nat \ j ^ 5 = - (1 / of-int \ n ^ 5 :: rat)$ **unfolding** jn

by (*simp add: power-minus-odd*)

show *?thesis* **using** 1 **unfolding** d **by** (*simp add: h i*)

next

case 2

thus *?thesis* **by** (*simp add: dIdx-def*)

next

case 3

define j **where** $j = nat \ n$

have $j0: j \neq 0$ **using** 3 **by** (*simp add: j-def*)

have $jn: of-nat \ j = (of-int \ n :: rat)$ **using** 3 **by** (*simp add: j-def*)

have $d: dIdx \ n = j \ dIdx \ (n - 1) = j - 1$ **using** 3

by (*auto simp: dIdx-def j-def nat-diff-distrib*)

have $h: H5 \ j = H5 \ (j - 1) + 1 / of-nat \ j ^ 5$ **by** (*rule H5-Suc-pred[OF j0]*)

show *?thesis* **using** 3 **unfolding** d **by** (*simp add: h jn*)

qed

qed

The scalar part of the truncated τ - $Y^{\sim\{ext\}}(1/(z - n/p))$.

definition *phi-tr* :: $nat \Rightarrow int \Rightarrow nat \Rightarrow rat$ **where**

phi-tr $p \ n \ M = (if \ int \ p \ dvd \ n \ then \ H5 \ (dIdx \ (n \ div \ int \ p)) - Cp-tr \ p \ M$

else tauAnPole-tr p (of-int n / of-nat p) M)

lemma *vp-int-div-p*: prime p $\implies \neg \text{int } p \text{ dvd } n \implies$
 $\text{vp } p \text{ (of-int } n \text{ / of-nat } p) = -1 \wedge (\text{of-int } n \text{ / of-nat } p :: \text{rat}) \neq 0$
proof –
 assume p: prime p and nd: $\neg \text{int } p \text{ dvd } n$
 have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer)
 have n0: $n \neq 0$ using nd by auto
 have p0: $\text{int } p \neq 0$ using prime-gt-0-nat[OF p] by simp
 have vp p (of-int n / of-nat p) = qmultiplicity (int p) (of-int n / of-int (int p))
 by (simp add: vp-def)
 also have ... = int (multiplicity (int p) n) – int (multiplicity (int p) (int p))
 by (rule qmultiplicity-divide-of-int[OF n0 p0 pe])
 also have ... = – 1 using nd pe by (simp add: not-dvd-imp-multiplicity-0
multiplicity-prime)
 finally show ?thesis using n0 p0 by simp
qed

lemma *phi-tr-sub*:
 assumes p: prime p
 shows $\exists M0. \forall M \geq M0. \text{vge } p \text{ (phi-tr } p \text{ n } M - \text{phi-tr } p \text{ (n – int } p) \text{ } M$
 $- \text{of-nat } p \wedge^5 * (H5 \text{ (dIdx } n) - H5 \text{ (dIdx (n – 1))})) B$
proof (cases int p dvd n)
 case True
 have phi-tr p n M – phi-tr p (n – int p) M – of-nat p $\wedge^5 * (H5 \text{ (dIdx } n) -$
 $H5 \text{ (dIdx (n – 1))}) =$
 0 for M
proof –
 obtain m where m: $n = \text{int } p * m$ using True by blast
 have p0: $\text{int } p \neq 0$ using prime-gt-0-nat[OF p] by simp
 have pq: $(\text{of-nat } p :: \text{rat}) \neq 0$ using prime-gt-0-nat[OF p] by simp
 have d1: $n \text{ div int } p = m$ using p0 by (simp add: m)
 have nm: $n - \text{int } p = \text{int } p * (m - 1)$ by (simp add: m algebra-simps)
 have d2: $(n - \text{int } p) \text{ div int } p = m - 1$ using p0 by (simp add: nm)
 have int p dvd (n – int p) using True by simp
 with True have phi-tr p n M – phi-tr p (n – int p) M = $H5 \text{ (dIdx } m) - H5$
 (dIdx (m – 1))
 by (simp add: phi-tr-def d1 d2)
 also have ... = (if m = 0 then 0 else 1 / of-int m $\wedge^5$) by (rule H5-dIdx-sub)
 also have ... = of-nat p $\wedge^5 * (H5 \text{ (dIdx } n) - H5 \text{ (dIdx (n – 1))})$
 unfolding H5-dIdx-sub using pq by (cases m = 0) (simp-all add: m
power-mult-distrib)
 finally show ?thesis by simp
qed
 thus ?thesis by (auto simp: vge-zero)
next
 case False
 have nd': $\neg \text{int } p \text{ dvd (n – int } p)$
 by (metis False diff-add-cancel dvd-add dvd-refl)

```

define  $s$  where  $s = \text{of-int } n / (\text{of-nat } p :: \text{rat})$ 
have  $sv0$ :  $vp \ p \ s = - \ 1$  and  $s0$ :  $s \neq 0$  using  $vp\text{-int-div-}p[OF \ p \ False]$  unfolding
 $s\text{-def}$ 
  by  $blast+$ 
have  $sv$ :  $vp \ p \ s < 0$  using  $sv0$  by  $simp$ 
have  $s1$ :  $\text{of-int } (n - \text{int } p) / \text{of-nat } p = s - 1$ 
  using  $\text{prime-gt-0-nat}[OF \ p]$  unfolding  $s\text{-def}$  by  $(simp \ add: \text{diff-divide-distrib})$ 
have  $e$ :  $\bigwedge M. \ \text{phi-tr } p \ n \ M - \text{phi-tr } p \ (n - \text{int } p) \ M - \text{of-nat } p \ ^5 * (H5 \ (dIdx \ n) -$ 
 $H5 \ (dIdx \ (n - 1))) =$ 
 $\tau\text{AnPole-tr } p \ s \ M - \tau\text{AnPole-tr } p \ (s - 1) \ M - s \ \text{powi } (- \ 5)$ 
proof  $-$ 
  fix  $M$ 
  have  $n0$ :  $n \neq 0$  using  $False$  by  $auto$ 
  have  $a$ :  $\text{phi-tr } p \ n \ M = \tau\text{AnPole-tr } p \ s \ M$  using  $False$  by  $(simp \ add: \text{phi-tr-def } s\text{-def})$ 
  have  $b$ :  $\text{phi-tr } p \ (n - \text{int } p) \ M = \tau\text{AnPole-tr } p \ (s - 1) \ M$ 
    by  $(simp \ only: \text{phi-tr-def } nd' \ \text{if-False } s1)$ 
  have  $c$ :  $\text{of-nat } p \ ^5 * (H5 \ (dIdx \ n) - H5 \ (dIdx \ (n - 1))) = s \ \text{powi } (- \ 5)$ 
    using  $n0$  by  $(simp \ add: H5\text{-dIdx-sub } s\text{-def } \text{power-int-minus } \text{power-divide})$ 
  show  $\text{phi-tr } p \ n \ M - \text{phi-tr } p \ (n - \text{int } p) \ M - \text{of-nat } p \ ^5 * (H5 \ (dIdx \ n) -$ 
 $H5 \ (dIdx \ (n - 1))) =$ 
 $\tau\text{AnPole-tr } p \ s \ M - \tau\text{AnPole-tr } p \ (s - 1) \ M - s \ \text{powi } (- \ 5)$ 
    by  $(simp \ only: a \ b \ c)$ 
qed
show  $?thesis$  unfolding  $e$  by  $(rule \ \tau\text{AnPole-tr-sub}[OF \ p \ sv \ s0])$ 
qed

```

definition $Gsum\text{-tr} :: \text{nat} \Rightarrow \text{int} \Rightarrow \text{nat} \Rightarrow \text{rat}$ **where**

$$Gsum\text{-tr } p \ r \ M = (\sum a < p. \ \text{phi-tr } p \ (r - \text{int } a) \ M)$$

lemma $Gsum\text{-tr-step}$: $0 < p \implies$

$$Gsum\text{-tr } p \ n \ M - Gsum\text{-tr } p \ (n - 1) \ M = \text{phi-tr } p \ n \ M - \text{phi-tr } p \ (n - \text{int } p) \ M$$

proof $-$

assume $0 < p$

then obtain q **where** $q: p = \text{Suc } q$ **by** $(cases \ p) \ auto$

have A : $Gsum\text{-tr } p \ n \ M = \text{phi-tr } p \ n \ M + (\sum a < q. \ \text{phi-tr } p \ (n - \text{int } (\text{Suc } a)) \ M)$

unfolding $Gsum\text{-tr-def } q \ \text{sum.lessThan-Suc-shift}$ **by** $simp$

have B : $Gsum\text{-tr } p \ (n - 1) \ M = (\sum a <$

$$q. \ \text{phi-tr } p \ (n - \text{int } (\text{Suc } a)) \ M) + \text{phi-tr } p \ (n - \text{int } p) \ M$$

unfolding $Gsum\text{-tr-def } q \ \text{sum.lessThan-Suc}$ **by** $(simp \ add: \text{algebra-simps})$

show $?thesis$ **unfolding** $A \ B$ **by** $simp$

qed

lemma $Gsum\text{-tr-zero}$: $\text{prime } p \implies Gsum\text{-tr } p \ 0 \ M = 0$

proof $-$

assume p : $\text{prime } p$

```

have 1 < p using p by (rule prime-gt-1-nat)
then obtain q where q: p = Suc q by (cases p) auto
have nd: ¬ int p dvd - int a if 0 < a a < p for a
  using that by (auto dest: dvd-imp-le)
have C: Cp-tr p M = (∑ a < q. tauAnPole-tr p (- of-nat (Suc a) / of-nat p)
M)
  unfolding Cp-tr-def q diff-Suc-1 image-Suc-lessThan[symmetric] sum.reindex[OF
inj-Suc] o-def ..
  have Gsum-tr p 0 M = phi-tr p 0 M + (∑ a < q. phi-tr p (- int (Suc a)) M)
  unfolding Gsum-tr-def q sum.lessThan-Suc-shift by simp
  also have phi-tr p 0 M = - Cp-tr p M by (simp add: phi-tr-def dIdx-def H5-def)
  also have (∑ a < q. phi-tr p (- int (Suc a)) M) = (∑ a <
q. tauAnPole-tr p (- of-nat (Suc a) / of-nat p) M)
  proof (rule sum.cong[OF refl])
    fix a assume a ∈ {..

```

```

      (Gsum-tr p (n - 1) M - of-nat p ^ 5 * H5 (dIdx
(n - 1)))) B
    unfolding eq using M0 by blast
  qed
  have base:  $\bigwedge M. \text{Gsum-tr } p \ 0 \ M - \text{of-nat } p \ ^5 * H5 \ (dIdx \ 0) = 0$ 
    by (simp add: Gsum-tr-zero[OF p] dIdx-def H5-def)
  define D where  $D \ r \ M = \text{Gsum-tr } p \ r \ M - \text{of-nat } p \ ^5 * H5 \ (dIdx \ r)$  for  $r \ M$ 
  have stepD:  $\bigwedge n. \exists M0. \forall M \geq M0. \text{vge } p \ (D \ n \ M - D \ (n - 1) \ M) \ B$ 
    unfolding D-def by (rule step)
  have baseD:  $\bigwedge M. D \ 0 \ M = 0$  unfolding D-def by (rule base)
  have  $\exists M0. \forall M \geq M0. \text{vge } p \ (D \ r \ M) \ B$ 
  proof (rule int-induct[where k = 0 and P =  $\lambda r. \exists M0. \forall M \geq$ 
     $M0. \text{vge } p \ (D \ r \ M) \ B$ ])
    show  $\exists M0. \forall M \geq M0. \text{vge } p \ (D \ 0 \ M) \ B$  by (simp add: baseD vge-zero)
  next
    fix i :: int assume  $0 \leq i$  and  $\exists M0. \forall M \geq M0. \text{vge } p \ (D \ i \ M) \ B$ 
    then obtain M1 where  $M1: \forall M \geq M1. \text{vge } p \ (D \ i \ M) \ B$  by blast
    obtain M2 where  $M2: \forall M \geq M2. \text{vge } p \ (D \ (i + 1) \ M - D \ i \ M) \ B$ 
      using stepD[of i + 1] by auto
    show  $\exists M0. \forall M \geq M0. \text{vge } p \ (D \ (i + 1) \ M) \ B$ 
    proof (intro exI[of - max M1 M2] allI impI)
      fix M assume  $\max M1 \ M2 \leq M$ 
      hence a:  $\text{vge } p \ (D \ i \ M) \ B$  and b:  $\text{vge } p \ (D \ (i + 1) \ M - D \ i \ M) \ B$  using
M1 M2 by auto
      show  $\text{vge } p \ (D \ (i + 1) \ M) \ B$  using vge-add[OF p b a] by simp
    qed
  next
    fix i :: int assume  $i \leq 0$  and  $\exists M0. \forall M \geq M0. \text{vge } p \ (D \ i \ M) \ B$ 
    then obtain M1 where  $M1: \forall M \geq M1. \text{vge } p \ (D \ i \ M) \ B$  by blast
    obtain M2 where  $M2: \forall M \geq M2. \text{vge } p \ (D \ i \ M - D \ (i - 1) \ M) \ B$  using
stepD[of i]
      by auto
    show  $\exists M0. \forall M \geq M0. \text{vge } p \ (D \ (i - 1) \ M) \ B$ 
    proof (intro exI[of - max M1 M2] allI impI)
      fix M assume  $\max M1 \ M2 \leq M$ 
      hence a:  $\text{vge } p \ (D \ i \ M) \ B$  and b:  $\text{vge } p \ (D \ i \ M - D \ (i - 1) \ M) \ B$  using
M1 M2 by auto
      show  $\text{vge } p \ (D \ (i - 1) \ M) \ B$  using vge-add[OF p a vge-neg[OF b]] by simp
    qed
  qed
  thus ?thesis by (simp only: D-def)
qed

```

20.3 The pole values of $\tau^{\sim\{ext\}}\text{-}Y$ on the distributed poles

lemma quotient-of-int-div-prime:

assumes prime $p \nmid \text{int } p \text{ dvd } n$

shows $\text{snd } (\text{quotient-of } (\text{of-int } n / \text{of-nat } p)) \neq 1$

using far-denom-ne-1[OF assms(1) conjunct1[OF vp-int-div-p[OF assms]]] .

```

lemma tauPoleP-tr-int-div:
  assumes prime p
  shows tauPoleP-tr p (Yp-tr p M) (of-int n / of-nat p) M =
    [:phi-tr p n M:] - (if int p dvd n then [:of-nat p ^ 5:] * pX else 0)
proof (cases int p dvd n)
  case True
    then obtain m where m: n = int p * m by blast
    have p0: int p ≠ 0 using prime-gt-0-nat[OF assms] by simp
    have pq: (of-nat p :: rat) ≠ 0 using prime-gt-0-nat[OF assms] by simp
    have e: of-int n / of-nat p = (of-int m :: rat) using pq by (simp add: m)
    have d: n div int p = m using p0 by (simp add: m)
    show ?thesis unfolding e tauPoleP-tr-near using True
      by (simp add: phi-tr-def d Yp-tr-def algebra-simps)
  next
    case False
    show ?thesis using False tauPoleP-tr-far[OF assms conjunct1[OF vp-int-div-p[OF
      assms False]]]
      by (simp add: phi-tr-def)
  qed

lemma sum-tauPoleP-tr:
  assumes p: prime p
  shows ∃ M0. ∀ M ≥ M0.
    vpGge p ((∑ a < p. tauPoleP-tr p (Yp-tr p M) ((of-int r - of-nat a) / of-nat
    p) M)
      - [:of-nat p ^ 5:] * ([:H5 (dIdx r):] - pX)) B
proof -
  have e: ∧ M. (∑ a < p. tauPoleP-tr p (Yp-tr p M) ((of-int r - of-nat a) /
    of-nat p) M) =
    [:Gsum-tr p r M:] - [:of-nat p ^ 5:] * pX
  proof -
    fix M
    have p0: 0 < p using p by (rule prime-gt-0-nat)
    have c: (of-int r - of-nat a) / of-nat p = (of-int (r - int a) / of-nat p :: rat)
  for a
    by simp
    have iff: int p dvd (r - int a) ⟷ a = nat (r mod int p) if a < p for a
  proof -
    have int a mod int p = int a using that by simp
    hence int p dvd (r - int a) ⟷ r mod int p = int a
      using mod-eq-dvd-iff[of r int p int a] by simp
    also have ... ⟷ a = nat (r mod int p)
      using pos-mod-sign[of int p r] p0 by auto
    finally show ?thesis .
  qed
  have lt: nat (r mod int p) < p using pos-mod-bound[of int p r] pos-mod-sign[of
    int p r] p0
    by (simp add: nat-less-iff)

```

have $(\sum a < p. \text{tauPoleP-tr } p (Yp\text{-tr } p M) ((\text{of-int } r - \text{of-nat } a) / \text{of-nat } p) M) =$
 $(\sum a < p. [\text{phi-tr } p (r - \text{int } a) M:] - (\text{if int } p \text{ dvd } (r - \text{int } a) \text{ then } [\text{of-nat } p \wedge 5:] * pX \text{ else } 0))$
unfolding $c \text{ tauPoleP-tr-int-div}[OF \ p] \dots$
also have $\dots = (\sum a < p. [\text{phi-tr } p (r - \text{int } a) M:]) - (\sum a < p. (\text{if int } p \text{ dvd } (r - \text{int } a) \text{ then } [\text{of-nat } p \wedge 5:] * pX \text{ else } 0))$
by $(\text{rule sum-subtractf})$
also have $(\sum a < p. [\text{phi-tr } p (r - \text{int } a) M:]) = [Gsum\text{-tr } p \ r \ M:]$
unfolding $Gsum\text{-tr-def sum-const-poly} \dots$
also have $(\sum a < p. (\text{if int } p \text{ dvd } (r - \text{int } a) \text{ then } [\text{of-nat } p \wedge 5:] * pX \text{ else } 0)) =$
 $(\sum a < p. (\text{if } a = \text{nat } (r \bmod \text{int } p) \text{ then } [\text{of-nat } p \wedge 5:] * pX \text{ else } 0))$
by $(\text{rule sum.cong}[OF \ refl]) (\text{simp add: iff})$
also have $\dots = [\text{of-nat } p \wedge 5:] * pX$ **using** lt
by $(\text{simp only: sum.delta finite-lessThan lessThan-iff if-True})$
finally show $(\sum a < p. \text{tauPoleP-tr } p (Yp\text{-tr } p M) ((\text{of-int } r - \text{of-nat } a) / \text{of-nat } p) M) =$
 $[Gsum\text{-tr } p \ r \ M:] - [\text{of-nat } p \wedge 5:] * pX .$
qed
from $Gsum\text{-tr-eq}[OF \ p, \text{ of } r \ B]$ **obtain** $M0$ **where** $M0: \forall M \geq$
 $M0. \text{vge } p (Gsum\text{-tr } p \ r \ M - \text{of-nat } p \wedge 5 * H5 \ (dIdx \ r)) \ B$ **by** $blast$
show $?thesis$
proof $(\text{intro exI}[\text{of } - M0] \text{ allI impI})$
fix M **assume** $M0 \leq M$
hence $v: \text{vge } p (Gsum\text{-tr } p \ r \ M - \text{of-nat } p \wedge 5 * H5 \ (dIdx \ r)) \ B$ **using** $M0$
by $blast$
have $\text{eq}: [Gsum\text{-tr } p \ r \ M:] - [\text{of-nat } p \wedge 5:] * pX - [\text{of-nat } p \wedge 5:] * ([H5 \ (dIdx \ r):] -$
 $pX) =$
 $[Gsum\text{-tr } p \ r \ M - \text{of-nat } p \wedge 5 * H5 \ (dIdx \ r):]$
by $(\text{simp add: algebra-simps})$
show $\text{vpGge } p ((\sum a < p. \text{tauPoleP-tr } p (Yp\text{-tr } p M) ((\text{of-int } r - \text{of-nat } a) / \text{of-nat } p) M)$
 $- [\text{of-nat } p \wedge 5:] * ([H5 \ (dIdx \ r):] - pX)) \ B$
unfolding $e \text{ eq}$ **by** $(\text{rule vpGge-C}[OF \ v])$
qed
qed

20.4 Partial fractions and the substitution $x = a + p \ z$

$\tau \wedge \{ext\}$ of a rational function given by an explicit partial fraction decomposition.

lemma tauExt-tr-decomp :

assumes $\text{finite } S \text{ inj-on } f \ S$
and $U = Q' * qPoleProd \ (f \ ' \ S) + (\sum r \in S. [\varrho \ r:] * qPoleProd \ (f \ ' \ (S - \{r\})))$
shows $\text{tauExt-tr } p \ Y \ U \ (f \ ' \ S) \ M = [\text{tau } Q':] + (\sum r \in S. [\varrho \ r:] * \text{tauPoleP-tr } p \ Y \ (f \ r) \ M)$

```

proof –
  note  $S = \text{assms}(1)$  and  $\text{inj} = \text{assms}(2)$  and  $U = \text{assms}(3)$ 
  define  $Pi$  where  $Pi = \text{qPoleProd } (f \text{ ' } S)$ 
  define  $R$  where  $R = (\sum r \in S. [\text{!} r] * \text{qPoleProd } (f \text{ ' } (S - \{r\})))$ 
  have  $fS$ :  $\text{finite } (f \text{ ' } S)$  using  $S$  by  $\text{simp}$ 
  have  $Pi0$ :  $Pi \neq 0$  using  $\text{qPoleProd-monic}[OF fS]$  by  $(\text{auto simp: } Pi\text{-def})$ 
  have  $dPi$ :  $\text{degree } Pi = \text{card } S$  unfolding  $Pi\text{-def}$   $\text{degree-qPoleProd}[OF fS]$  using
 $\text{card-image}[OF \text{inj}]$  .
  have  $\text{cardr}$ :  $\text{card } (f \text{ ' } (S - \{r\})) = \text{card } S - 1$  if  $r \in S$  for  $r$ 
    using  $\text{card-image}[OF \text{inj-on-subset}[OF \text{inj}, \text{of } S - \{r\}]]$   $S$  that by  $\text{auto}$ 
  have  $R\text{div}Pi$ :  $R \text{ div } Pi = 0$ 
  proof  $(\text{cases } S = \{\})$ 
    case  $\text{True}$  thus  $?thesis$  by  $(\text{simp add: } R\text{-def})$ 
  next
    case  $\text{False}$ 
    have  $cS$ :  $0 < \text{card } S$  using  $S$   $\text{False}$  by  $(\text{simp add: } \text{card-gt-0-iff})$ 
    have  $\text{degree } R \leq \text{card } S - 1$ 
      unfolding  $R\text{-def}$ 
    proof  $(\text{rule degree-sum-le}[OF S])$ 
      fix  $r$  assume  $r$ :  $r \in S$ 
      have  $\text{degree } ([\text{!} r] * \text{qPoleProd } (f \text{ ' } (S - \{r\}))) \leq$ 
         $\text{degree } (\text{qPoleProd } (f \text{ ' } (S - \{r\})))$ 
      by  $(\text{simp add: degree-smult-le})$ 
      also have  $\dots = \text{card } S - 1$  using  $S$   $r$  by  $(\text{simp add: degree-qPoleProd cardr})$ 
      finally show  $\text{degree } ([\text{!} r] * \text{qPoleProd } (f \text{ ' } (S - \{r\}))) \leq \text{card } S - 1$  .
    qed
    hence  $\text{degree } R < \text{degree } Pi$  using  $cS$   $dPi$  by  $\text{simp}$ 
    thus  $?thesis$  by  $(\text{rule div-poly-less})$ 
  qed
  have  $\text{div}$ :  $U \text{ div } \text{qPoleProd } (f \text{ ' } S) = Q'$ 
    unfolding  $U$   $Pi\text{-def}[\text{symmetric}]$   $R\text{-def}[\text{symmetric}]$  using  $Pi0$   $R\text{div}Pi$  by  $\text{simp}$ 
  have  $\text{res}$ :  $\text{qRes } (f \text{ ' } S) \ U \ (f \ r) = \text{!} r$  if  $r$ :  $r \in S$  for  $r$ 
  proof –
    have  $\text{img}$ :  $f \text{ ' } S - \{f \ r\} = f \text{ ' } (S - \{r\})$  using  $\text{inj } r$  by  $(\text{auto simp: inj-on-def})$ 
    have  $f \ r \notin f \text{ ' } (S - \{r\})$  using  $\text{inj } r$  by  $(\text{auto simp: inj-on-def})$ 
    with  $S$  have  $\text{den}$ :  $(\prod t \in f \text{ ' } (S - \{r\}). f \ r - t) \neq 0$ 
      by  $(\text{auto simp: prod-zero-iff})$ 
    have  $z$ :  $\text{poly } (\text{qPoleProd } (f \text{ ' } S)) \ (f \ r) = 0$  using  $S$   $r$ 
      by  $(\text{auto simp: poly-qPoleProd prod-zero-iff})$ 
    have  $z2$ :  $\text{poly } (\text{qPoleProd } (f \text{ ' } (S - \{r'\}))) \ (f \ r) = 0$  if  $r' \in S$   $r' \neq r$  for  $r'$ 
      using  $S$   $r$  that by  $(\text{auto simp: poly-qPoleProd prod-zero-iff})$ 
    have  $\text{poly } U \ (f \ r) = (\sum r' \in S. \text{!} r' * \text{poly } (\text{qPoleProd } (f \text{ ' } (S - \{r'\}))) \ (f \ r))$ 
      unfolding  $U$  using  $z$  by  $(\text{simp add: poly-sum})$ 
    also have  $\dots = \text{!} r * \text{poly } (\text{qPoleProd } (f \text{ ' } (S - \{r\}))) \ (f \ r) + (\sum r' \in$ 
       $S - \{r\}. \text{!} r' * \text{poly } (\text{qPoleProd } (f \text{ ' } (S - \{r'\}))) \ (f \ r))$ 
    by  $(\text{rule sum.remove}[OF S r])$ 
    also have  $(\sum r' \in S - \{r\}. \text{!} r' * \text{poly } (\text{qPoleProd } (f \text{ ' } (S - \{r'\}))) \ (f \ r)) =$ 
0
    using  $z2$  by  $(\text{intro sum.neutral})$   $\text{auto}$ 

```


also have $\text{poly } (qPoleProd (f' (S - \{r\}))) (f r) = (\prod t \in f' (S - \{r\}). f r - t)$
by (*rule poly-qPoleProd*)
finally show *?thesis* **using** *den unfolding qRes-def img by simp*
qed
have $\text{tauExt-tr } p \ Y \ U \ (f' S) \ M = [: \text{tauK } Q':] + (\sum s \in f' S. [:qRes (f' S) \ U \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M)$
unfolding *tauExt-tr-def div ..*
also have $(\sum s \in f' S. [:qRes (f' S) \ U \ s:] * \text{tauPoleP-tr } p \ Y \ s \ M) = (\sum r \in S. [:qRes (f' S) \ U \ (f r):] * \text{tauPoleP-tr } p \ Y \ (f r) \ M)$
by (*simp only: sum.reindex[OF inj] o-def*)
also have $\dots = (\sum r \in S. [:q \ r:] * \text{tauPoleP-tr } p \ Y \ (f r) \ M)$
by (*rule sum.cong[OF refl] (simp add: res)*)
finally show *?thesis* **by** (*simp only: tauK-eq-tau*)
qed

lemma *distrib-inj: 0 < p $\implies$*
inj ($\lambda r. (\text{of-int } r - \text{of-nat } a) / (\text{of-nat } p :: \text{rat})$)
by (*auto simp: inj-def*)

$$(\prod -\{r \in T\}(x - r))(a + p \ z) = p^{\sim\{|T|\}} \prod -\{r \in T\}(z - (r - a)/p).$$

lemma *linPoleProd-distrib:*
assumes $0 < p$ *finite T*
shows $\text{pcompose } (\text{linPoleProd } T) ([: \text{of-nat } a:] + [: \text{of-nat } p:] * pX) = \text{Polynomial.smult } (\text{of-nat } p \wedge \text{card } T) (qPoleProd (\text{distribPoles } p \ T \ a))$
proof –
have $p0: (\text{of-nat } p :: \text{rat}) \neq 0$ **using** *assms(1)* **by** *simp*
have *inj: inj-on* ($\lambda r. (\text{of-int } r - \text{of-nat } a) / (\text{of-nat } p :: \text{rat})$) *T*
using *distrib-inj[OF assms(1), of a]* **by** (*simp add: inj-on-def inj-def*)
have $X: pX = [:0, 1:]$ **by** (*simp add: monom-Suc*)
have *fac:* $\text{pcompose } (pX - [: \text{of-int } r:]) ([: \text{of-nat } a:] + [: \text{of-nat } p:] * pX) = [: \text{of-nat } p:] * (pX - [: \text{of-int } r - \text{of-nat } a] / \text{of-nat } p:)$ **for** r
unfolding X **using** $p0$ **by** (*simp add: pcompose-pCons algebra-simps*)
have $\text{pcompose } (\text{linPoleProd } T) ([: \text{of-nat } a:] + [: \text{of-nat } p:] * pX) = (\prod r \in T. [: \text{of-nat } p:] * (pX - [: \text{of-int } r - \text{of-nat } a] / \text{of-nat } p:))$
unfolding *linPoleProd-def pcompose-prod fac ..*
also have $\dots = [: \text{of-nat } p:] \wedge \text{card } T * (\prod r \in T. pX - [: \text{of-int } r - \text{of-nat } a] / \text{of-nat } p:)$
by (*simp only: prod.distrib prod-constant*)
also have $(\prod r \in T. pX - [: \text{of-int } r - \text{of-nat } a] / \text{of-nat } p:) = qPoleProd (\text{distribPoles } p \ T \ a)$
unfolding *qPoleProd-def distribPoles-def prod.reindex[OF inj] o-def ..*
finally show *?thesis* **by** (*simp add: const-poly-power*)
qed

lemma *summand-eq:*
assumes p : *prime p* **and** S : *finite S*
shows $\text{tauExt-tr } p \ Y \ (\text{distribNum } p \ S \ P \ a) \ (\text{distribPoles } p \ S \ a) \ M = [: \text{tau } (\text{pcompose } (P \ \text{div } \text{linPoleProd } S) ([: \text{of-nat } a:] + [: \text{of-nat } p:] * pX)):]$

$+$ $(\sum r \in S. [:linResidue\ S\ P\ r\ /\ of-nat\ p:] * tauPoleP-tr\ p\ Y\ ((of-int\ r - of-nat\ a) /\ of-nat\ p)\ M)$

proof –

define Q **where** $Q = P\ div\ linPoleProd\ S$
define f **where** $f = (\lambda r. (of-int\ r - of-nat\ a) /\ (of-nat\ p :: rat))$
have $p0: 0 < p$ **using** p **by** $(rule\ prime-gt-0-nat)$
have $inj: inj-on\ f\ S$ **using** $distrib-inj[OF\ p0, of\ a]$ **by** $(simp\ add: f-def\ inj-on-def)$
have $P: P = Q * linPoleProd\ S + (\sum r \in S. [:linResidue\ S\ P\ r:] * linPoleProd\ (S - \{r\}))$
unfolding $Q-def$ **by** $(rule\ div-mult-mod-eq[of\ P\ linPoleProd\ S, unfolded\ mod-linPoleProd[OF\ S], symmetric])$
have $U: distribNum\ p\ S\ P\ a = pcompose\ Q\ ([:of-nat\ a:] + [:of-nat\ p:] * pX) * qPoleProd\ (f\ 'S)$
 $+$ $(\sum r \in S. [:linResidue\ S\ P\ r\ /\ of-nat\ p:] * qPoleProd\ (f\ ' (S - \{r\})))$

proof –

define R **where** $R = [:of-nat\ a:] + [:of-nat\ p:] * pX$
define c **where** $c = (of-nat\ p :: rat)\ powi\ (-\ int\ (card\ S))$
have $pq: (of-nat\ p :: rat) \neq 0$ **using** $p0$ **by** $simp$
have $dS: distribPoles\ p\ T\ a = f\ 'T$ **for** T **by** $(simp\ add: distribPoles-def\ f-def)$
have $c1: c * of-nat\ p \wedge^{card\ S} = 1$ **using** pq **by** $(simp\ add: c-def\ power-int-minus)$
have $c2: c * of-nat\ p \wedge^{card\ (S - \{r\})} = 1 /\ of-nat\ p$ **if** $r: r \in S$ **for** r

proof –

have $cS: card\ S = Suc\ (card\ (S - \{r\}))$ **by** $(rule\ card.remove[OF\ S\ r])$
have $c * of-nat\ p \wedge^{card\ (S - \{r\})} * of-nat\ p = 1$ **using** $c1$ **unfolding** cS **by** $(simp\ add: mult-ac)$
thus $?thesis$ **using** pq **by** $(simp\ add: field-simps)$

qed

have $e1: pcompose\ P\ R = pcompose\ Q\ R * pcompose\ (linPoleProd\ S)\ R + (\sum r \in S. [:linResidue\ S\ P\ r:] * pcompose\ (linPoleProd\ (S - \{r\})))$

$R)$

proof –

have $pcompose\ P\ R = pcompose\ (Q * linPoleProd\ S + (\sum r \in S. [:linResidue\ S\ P\ r:] * linPoleProd\ (S - \{r\})))\ R$
by $(rule\ arg-cong[where\ f = \lambda x. pcompose\ x\ R, OF\ P])$
thus $?thesis$ **by** $(simp\ only: pcompose-add\ pcompose-mult\ pcompose-sum\ pcompose-const)$

qed

have $L1: pcompose\ (linPoleProd\ S)\ R = Polynomial.smult\ (of-nat\ p \wedge^{card\ S})\ (qPoleProd\ (f\ 'S))$
unfolding $R-def\ dS[symmetric]$ **by** $(rule\ linPoleProd-distrib[OF\ p0\ S])$
have $L2: pcompose\ (linPoleProd\ (S - \{r\}))\ R = Polynomial.smult\ (of-nat\ p \wedge^{card\ (S - \{r\})})\ (qPoleProd\ (f\ ' (S - \{r\})))$ **for**

r

unfolding $R-def\ dS[symmetric]$ **by** $(rule\ linPoleProd-distrib[OF\ p0\ finite-Diff[OF\ S]])$
have $h1: [:c:] * (A * Polynomial.smult\ k\ B) = Polynomial.smult\ (c * k)\ (A * B)$
for $A\ B :: rat\ poly$ **and** k

```

    by (simp add: mult-ac)
  have h2: [:c:] * ([:x:] * Polynomial.smult k B) = [:x * (c * k):] * B
    for B :: rat poly and x k
    by (simp add: mult-ac)
  have distribNum p S P a = [:c:] * pcompose P R unfolding distribNum-def
c-def R-def ..
  also have ... = [:c:] * (pcompose Q R *
    Polynomial.smult (of-nat p ^ card S) (qPoleProd (f ' S)))
    + (∑ r ∈ S. [:c:] * ([:linResidue S P r:] * Polynomial.smult (of-nat p ^
card (S -
    {r}))) (qPoleProd (f ' (S - {r}))))))
  unfolding e1 L1 L2 distrib-left sum-distrib-left ..
  also have ... = pcompose Q R * qPoleProd (f ' S) + (∑ r ∈
    S. [:linResidue S P r / of-nat p:] * qPoleProd (f ' (S - {r})))
  proof -
  have (∑ r ∈ S. [:c:] * ([:linResidue S P r:] *
    Polynomial.smult (of-nat p ^ card (S - {r}))) (qPoleProd (f ' (S - {r}))))
    = (∑ r ∈ S. [:linResidue S P r / of-nat p:] * qPoleProd (f ' (S - {r})))
  proof (rule sum.cong[OF refl])
  fix r assume r: r ∈ S
  show [:c:] * ([:linResidue S P r:] * Polynomial.smult (of-nat p ^ card (S -
    {r}))) (qPoleProd (f ' (S - {r}))) =
    [:linResidue S P r / of-nat p:] * qPoleProd (f ' (S - {r}))
  unfolding h2 c2[OF r] by simp
  qed
  thus ?thesis unfolding h1 c1 smult-1-left by simp
  qed
  finally show ?thesis unfolding R-def .
  qed
  have tauExt-tr p Y (distribNum p S P a) (f ' S) M =
    [:tau (pcompose Q ([:of-nat a:] + [:of-nat p:] * pX)):] + (∑ r ∈
    S. [:linResidue S P r / of-nat p:] * tauPoleP-tr p Y (f r) M)
  by (rule tauExt-tr-decomp[OF S inj U])
  thus ?thesis by (simp add: f-def Q-def distribPoles-def)
  qed

```

20.5 Lemma 3.2

lemma *finite-eventually-all*:

assumes *finite* S $\bigwedge r. r \in S \implies$

$\exists M0 :: nat. \forall M \geq M0. Q\ r\ M$

shows $\exists M0. \forall r \in S. \forall M \geq M0. Q\ r\ M$

using *assms*

proof (*induction* S *rule*: *finite-induct*)

case *empty* **thus** ?case **by** *simp*

next

case (*insert* x F)

obtain M1 **where** M1: $\forall r \in F. \forall M \geq M1. Q\ r\ M$ **using** *insert* **by** *blast*

obtain M2 **where** M2: $\forall M \geq M2. Q\ x\ M$ **using** *insert.prem*s **by** *blast*

show ?case **by** (rule exI[of - max M1 M2]) (use M1 M2 **in** auto)
qed

lemma C-mul-diff-C-mul: $[:c:] * (X - [:k:] * T) = [:c:] * X - [:c * k:] * (T :: \text{rat poly})$
by (simp add: smult-diff-right)

lemma distribution-formula-aux:

fixes c k t :: rat **and** A :: rat poly **and** x :: 'a $\Rightarrow$ rat **and** T :: 'a $\Rightarrow$ rat poly

assumes ck: $c * k = 1$

shows $[:t:] + (\sum r \in S. [:x r:] * T r) - [:c:] * ([:k * t:] + A) = -[:c:] * (A - (\sum r \in S. [:x r * k:] * T r))$

proof -

have $\exists y. y * k * c = y$ **for** y **using** ck

by (metis mult.commute mult.left-commute mult.right-neutral)

have 1: $[:c:] * [:k * t:] = [:t:]$ **using** ck

by (simp add: mult.assoc[symmetric] mult.commute[of k c])

have 2: $[:c:] * (\sum r \in S. [:x r * k:] * T r) = (\sum r \in S. [:x r:] * T r)$

unfolding sum-distrib-left **by** (rule sum.cong[OF refl]) (simp add: 3)

have $[:t:] + (\sum r \in S. [:x r:] * T r) - [:c:] * ([:k * t:] + A) = (\sum r \in S. [:x r:] * T r) - [:c:] * A$

unfolding distrib-left 1 **by** simp

also have $\dots = -[:c:] * (A - (\sum r \in S. [:x r * k:] * T r))$

by (simp only: mult-minus-left right-diff-distrib 2 minus-diff-eq)

finally show ?thesis .

qed

$\tau \cdot X(g) = p^{-4} \sum -\{a < p\} \tau \cdot Y^{\text{ext}}(g(a + p x))$ with $Y = p^{-5}X + C \cdot p$ (Lean: lemma-3-2'); the truncated right-hand side agrees with the left-hand side modulo any prescribed power of p once M is large.

theorem tauX-distribution-formula:

assumes p: prime p **and** p7: $7 \leq p$ **and** S: finite S

shows $\exists M0. \forall M \geq M0.$

$vpGge\ p\ (\tau X\ S\ P - [:of\text{-}nat\ p\ powi\ (-\ 4)]:) *$

$(\sum a < p. \tau Ext\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ a)\ (distribPoles\ p\ S\ a)\ M))$

B

proof -

define Q **where** $Q = P \text{ div } \text{linPoleProd } S$

define res **where** $res = \text{linResidue } S\ P$

have p0: $(of\text{-}nat\ p :: \text{rat}) \neq 0$ **using** p **by** (simp add: prime-gt-0-nat)

— the right-hand side, rearranged with summand-eq and Raabe

have rhs: $\bigwedge M. (\sum a < p. \tau Ext\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ a)\ (distribPoles\ p\ S\ a)\ M) =$

$[:of\text{-}nat\ p^{-4} * \tau Q:] + (\sum r \in S. [:res\ r / of\text{-}nat\ p:] *$

$(\sum a < p. \tau PoleP\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ ((of\text{-}int\ r - of\text{-}nat\ a) / of\text{-}nat\ p)\ M))$

unfolding *summand-eq*[*OF p S*] *sum.distrib sum-const-poly tau-raabe Q-def*[*symmetric*]
res-def[*symmetric*]
by (*subst sum.swap*) (*simp only: sum-distrib-left*)
— the pole parts converge to $[:res\ r * p^{\wedge} 4:] * \tau Pole\ r$
have *pole*: $\bigwedge r\ B'.\ r \in S \implies \exists M0.\ \forall M \geq M0.$
 $vpGge\ p\ ([:res\ r / of-nat\ p:] * (\sum a <$
 $p.\ \tau PoleP-tr\ p\ (Yp-tr\ p\ M)\ ((of-int\ r - of-nat\ a) / of-nat\ p)\ M)$
 $- [:res\ r * of-nat\ p^{\wedge} 4:] * \tau Pole\ r)\ B'$
proof —
fix *r B'* **assume** *r*: $r \in S$
define *c* **where** $c = res\ r / of-nat\ p$
obtain *M0* **where** $M0: \forall M \geq M0.\ vpGge\ p\ ((\sum a <$
 $p.\ \tau PoleP-tr\ p\ (Yp-tr\ p\ M)\ ((of-int\ r - of-nat\ a) / of-nat\ p)\ M)$
 $- [:of-nat\ p^{\wedge} 5:] * ([:H5\ (dIdx\ r):] - pX))\ (B' - of-int\ (vp\ p\ c))$
using *sum-tauPoleP-tr*[*OF p, of r B' - of-int (vp p c)*] **by** *blast*
have *k*: $c * of-nat\ p^{\wedge} 5 = res\ r * of-nat\ p^{\wedge} 4$ **unfolding** *c-def* **using** *p0*
by (*simp add: field-simps eval-nat-numeral*)
have *e*: $[:c:] * (X - [:of-nat\ p^{\wedge} 5:] * ([:H5\ (dIdx\ r):] - pX)) =$
 $[:res\ r / of-nat\ p:] * X - [:res\ r * of-nat\ p^{\wedge} 4:] * \tau Pole\ r$ **for** *X*
unfolding *C-mul-diff-C-mul k* **unfolding** *c-def tauPole-def ..*
show $\exists M0.\ \forall M \geq M0.$
 $vpGge\ p\ ([:res\ r / of-nat\ p:] * (\sum a <$
 $p.\ \tau PoleP-tr\ p\ (Yp-tr\ p\ M)\ ((of-int\ r - of-nat\ a) / of-nat\ p)\ M)$
 $- [:res\ r * of-nat\ p^{\wedge} 4:] * \tau Pole\ r)\ B'$
proof (*intro exI*[*of - M0*] *allI impI*)
fix *M* **assume** *M*: $M0 \leq M$
have *v*: $vpGge\ p\ ([:c:] * ((\sum a <$
 $p.\ \tau PoleP-tr\ p\ (Yp-tr\ p\ M)\ ((of-int\ r - of-nat\ a) / of-nat\ p)\ M)$
 $- [:of-nat\ p^{\wedge} 5:] * ([:H5\ (dIdx\ r):] - pX)))\ (of-int\ (vp\ p\ c) + (B' -$
 $of-int\ (vp\ p\ c)))$
using *vpGge-mul*[*OF p vpGge-C-vp M0* [*rule-format, OF M*]] .
show $vpGge\ p\ ([:res\ r / of-nat\ p:] * (\sum a <$
 $p.\ \tau PoleP-tr\ p\ (Yp-tr\ p\ M)\ ((of-int\ r - of-nat\ a) / of-nat\ p)\ M)$
 $- [:res\ r * of-nat\ p^{\wedge} 4:] * \tau Pole\ r)\ B'$
using *v*[*unfolded e*] **by** *simp*
qed
qed
have $\exists M0.\ \forall r \in S.\ \forall M \geq M0.\ vpGge\ p\ ([:res\ r / of-nat\ p:] * (\sum a <$
 $p.\ \tau PoleP-tr\ p\ (Yp-tr\ p\ M)\ ((of-int\ r - of-nat\ a) / of-nat\ p)\ M)$
 $- [:res\ r * of-nat\ p^{\wedge} 4:] * \tau Pole\ r)\ (B + 4)$
by (*rule finite-eventually-all*[*OF S*]) (*rule pole*)
then obtain *M0* **where** $M0: \bigwedge r\ M.\ r \in S \implies M0 \leq M \implies$
 $vpGge\ p\ ([:res\ r / of-nat\ p:] * (\sum a <$
 $p.\ \tau PoleP-tr\ p\ (Yp-tr\ p\ M)\ ((of-int\ r - of-nat\ a) / of-nat\ p)\ M)$
 $- [:res\ r * of-nat\ p^{\wedge} 4:] * \tau Pole\ r)\ (B + 4)$ **by** *blast*
show *?thesis*
proof (*intro exI*[*of - M0*] *allI impI*)
fix *M* **assume** *M*: $M0 \leq M$
have *lhs*: $\tau X\ S\ P = [:tau\ Q:] + (\sum r \in S.\ [:res\ r:] * \tau Pole\ r)$

```

    by (simp add: tauX-def Q-def res-def)
  have diff: tauX S P - [:of-nat p powi (- 4):] *
    (∑ a < p. tauExt-tr p (Yp-tr p M) (distribNum p S P a) (distribPoles p S
a) M) =
    - [:of-nat p powi (- 4):] * (∑ r ∈ S. [:res r / of-nat p:] *
    (∑ a < p. tauPoleP-tr p (Yp-tr p M) ((of-int r - of-nat a) / of-nat p)
M)
    - [:res r * of-nat p ^ 4:] * tauPole r)
  unfolding lhs rhs sum-subtractf
  by (rule distribution-formula-aux) (use p0 in ⟨simp add: power-int-minus⟩)
  have vpGge p (∑ r ∈ S. [:res r / of-nat p:] *
    (∑ a < p. tauPoleP-tr p (Yp-tr p M) ((of-int r - of-nat a) / of-nat p)
M)
    - [:res r * of-nat p ^ 4:] * tauPole r) (B + 4)
  by (rule vpGge-sum[OF p S]) (rule M0[OF - M])
  moreover have vpGge p (- [:of-nat p powi (- 4):]) (- 4)
  by (rule vpGge-neg, rule vpGge-C) (simp add: vge-def vp-def qmultiplic-
ity-power-int[OF p])
  ultimately show vpGge p (tauX S P - [:of-nat p powi (- 4):] *
    (∑ a < p. tauExt-tr p (Yp-tr p M) (distribNum p S P a) (distribPoles p S
a) M)) B
  unfolding diff using vpGge-mul[OF p] by fastforce
qed
qed
end

```

```

theory Zeta5-Inner-Range
  imports Zeta5-Small-Primes Zeta5-Pole-Distribution Polynomial-Taylor-Shift
  Unimodular-Determinants
begin

```

21 The inner range: Proposition 4.1

Section 4.1 of the paper: the inner basis, extraction of powers of p from $g(c + pz)$, the entry bounds (4.2) and (4.3), and the unimodularity of the inner Gram matrix (Proposition 4.1, *DeltaK-vpGge-inner*). Lean theories *Section4*, *InnerBasis*, *InnerEntries*, *InnerFinal*.

21.1 Section 4.1 The inner range

(4.1): the inner range.

definition *InnerHyp* :: $\text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{bool}$ **where**
InnerHyp K M p $\longleftrightarrow 40 \leq M \wedge 40 \text{ dvd } K \wedge 0 < K \wedge$
 $200 * M \wedge^2 \leq K \wedge K < p * M \wedge 3 * p \leq K$

Under (4.1): $p > 200M$ and $p^2 > 5K$.

lemma *innerHyp-bounds*:
assumes *InnerHyp K M p*
shows $200 * M < p \wedge 5 * K < p^{\wedge} 2$
proof –
have *hM*: $40 \leq M$ **and** *hK*: $200 * M^{\wedge} 2 \leq K$ **and** *hp*: $K < p * M$
using *assms* **by** (*auto simp: InnerHyp-def*)
have $200 * M * M < p * M$ **using** *le-less-trans[OF hK hp]*
by (*simp only: power2-eq-square mult.assoc*)
hence *h1*: $200 * M < p$ **using** *hM* **by** *simp*
have $5 * K < p * (5 * M)$ **using** *hp* **by** *simp*
also have $\dots \leq p * p$ **using** *h1* **by** (*auto intro: mult-le-mono2*)
finally show *?thesis* **using** *h1* **by** (*simp add: power2-eq-square*)
qed

(4.4): $0 \leq E < m$ and $mT + E = h - L - 0 + 3(N - mN)$.

lemma *TE-spec*:
assumes *prime p InnerHyp K M p*
shows $0 \leq \text{Ein } K \ M \ p \wedge \text{Ein } K \ M \ p < \text{int } (m\text{Half } p) \wedge$
 $\text{int } (m\text{Half } p) * \text{Tin } K \ M \ p + \text{Ein } K \ M \ p = \text{rhs44 } K \ M \ p$
proof –
have $8000 < p$ **using** *innerHyp-bounds[OF assms(2)] assms(2)* **by** (*auto simp: InnerHyp-def*)
hence $0 < m\text{Half } p$ **by** (*simp add: mHalf-def*)
hence $0 < \text{int } (m\text{Half } p)$ **by** *simp*
thus *?thesis* **unfolding** *Ein-def Tin-def* **using** *pos-mod-sign pos-mod-bound mult-div-mod-eq*
by *presburger*
qed

(4.8): $\gamma \cdot p^{\wedge} \text{in}$ is an integer (twice a sum of half-integers).

lemma *two-wIn-Ints*: $2 * w\text{In } K \ M \ p \ a \ i \in (\mathbb{Z} :: \text{rat set})$
proof (*cases a = 0*)
case *True*
define *S* **where** $S = \text{insert } (2 * \text{of-nat } i + 6 * \text{of-nat } (\text{Nof } K \ \text{div } p) - \text{of-nat } (K \ \text{div } p) + 1 /$
 $2 :: \text{rat})$
 $((\lambda c. \text{of-int } (\text{Zin } K \ M \ p \ c) - (\text{of-nat } (\text{ell } p \ K \ c) + 4) / 2) ' \{1..m\text{Half } p\})$
have *wS*: $w\text{In } K \ M \ p \ a \ i = \text{Min } S$ **using** *True* **by** (*simp add: wIn-def S-def*)
have $\text{Min } S \in S$ **unfolding** *S-def* **by** (*rule Min-in*) *auto*
moreover have $2 * y \in \mathbb{Z}$ **if** $y \in S$ **for** *y*
proof –
from *that* **consider** $y = 2 * \text{of-nat } i + 6 * \text{of-nat } (\text{Nof } K \ \text{div } p) - \text{of-nat } (K \ \text{div } p) + 1 / 2$
 $| \ c$ **where** $y = \text{of-int } (\text{Zin } K \ M \ p \ c) - (\text{of-nat } (\text{ell } p \ K \ c) + 4) / 2$
unfolding *S-def* **by** *auto*
thus *?thesis*
proof *cases*
case *1*
have $2 * y = \text{of-int } (4 * \text{int } i + 12 * \text{int } (\text{Nof } K \ \text{div } p) - 2 * \text{int } (K \ \text{div } p))$

```

+ 1)
  using 1 by (simp add: algebra-simps)
  thus ?thesis by (metis Ints-of-int)
next
  case (2 c)
  have 2 * y = of-int (2 * Zin K M p c - int (ell p K c) - 4)
    using 2 by (simp add: field-simps)
  thus ?thesis by (metis Ints-of-int)
qed
qed
ultimately show ?thesis using wS by simp
next
  case False
  have 2 * wIn K M p a i = of-int (2 * int i + 2 * bIn K p a - int (ell p K a)
- 4)
    using False by (simp add: wIn-def field-simps)
  thus ?thesis by (metis Ints-of-int)
qed

```

lemma *gammaIn-isInt*: $\exists n::\text{int}. \text{gammaIn } K M p = \text{of-int } n$

```

proof -
  have gammaIn  $K M p \in \mathbb{Z}$ 
    unfolding gammaIn-def sum-distrib-left by (intro Ints-sum two-wIn-Ints)
  thus ?thesis by (auto elim: Ints-cases)
qed

```

Section 4.1: the dimensions $L-a$ are nonnegative and $L-0 + \sum -\{a \geq 1\} L-a = h$.

lemma *card-cong-class-le*:

```

assumes  $0 < p \ A \subseteq \{..N\} \wedge x y. x \in A \implies$ 
   $y \in A \implies x \bmod p = y \bmod p$ 
shows  $\text{card } A \leq N \text{ div } p + 1$ 
proof -
  have inj: inj-on  $(\lambda j. j \text{ div } p) A$ 
  proof (rule inj-onI)
    fix  $x y$  assume  $xy: x \in A \ y \in A \ x \text{ div } p = y \text{ div } p$ 
    have  $x = x \text{ div } p * p + x \bmod p$  by simp
    also have  $\dots = y \text{ div } p * p + y \bmod p$  using  $xy \text{ assms}(3)[OF xy(1,2)]$  by
simp
    also have  $\dots = y$  by simp
    finally show  $x = y$  .
  qed
  have  $(\lambda j. j \text{ div } p) ' A \subseteq \{..N \text{ div } p\}$  using assms(2)
    by (auto intro: div-le-mono)
  hence  $\text{card } ((\lambda j. j \text{ div } p) ' A) \leq \text{card } \{..N \text{ div } p\}$  by (intro card-mono) auto
  thus ?thesis using card-image[OF inj] by simp
qed

```

lemma *ell-le*:


```

    assumes  $0 < p$ 
    shows  $\text{ell } p \ N \ a \leq 2 * (N \text{ div } p) + 2$ 
  proof -
    define A1 where  $A1 = \{j \in \{1..N\}. j \text{ mod } p = a \text{ mod } p\}$ 
    define A2 where  $A2 = \{j \in \{1..N\}. (j + a) \text{ mod } p = 0\}$ 
    have  $\text{ell } p \ N \ a = \text{card } (A1 \cup A2)$  unfolding ell-def A1-def A2-def
      by (rule arg-cong[where  $f = \text{card}$ ]) auto
    also have  $\dots \leq \text{card } A1 + \text{card } A2$  by (rule card-Un-le)
    also have  $\text{card } A1 \leq N \text{ div } p + 1$ 
      by (rule card-cong-class-le[OF assms]) (auto simp: A1-def)
    also have  $\text{card } A2 \leq N \text{ div } p + 1$ 
    proof (rule card-cong-class-le[OF assms])
      show  $A2 \subseteq \{..N\}$  by (auto simp: A2-def)
    next
      fix  $x \ y$  assume  $x \in A2 \ y \in A2$ 
      hence  $[x + a = y + a] \text{ (mod } p)$  by (simp add: A2-def cong-def)
      hence  $[x = y] \text{ (mod } p)$  by (rule cong-add-rcancel-nat[THEN iffD1])
      thus  $x \text{ mod } p = y \text{ mod } p$  by (simp only: cong-def)
    qed
    finally show ?thesis by simp
  qed

lemma Lin-nonneg: prime  $p \implies \text{InnerHyp } K \ M \ p \implies$ 
   $a \leq m\text{Half } p \implies 0 \leq \text{Lin } K \ M \ p \ a$ 
proof -
  assume prime  $p$  and  $H: \text{InnerHyp } K \ M \ p$  and  $a \leq m\text{Half } p$ 
  show  $0 \leq \text{Lin } K \ M \ p \ a$ 
  proof (cases  $a = 0$ )
    case True
      thus ?thesis by (simp add: Lin-def)
    next
      case False
      from  $H$  have  $M40: 40 \leq M$  and  $K40: 40 \text{ dvd } K$  and  $KM: 200 * M^2 \leq K$ 
        and  $pK: 3 * p \leq K$  by (auto simp: InnerHyp-def)
      obtain  $k$  where  $k: K = 40 * k$  using  $K40$  by (elim dvdE)
      have  $hk: \text{hof } K = 37 * k$  and  $Nk: \text{Nof } K = 3 * k$  unfolding  $k$  hof-def Nof-def
by simp-all
      have  $p8: 8000 < p$  using innerHyp-bounds[OF H] M40 by linarith
      define  $m$  where  $m = m\text{Half } p$ 
      define  $q$  where  $q = \text{Nof } K \text{ div } p$ 
      have  $m0: 0 < m$  using  $p8$  by (simp add: m-def mHalf-def)
      have  $m2: 2 * m \leq p$  by (simp add: m-def mHalf-def)
      have  $q * p \leq 3 * k$  unfolding  $q$ -def  $Nk$ [symmetric] by (rule div-times-less-eq-dividend)
      hence  $qp: p * q \leq 3 * k$  by (simp add: mult.commute)
      have  $2 * m * q \leq p * q$  using  $m2$  by (rule mult-le-mono1)
      hence  $mq: 2 * (m * q) \leq p * q$  by (simp add: mult.assoc)
      have  $MM: 40 * M \leq M * M$  using  $M40$  by (rule mult-le-mono1)
      have  $kM: 200 * (M * M) \leq 40 * k$  using  $KM \ k$  by (simp add: power2-eq-square)
      have  $qq: q \leq p * q$  using  $p8$  by simp

```

```

have 6 * (m * (q + 1)) + 4 * M + 10 + 3 * q ≤ 46 * k
proof (cases q = 0)
  case True
    have e: m * (q + 1) = m using True by simp
    show ?thesis unfolding e using True m2 pK k MM kM M40 by linarith
  next
    case False
    have e: m * (q + 1) = m * q + m by simp
    have e2: 6 * (m * q + m) = 6 * (m * q) + 6 * m by (rule add-mult-distrib2)
    have pq: p ≤ p * q using False by simp
    have 6 * (m * q) ≤ 9 * k using m2 pq qp by linarith
    moreover have 6 * m ≤ 9 * k using m2 pq qp by linarith
    moreover have 3 * q ≤ 9 * k using qq qp by linarith
    moreover have 4 * M + 10 ≤ 19 * k using MM kM M40 by linarith
    ultimately show ?thesis unfolding e e2 by linarith
qed
hence int (6 * (m * (q + 1)) + 4 * M + 10 + 3 * q) ≤ int (46 * k)
  by (simp only: of-nat-le-iff)
hence r: int (6 * (q + 1)) * int m ≤ rhs44 K M p
  unfolding rhs44-def q-def[symmetric] unfolding hk Nk L0-def
  by (simp add: algebra-simps)
have T: int (6 * (q + 1)) ≤ Tin K M p
proof -
  have int (6 * (q + 1)) = (int (6 * (q + 1)) * int m) div int m using m0
by simp
  also have ... ≤ rhs44 K M p div int m by (rule zdiv-mono1[OF r]) (use m0
in simp)
  finally show ?thesis by (simp add: Tin-def m-def)
qed
have e: 0 ≤ epsIn K M p a by (simp add: epsIn-def)
have l: ell p (Nof K) a ≤ 2 * q + 2 unfolding q-def using p8 by (intro
ell-le) simp
show ?thesis using False T e l by (simp add: Lin-def bIn-def)
qed
qed

lemma card-multiples-upto:
  assumes 0 < p
  shows card {j ∈ {1..N}. p dvd j} = N div p
proof -
  have {j ∈ {1..N}. p dvd j} = (λi. p * i) ‘ {1..N div p}
proof (intro equalityI subsetI)
  fix j assume j ∈ {j ∈ {1..N}. p dvd j}
  then obtain i where j: j = p * i 1 ≤ p * i p * i ≤ N by (auto elim!: dvdE)
  have 1 ≤ i using j(2) by (cases i) auto
  moreover have i ≤ N div p using div-le-mono[OF j(3), of p] assms by simp
  ultimately show j ∈ (λi. p * i) ‘ {1..N div p} using j(1) by auto
next
  fix j assume j ∈ (λi. p * i) ‘ {1..N div p}

```

```

    then obtain  $i$  where  $i \in \{1..N \text{ div } p\}$   $j = p * i$  by blast
    hence  $i: j = p * i$   $1 \leq i \leq N \text{ div } p$  by auto
    have  $p * i \leq N$  using mult-le-mono2[OF  $i(3)$ , of  $p$ ] times-div-less-eq-dividend[of
 $p$   $N$ ]
      by linarith
    moreover have  $1 \leq p * i$  using  $i(2)$  assms by simp
    ultimately show  $j \in \{j \in \{1..N\}. p \text{ dvd } j\}$  using  $i(1)$  by auto
  qed
  moreover have inj-on  $(\lambda i. p * i)$   $\{1..N \text{ div } p\}$  using assms by (auto intro:
inj-onI)
  ultimately show ?thesis by (simp add: card-image)
qed

lemma sum-ell:
  assumes prime  $p$   $2 < p$ 
  shows  $(\sum a = 1..mHalf\ p. ell\ p\ N\ a) = N - N \text{ div } p$ 
proof -
  define  $m$  where  $m = mHalf\ p$ 
  have  $p0: 0 < p$  using assms by simp
  have odd  $p$  using assms by (metis prime-odd-nat)
  hence  $m2: 2 * m = p - 1$  unfolding m-def mHalf-def by (auto elim!: oddE)
  define  $C$  where  $C\ a = \{j \in \{1..N\}. j \bmod p = a \vee j \bmod p = p - a\}$  for  $a$ 
  have  $Ceq: \{j \in \{1..N\}. j \bmod p = a \bmod p \vee (j + a) \bmod p = 0\} = C\ a$ 
    if  $a: a \in \{1..m\}$  for  $a$ 
  proof -
    have  $ap: 0 < a$   $a < p$  using  $a\ m2$  by auto
    have  $(j + a) \bmod p = 0 \iff j \bmod p = p - a$  for  $j$ 
  proof -
    have  $r: j \bmod p < p$  using  $p0$  by simp
    have  $(j + a) \bmod p = (j \bmod p + a) \bmod p$  by (simp add: mod-add-left-eq)
    also have  $\dots = 0 \iff j \bmod p + a = p$ 
  proof
    assume  $h: (j \bmod p + a) \bmod p = 0$ 
    have  $j \bmod p + a < 2 * p$  using  $r\ ap$  by linarith
    moreover have  $p \text{ dvd } j \bmod p + a$  using  $h$  by (simp add: mod-eq-0-iff-dvd)
    then obtain  $t$  where  $t: j \bmod p + a = p * t$  by (elim dvdE)
    ultimately have  $t < 2$  by simp
    moreover have  $t \neq 0$ 
  proof
    assume  $t = 0$ 
    hence  $j \bmod p + a = 0$  using  $t$  by simp
    thus False using  $ap$  by simp
  qed
    ultimately show  $j \bmod p + a = p$  using  $t$  by (cases  $t$ ) auto
  qed simp
  finally show ?thesis using  $ap$  by linarith
qed
  thus ?thesis using  $ap$  by (auto simp: C-def)
qed

```

```

have ( $\sum a = 1..m. \text{ell } p \ N \ a$ ) = ( $\sum a = 1..m. \text{card } (C \ a)$ )
proof (rule sum.cong)
  fix a assume a:  $a \in \{1..m\}$ 
  show  $\text{ell } p \ N \ a = \text{card } (C \ a)$  unfolding ell-def Ceq[OF a] ..
qed simp
also have ... =  $\text{card } (\bigcup a \in \{1..m\}. C \ a)$ 
proof (rule card-UN-disjoint[symmetric])
  show  $\forall i \in \{1..m\}. \forall j \in \{1..m\}. i \neq j \longrightarrow$ 
     $C \ i \cap C \ j = \{\}$ 
  using m2 by (auto simp: C-def)
qed (auto simp: C-def)
also have  $(\bigcup a \in \{1..m\}. C \ a) = \{1..N\} - \{j \in \{1..N\}. p \ \text{dvd } j\}$ 
proof (intro equalityI subsetI)
  fix j assume  $j \in (\bigcup a \in \{1..m\}. C \ a)$ 
  then obtain a where a:  $a \in \{1..m\} \ j \in C \ a$  by auto
  hence  $j \bmod p \neq 0$  using m2 by (auto simp: C-def)
  thus  $j \in \{1..N\} - \{j \in \{1..N\}. p \ \text{dvd } j\}$  using a by (auto simp: C-def
dvd-eq-mod-eq-0)
next
  fix j assume  $j: j \in \{1..N\} - \{j \in \{1..N\}. p \ \text{dvd } j\}$ 
  hence  $r0: j \bmod p \neq 0$  by (auto simp: dvd-eq-mod-eq-0)
  have  $rp: j \bmod p < p$  using p0 by simp
  show  $j \in (\bigcup a \in \{1..m\}. C \ a)$ 
  proof (cases  $j \bmod p \leq m$ )
    case True
    thus ?thesis using j r0 by (auto simp: C-def intro!: bexI[of - j mod p])
  next
    case False
    hence  $p - j \bmod p \in \{1..m\}$  using rp m2 by auto
    moreover have  $j \in C \ (p - j \bmod p)$  using j rp by (auto simp: C-def)
    ultimately show ?thesis by blast
  qed
qed
also have  $\text{card } (\{1..N\} - \{j \in \{1..N\}. p \ \text{dvd } j\}) = N - N \ \text{div } p$ 
  by (subst card-Diff-subset) (use card-multiples-upto[OF p0, of N] in auto)
finally show ?thesis by (simp add: m-def)
qed

lemma rank-props:
  fixes r :: 'a  $\Rightarrow$  'a  $\Rightarrow$  bool
  assumes A: finite A and irr:  $\bigwedge x. \neg r \ x \ x$  and tr:  $\bigwedge x \ y \ z. r \ x \ y$ 
     $\Longrightarrow r \ y \ z \Longrightarrow r \ x \ z$ 
  and tot:  $\bigwedge x \ y. x \in A \Longrightarrow y \in A \Longrightarrow$ 
     $x \neq y \Longrightarrow r \ x \ y \vee r \ y \ x$ 
  defines rk  $\equiv \lambda x. \text{card } \{c \in A. r \ c \ x\}$ 
  shows  $\bigwedge x \ y. x \in A \Longrightarrow y \in A \Longrightarrow$ 
     $rk \ x < rk \ y \longleftrightarrow r \ x \ y$ 
    and bij-betw rk A  $\{..<\text{card } A\}$ 
proof -

```

```

have mono:  $rk\ x < rk\ y$  if  $x \in A\ y \in A\ r\ x\ y$  for  $x\ y$ 
proof -
  have  $\{c \in A. r\ c\ x\} \subseteq \{c \in A. r\ c\ y\}$ 
    using that irr tr by blast
  thus ?thesis unfolding rk-def by (rule psubset-card-mono[rotated]) (use A in
simp)
qed
show  $rk\ x < rk\ y \iff r\ x\ y$  if  $x \in A\ y \in A$  for  $x\ y$ 
proof
  assume  $rk\ x < rk\ y$ 
  thus  $r\ x\ y$  using mono[OF that(2,1)] tot[OF that] by (cases  $x = y$ ) auto
qed (rule mono[OF that])
have inj: inj-on  $rk\ A$ 
proof (rule inj-onI)
  fix  $x\ y$  assume  $x \in A\ y \in A\ rk\ x = rk\ y$ 
  thus  $x = y$  using mono tot by (metis less-irrefl)
qed
have sub:  $rk\ ` A \subseteq \{.. $card\ A\}$ 
proof
  fix  $i$  assume  $i \in rk\ ` A$ 
  then obtain  $x$  where  $x: x \in A\ i = rk\ x$  by auto
  have  $\{c \in A. r\ c\ x\} \subseteq A - \{x\}$  using irr by auto
  hence  $rk\ x \leq card\ (A - \{x\})$  unfolding rk-def by (intro card-mono) (use A
in auto)
  also have  $\dots < card\ A$  by (rule card-Diff1-less[OF  $A\ x(1)$ ])
  finally show  $i \in \{.. $card\ A\}$  using  $x$  by simp
qed
have  $rk\ ` A = \{.. $card\ A\}$ 
  by (rule card-subset-eq) (use sub card-image[OF inj] in auto)
thus bij-betw  $rk\ A\ \{.. $card\ A\}$  using inj by (simp add: bij-betw-def)
qed

lemma sum-epsIn:
  assumes prime  $p$  InnerHyp  $K\ M\ p$ 
  shows  $(\sum\ a = 1..mHalf\ p. epsIn\ K\ M\ p\ a) = Ein\ K\ M\ p$ 
proof -
  define  $m$  where  $m = mHalf\ p$ 
  define  $A$  where  $A = \{1..m\}$ 
  define  $r$  where  $r\ c\ a \iff ell\ p\ K\ a < ell\ p\ K\ c \vee$ 
    ( $ell\ p\ K\ c = ell\ p\ K\ a \wedge c < a$ ) for  $c\ a$ 
  have rk: defaultPos  $K\ p\ a = card\ \{c \in A. r\ c\ a\}$  for  $a$ 
    by (simp add: defaultPos-def A-def m-def r-def)
  have irr:  $\bigwedge x. \neg r\ x\ x$  and tr:  $\bigwedge x\ y\ z. r\ x\ y \implies$ 
     $r\ y\ z \implies r\ x\ z$ 
  and tot:  $\bigwedge x\ y. x \in A \implies y \in A \implies$ 
     $x \neq y \implies r\ x\ y \vee r\ y\ x$ 
  unfolding r-def by auto
  note RP = rank-props[of  $A\ r$ , OF - irr tr tot, folded rk]
  have cA:  $card\ A = m$  by (simp add: A-def)$$$$ 
```

```

have iff:  $x \in A \implies y \in A \implies$ 
  defaultPos  $K\ p\ x < \text{defaultPos } K\ p\ y \iff r\ x\ y$  for  $x\ y$ 
using RP(1) by (simp add: A-def)
have bij: bij-betw (defaultPos  $K\ p$ )  $A\ \{..<m\}$  using RP(2) cA by (simp add:
A-def)
have TE:  $0 \leq \text{Ein } K\ M\ p\ \text{Ein } K\ M\ p < \text{int } m$  using TE-spec[OF assms] by
(auto simp: m-def)
define E where  $E = \text{nat } (\text{Ein } K\ M\ p)$ 
have E:  $\text{Ein } K\ M\ p = \text{int } E\ E \leq m$  using TE by (auto simp: E-def)
have eps:  $\text{epsIn } K\ M\ p\ a = (\text{if } \text{defaultPos } K\ p\ a < E \text{ then } 1 \text{ else } 0)$  if  $a: a \in A$ 
for  $a$ 
proof -
  have  $\{c \in \{1..m\text{Half } p\}. \text{defaultPos } K\ p\ c < \text{defaultPos } K\ p\ a\} = \{c \in A. r\ c\ a\}$ 
  using iff[OF - a] by (auto simp: A-def m-def)
  thus ?thesis by (simp add: epsIn-def rk[symmetric] E)
qed
have  $(\sum a = 1..m\text{Half } p. \text{epsIn } K\ M\ p\ a) = (\sum a \in A. \text{if } \text{defaultPos } K\ p\ a < E \text{ then } 1 \text{ else } 0)$ 
  unfolding A-def m-def by (rule sum.cong) (simp-all add: eps[unfolded A-def
m-def])
also have  $\dots = \text{int } (\text{card } \{a \in A. \text{defaultPos } K\ p\ a < E\})$ 
  by (simp add: sum.If-cases A-def Int-def)
also have  $\text{card } \{a \in A. \text{defaultPos } K\ p\ a < E\} = \text{card } \{..<E\}$ 
proof -
  have inj: inj-on (defaultPos  $K\ p$ )  $\{a \in A. \text{defaultPos } K\ p\ a < E\}$ 
  using bij by (auto simp: bij-betw-def intro: inj-on-subset)
  have defaultPos  $K\ p\ ' \{a \in A. \text{defaultPos } K\ p\ a < E\} = \{..<E\}$ 
  proof (intro equalityI subsetI)
    fix i assume  $i \in \{..<E\}$ 
    hence  $i \in \{..<m\}$  using E by auto
    hence  $i \in \text{defaultPos } K\ p\ ' A$  using bij by (simp add: bij-betw-def)
    then obtain a where  $a \in A\ \text{defaultPos } K\ p\ a = i$  by blast
    thus  $i \in \text{defaultPos } K\ p\ ' \{a \in A. \text{defaultPos } K\ p\ a < E\}$ 
    using  $\langle i \in \{..<E\} \rangle$  by auto
  qed auto
  thus ?thesis using card-image[OF inj] by simp
qed
finally show ?thesis using E by simp
qed

lemma sum-Lin:  $\text{prime } p \implies \text{InnerHyp } K\ M\ p \implies$ 
   $(\sum a \leq m\text{Half } p. \text{Lin } K\ M\ p\ a) = \text{int } (\text{hof } K)$ 
proof -
  assume p:  $\text{prime } p$  and H:  $\text{InnerHyp } K\ M\ p$ 
  define m where  $m = m\text{Half } p$ 
  have p8:  $8000 < p$  using innerHyp-bounds[OF H] H by (auto simp: Inner-
Hyp-def)
  have  $\{..m\} = \text{insert } 0\ \{1..m\}$  by auto

```

hence $(\sum a \leq m. \text{Lin } K \ M \ p \ a) = \text{Lin } K \ M \ p \ 0 + (\sum a = 1..m. \text{Lin } K \ M \ p \ a)$
 by *simp*
 also have $(\sum a = 1..m. \text{Lin } K \ M \ p \ a)$
 $= (\sum a = 1..m. \text{Tin } K \ M \ p - 3 * \text{int } (\text{ell } p \ (\text{Nof } K) \ a) + \text{epsIn } K \ M \ p \ a)$
 by *(rule sum.cong) (auto simp: Lin-def bIn-def)*
 also have $\dots = \text{int } m * \text{Tin } K \ M \ p - 3 * \text{int } (\sum a = 1..m. \text{ell } p \ (\text{Nof } K) \ a)$
 $+ (\sum a = 1..m. \text{epsIn } K \ M \ p \ a)$
 by *(simp add: sum.distrib sum-subtractf sum-distrib-left of-nat-sum)*
 also have $(\sum a = 1..m. \text{ell } p \ (\text{Nof } K) \ a) = \text{Nof } K - \text{Nof } K \ \text{div } p$
 unfolding *m-def* by *(rule sum-ell[OF p]) (use p8 in simp)*
 also have $(\sum a = 1..m. \text{epsIn } K \ M \ p \ a) = \text{Ein } K \ M \ p$
 unfolding *m-def* by *(rule sum-epsIn[OF p H])*
 finally have $(\sum a \leq m. \text{Lin } K \ M \ p \ a)$
 $= \text{int } (L0 \ M) + (\text{int } m * \text{Tin } K \ M \ p + \text{Ein } K \ M \ p) - 3 * \text{int } (\text{Nof } K - \text{Nof } K \ \text{div } p)$
 by *(simp add: Lin-def)*
 also have $\text{int } m * \text{Tin } K \ M \ p + \text{Ein } K \ M \ p = \text{rhs44 } K \ M \ p$ using *TE-spec[OF p H]* by *(simp add: m-def)*
 also have $\text{int } (\text{Nof } K - \text{Nof } K \ \text{div } p) = \text{int } (\text{Nof } K) - \text{int } (\text{Nof } K \ \text{div } p)$ by *(simp add: of-nat-diff)*
 finally show $(\sum a \leq m \text{Half } p. \text{Lin } K \ M \ p \ a) = \text{int } (\text{hof } K)$ by *(simp add: rhs44-def m-def)*
 qed

lemma *length-inner-idx: prime p $\implies$ InnerHyp K M p $\implies$*

length (inner-idx K M p) = hof K

proof –

assume *p: prime p* and *H: InnerHyp K M p*

have $\text{length } (\text{inner-idx } K \ M \ p) = (\sum a \in \text{set } [0..<m\text{Half } p + 1]. \text{nat } (\text{Lin } K \ M \ p \ a))$

unfolding *inner-idx-def length-concat map-map*

by *(simp add: o-def sum-list-distinct-conv-sum-set)*

also have $\text{set } [0..<m\text{Half } p + 1] = \{..m\text{Half } p\}$ by *auto*

also have $(\sum a \leq m\text{Half } p. \text{nat } (\text{Lin } K \ M \ p \ a)) = \text{hof } K$

proof –

have $\text{int } (\sum a \leq m\text{Half } p. \text{nat } (\text{Lin } K \ M \ p \ a)) = (\sum a \leq m\text{Half } p. \text{Lin } K \ M \ p \ a)$

unfolding *of-nat-sum* by *(rule sum.cong) (use Lin-nonneg[OF p H] in auto)*

also have $\dots = \text{int } (\text{hof } K)$ by *(rule sum-Lin[OF p H])*

finally show *?thesis* by *(simp only: of-nat-eq-iff)*

qed

finally show $\text{length } (\text{inner-idx } K \ M \ p) = \text{hof } K$.

qed

(4.5): the inner basis is $\mathbb{Z}$ -*p*-unimodular: the coefficient matrix of the basis polynomials has a determinant that is a *p*-adic unit (CRT basis lemma over $\mathbb{Z}/p\mathbb{Z}$).

The index list is lexicographically ordered: an earlier position has a smaller

class, or the same class and a smaller exponent.

lemma *inner-idx-sorted*:

sorted-wrt ($\lambda u v. \text{fst } u < \text{fst } v \vee (\text{fst } u = \text{fst } v \wedge \text{snd } u < \text{snd } v)$)
 $(\text{concat } (\text{map } (\lambda a. \text{map } (\lambda i. (a, i)) [0..<f a]) [0..<n]))$

proof (*induction n*)

case 0

show ?case **by** *simp*

next

case (*Suc n*)

have *sorted-wrt* ($\lambda u v. \text{fst } u < \text{fst } v \vee (\text{fst } u = \text{fst } v \wedge$
 $\text{snd } u < \text{snd } v)$) ($\text{map } (\lambda i. (n, i)) [0..<f n]$)

unfolding *sorted-wrt-map* **by** (*simp add: sorted-wrt-upt*)

with *Suc* **show** ?case **by** (*auto simp: sorted-wrt-append*)

qed

lemma *inner-idx-lex*:

assumes $y < x$ $x < \text{length } (\text{inner-idx } K M p)$

shows $\text{fst } (\text{inner-idx } K M p ! y) < \text{fst } (\text{inner-idx } K M p ! x) \vee$
 $(\text{fst } (\text{inner-idx } K M p ! y) = \text{fst } (\text{inner-idx } K M p ! x) \wedge$
 $\text{snd } (\text{inner-idx } K M p ! y) < \text{snd } (\text{inner-idx } K M p ! x))$

using *sorted-wrt-nth-less*[*OF inner-idx-sorted assms(1) assms(2)*][*unfolded inner-idx-def*]]

unfolding *inner-idx-def* .

lemma *vpGge-innerBasis*: $\text{prime } p \implies \text{vpGge } p (\text{innerBasis } K M p x) 0$

proof –

assume p : *prime p*

have f : $\text{vpGge } p (pX + [: \text{of-nat } c \wedge 2:]) 0$ **for** c

proof –

have $\text{vge } p (\text{of-nat } c \wedge 2) 0$ **using** *vge-pow*[*OF p vge-of-int*][*OF p, of int c*] **by**
simp

thus ?thesis **by** (*intro vpGge-add*)[*OF p vpGge-pX vpGge-C*])

qed

have fp : $\text{vpGge } p ((pX + [: \text{of-nat } c \wedge 2:]) \wedge n) 0$ **for** $c n$ **using** *vpGge-pow*[*OF*
 $p f [\text{of } c], \text{ of } n]$

by *simp*

have pr : $\text{vpGge } p (\prod c \in \{0..m\text{Half } p\} - \{\text{fst } x\}. (pX +$
 $[: \text{of-nat } c \wedge 2:]) \wedge \text{nat } (\text{Lin } K M p c)) 0$

using *vpGge-prod*[*OF p, of* { $0..m\text{Half } p\} - \{\text{fst } x\}$ $\lambda c. (pX +$
 $[: \text{of-nat } c \wedge 2:]) \wedge \text{nat } (\text{Lin } K M p c) \lambda \cdot. 0]$ fp

by *simp*

show ?thesis **unfolding** *innerBasis-def* **using** *vpGge-mul*[*OF p pr fp*] **by** *simp*

qed

Shapes of pulled-back numerators (Lean: *FarGood, Decomposition*).

definition *FarGood* :: $\text{nat} \Rightarrow \text{rat poly} \Rightarrow \text{bool}$ **where**

FarGood $p F \iff \text{vpGge } p F 0 \wedge (\forall d. 1 \leq d \implies$
 $\text{vge } p (\text{Polynomial.coeff } F d) 1)$

definition *Decomp* :: *nat* $\Rightarrow$ *nat* $\Rightarrow$ *nat* $\Rightarrow$ *rat poly* $\Rightarrow$ *bool* **where**
Decomp *p e n Q* $\longleftrightarrow (\exists N F. Q = [:of\text{-}nat\ p \wedge e:] * N * F \wedge$
vpGge *p N 0* $\wedge$ *degree* *N* $\leq n \wedge$ *FarGood* *p F*)

The number of *p*'s in $j^2 - (c + p z)^2 = (j - c - p z)(j + c + p z)$.

definition *omg* :: *nat* $\Rightarrow$ *nat* $\Rightarrow$ *nat* $\Rightarrow$ *nat* **where**
omg *p c j* = (if *int p dvd (int j - int c)* then 1 else 0) + (if *int p dvd (int j + int c)* then 1 else 0)

The exponent of *p* extracted from the pulled-back numerator at the class *c*.

definition *eNum* :: *nat* $\Rightarrow$ *nat* $\Rightarrow$ *nat set* $\Rightarrow$ (*nat* $\Rightarrow$ *nat*) **where**
eNum *p c J k* = 5 * (if *int p dvd int c* then 1 else 0) + ($\sum j \in J. k j * omg\ p\ c\ j$)

The near poles of the class *c*: $r \in \pm[1, K]$ with $r \equiv c \pmod{p}$.

definition *nearCount* :: *nat* $\Rightarrow$ *nat* $\Rightarrow$ *nat* $\Rightarrow$ *nat* **where**
nearCount *p K c* = *card* {*r* $\in$ *signedPoles* {1..*K*}. *int p dvd (r - int c)*}

21.2 Extracting powers of *p* from $g(c + p z)$

lemma *FarGood-one*: *FarGood* *p 1*
unfolding *FarGood-def* **by** (*auto simp: vpGge-one vge-zero*)

lemma *FarGood-mult*: *prime p* $\implies$ *FarGood* *p F* $\implies$ *FarGood* *p G* $\implies$ *FarGood* *p (F * G)*

proof –

assume *p*: *prime p* **and** *F*: *FarGood* *p F* **and** *G*: *FarGood* *p G*
have *F0*: $\bigwedge i. vge\ p\ (Polynomial.coeff\ F\ i)\ 0$ **and** *G0*: $\bigwedge i. vge\ p\ (Polynomial.coeff\ G\ i)\ 0$
using *F G* **unfolding** *FarGood-def vpGge-iff* **by** *auto*
have *F1*: $\bigwedge i. 1 \leq i \implies vge\ p\ (Polynomial.coeff\ F\ i)\ 1$
and *G1*: $\bigwedge i. 1 \leq i \implies vge\ p\ (Polynomial.coeff\ G\ i)\ 1$
using *F G* **unfolding** *FarGood-def* **by** *auto*
have *vpGge* *p (F * G) 0* **using** *vpGge-mul[OF p, of F 0 G 0]* *F G* **unfolding** *FarGood-def* **by** *simp*
moreover **have** *vge* *p (Polynomial.coeff (F * G) d) 1* **if** *d*: $1 \leq d$ **for** *d*
unfolding *coeff-mult*
proof (*rule vge-sum[OF p]*)
show *finite* {..*d*} **by** *simp*
next
fix *i* **assume** *i*: $i \in \{..d\}$
show *vge* *p (Polynomial.coeff F i * Polynomial.coeff G (d - i)) 1*
proof (*cases i = 0*)
case *True*
have *vge* *p (Polynomial.coeff F i * Polynomial.coeff G (d - i)) (0 + 1)*
by (*rule vge-mul[OF p F0 G1]*) (*use True d in simp*)

```

      thus ?thesis by simp
    next
      case False
      have vge p (Polynomial.coeff F i * Polynomial.coeff G (d - i)) (1 + 0)
        by (rule vge-mul[OF p F1 G0]) (use False in simp)
      thus ?thesis by simp
    qed
  qed
  ultimately show ?thesis unfolding FarGood-def by blast
qed

lemma Decomp-one:  $\text{Decomp } p \ 0 \ 0 \ 1$ 
  unfolding Decomp-def by (rule exI[of - 1], rule exI[of - 1]) (simp add: vpGge-one FarGood-one)

lemma Decomp-mult:
  prime p  $\implies \text{Decomp } p \ e \ n \ Q \implies \text{Decomp } p \ e' \ n' \ Q' \implies$ 
   $\text{Decomp } p \ (e + e') \ (n + n') \ (Q * Q')$ 
proof -
  assume p: prime p and D1:  $\text{Decomp } p \ e \ n \ Q$  and D2:  $\text{Decomp } p \ e' \ n' \ Q'$ 
  obtain N F where Q:  $Q = [:of\text{-}nat \ p \ \wedge \ e:] * N * F$  and N:  $vpGge \ p \ N \ 0$  and
  dN:  $\text{degree } N \leq n$ 
  and F: FarGood p F using D1 unfolding Decomp-def by blast
  obtain N' F' where Q':  $Q' = [:of\text{-}nat \ p \ \wedge \ e':] * N' * F'$ 
  F' and N':  $vpGge \ p \ N' \ 0$  and dN':  $\text{degree } N' \leq n'$ 
  and F': FarGood p F' using D2 unfolding Decomp-def by blast
  have  $Q * Q' = [:of\text{-}nat \ p \ \wedge \ (e + e'):] * (N * N') * (F * F')$ 
  unfolding Q Q' power-add by (simp add: mult-ac)
  moreover have  $vpGge \ p \ (N * N') \ 0$  using vpGge-mul[OF p N N'] by simp
  moreover have  $\text{degree } (N * N') \leq n + n'$  using degree-mult-le[of N N'] dN
  dN' by linarith
  moreover have FarGood p (F * F') by (rule FarGood-mult[OF p F F'])
  ultimately show ?thesis unfolding Decomp-def by blast
qed

lemma Decomp-pow: prime p  $\implies \text{Decomp } p \ e \ n \ Q \implies$ 
   $\text{Decomp } p \ (k * e) \ (k * n) \ (Q \wedge^k)$ 
proof (induction k)
  case 0
  show ?case using Decomp-one by simp
next
  case (Suc k)
  have  $\text{Decomp } p \ (k * e + e) \ (k * n + n) \ (Q \wedge^k * Q)$ 
  by (rule Decomp-mult[OF Suc.prem1] Suc.IH[OF Suc.prem2] Suc.prem2)
  thus ?case by (simp add: ac-simps)
qed

lemma Decomp-C-unit: prime p  $\implies \text{Decomp } p \ e \ n \ Q \implies$ 
   $vge \ p \ u \ 0 \implies \text{Decomp } p \ e \ n \ ([:u:] * Q)$ 

```

proof –
 assume p : prime p and D : *Decomp* $p \in n$ Q and u : $vge\ p\ u\ 0$
 obtain $N\ F$ where Q : $Q = [:of\text{-}nat\ p\ \wedge\ e:] * N * F$ and N : $vpGge\ p\ N\ 0$ and
 dN : $degree\ N \leq n$
 and F : *FarGood* $p\ F$ using D unfolding *Decomp-def* by *blast*
 have e : $[:u:] * Q = [:of\text{-}nat\ p\ \wedge\ e:] * N * ([:u:] * F)$ unfolding Q by (*simp add*:
mult-ac)
 have $vpGge\ p\ ([:u:] * F)\ 0$ using $vpGge\text{-}mul[OF\ p\ vpGge\text{-}C[OF\ u],\ of\ F\ 0]\ F$
 unfolding *FarGood-def*
 by *simp*
 moreover have $vge\ p\ (Polynomial.coeff\ ([:u:] * F)\ d)\ 1$ if d : $1 \leq d$ for d
proof –
 have $vge\ p\ (u * Polynomial.coeff\ F\ d)\ (0 + 1)$
 by (*rule vge-mul[OF\ p\ u]*) (*use F\ d in* $\langle simp\ add:\ FarGood\text{-}def \rangle$)
 thus $?thesis$ by *simp*
qed
 ultimately have *FarGood* $p\ ([:u:] * F)$ unfolding *FarGood-def* by *blast*
 thus $?thesis$ using $e\ N\ dN$ unfolding *Decomp-def* by *blast*
qed

A linear factor $\alpha + \sigma\ p\ z$ ($\sigma = \pm 1$) is $p \cdot (near)$ if $p \mid \alpha$, and far-good otherwise.

lemma *Decomp-linear*:

assumes prime $p\ \sigma = 1 \vee \sigma = -1$
 shows *Decomp* p (if $int\ p\ dvd\ \alpha$ then 1 else 0) (if $int\ p\ dvd\ \alpha$ then 1 else 0)
 $[:of\text{-}int\ \alpha,\ of\text{-}int\ (\sigma * int\ p):]$
proof (*cases int\ p\ dvd\ \alpha*)
 case *True*
 then obtain β where b : $\alpha = int\ p * \beta$ by (*elim dvdE*)
 have e : $[:of\text{-}int\ \alpha,\ of\text{-}int\ (\sigma * int\ p):] =$
 $[:of\text{-}nat\ p\ \wedge\ 1:] * [:of\text{-}int\ \beta,\ of\text{-}int\ \sigma:] * (1 :: rat\ poly)$
 unfolding b by (*simp add: mult.commute*)
 have N : $vpGge\ p\ [:of\text{-}int\ \beta,\ of\text{-}int\ \sigma :: rat:]\ 0$ unfolding *vpGge-iff*
 by (*auto simp: coeff-pCons vge-of-int[OF\ assms(1)] vge-zero split: nat.split*)
 have dN : $degree\ [:of\text{-}int\ \beta,\ of\text{-}int\ \sigma :: rat:] \leq 1$
 using *degree-pCons-le[of\ of-int\ \beta\ [:of-int\ \sigma :: rat:]]* by *simp*
 have *Decomp* $p\ 1\ 1\ [:of\text{-}int\ \alpha,\ of\text{-}int\ (\sigma * int\ p):]$
 unfolding *Decomp-def* by (*rule exI[of - [:of-int\ \beta,\ of-int\ \sigma:]]*,
rule exI[of - 1]) (*intro conjI e\ N\ dN\ FarGood-one*)
 thus $?thesis$ using *True* by *simp*
 next
 case *False*
 have $vp1$: $vge\ p\ (of\text{-}nat\ p)\ 1$ using *qmultiplicity-power-int[OF\ assms(1), of\ 1]*
 by (*simp add: vge-def vp-def*)
 have $vp1'$: $vge\ p\ (-\ of\text{-}nat\ p)\ 1$ by (*rule vge-neg[OF\ vp1]*)
 have $vp0$: $vge\ p\ (of\text{-}nat\ p)\ 0$ and $vp0'$: $vge\ p\ (-\ of\text{-}nat\ p)\ 0$
 by (*rule vge-mono[OF\ vp1]*, *simp*, *rule vge-mono[OF\ vp1']*, *simp*)
 have FG : *FarGood* $p\ [:of\text{-}int\ \alpha,\ of\text{-}int\ (\sigma * int\ p) :: rat:]$
 unfolding *FarGood-def vpGge-iff*

```

using assms(2) by (auto simp: coeff-pCons vge-of-int[OF assms(1)] vge-zero
vp1 vp1' vp0 vp0'
split: nat.split)
have e: [:of-int  $\alpha$ , of-int ( $\sigma * \text{int } p$ ):] =
[:of-nat  $p \wedge 0$ :] * 1 * [:of-int  $\alpha$ , of-int ( $\sigma * \text{int } p$ ) :: rat:]
by simp
have Decomp p 0 0 [:of-int  $\alpha$ , of-int ( $\sigma * \text{int } p$ ):]
unfolding Decomp-def by (rule exI[of - 1], rule exI[of - [:of-int  $\alpha$ ,
of-int ( $\sigma * \text{int } p$ ):]]) (intro conjI e vpGge-one FG, simp)
thus ?thesis using False by simp
qed

```

lemma *Decomp-prod*:

```

prime p  $\implies$  finite J  $\implies$  ( $\bigwedge j. j \in J \implies$ 
Decomp p (e j) (e j) (f j))  $\implies$ 
Decomp p ( $\sum j \in J. e j$ ) ( $\sum j \in J. e j$ ) ( $\prod j \in J. f j$ )
proof –
assume p: prime p and J: finite J and D:  $\bigwedge j. j \in J \implies$ 
Decomp p (e j) (e j) (f j)
show ?thesis using J D
proof (induction J rule: finite-induct)
case empty
show ?case using Decomp-one by simp
next
case (insert a J)
have Decomp p ( $e a + (\sum j \in J. e j)$ ) ( $e a + (\sum j \in$ 
J. e j)) ( $f a * (\prod j \in J. f j)$ )
by (rule Decomp-mult[OF p]) (use insert in auto)
thus ?case unfolding sum.insert[OF insert.hyps] prod.insert[OF insert.hyps] .
qed
qed

```

The pulled-back numerator $\varepsilon x^5 \prod -j (j^2 - x^2)^{\sim\{k-j\}}$ at $x = c + p z$: every factor $j^2 - (c + p z)^2 = (j - c - p z)(j + c + p z)$ contributes *omg* $p c j$ powers of p .

lemma *decomp-pullback*:

```

assumes p: prime p and J: finite J and  $\varepsilon$ : vge p  $\varepsilon$  0
shows Decomp p (eNum p c J k) (eNum p c J k)
(pcompose ([: $\varepsilon$ :] *  $pX^5$  * pcompose ( $\prod j \in$ 
J. (pX + [:of-nat j^2:]) ^ k j) (- ( $pX^2$ ))))
([:of-nat  $c$ :] + [:of-nat  $p$ :] *  $pX$ ))
proof –
define L where  $L = [:of-nat c:] + [:of-nat p:] * (pX :: \text{rat poly})$ 
have Leq:  $L = [:of-int (\text{int } c), of-int (1 * \text{int } p):]$ 
unfolding L-def by (simp add: monom-Suc)
have fac:  $\bigwedge j. \text{pcompose} (\text{pcompose} (pX + [:of-nat j^2:]) (- (pX^2))) L =$ 
[:of-int ( $\text{int } j - \text{int } c$ ), of-int ( $- 1 * \text{int } p$ ):] * [:of-int ( $\text{int } j + \text{int } c$ ),
of-int ( $1 * \text{int } p$ ):]
proof –

```

```

fix j :: nat
have e: poly (pcompose (pcompose (pX + [:of-nat j ^ 2:]) (- (pX ^ 2))) L) x
=
  poly ([:of-int (int j - int c), of-int (- 1 * int p):] * [:of-int (int j + int c),
    of-int (1 * int p):]) x for x :: rat
  unfolding L-def by (simp add: poly-pcompose poly-monom algebra-simps
    power2-eq-square)
  show pcompose (pcompose (pX + [:of-nat j ^ 2:]) (- (pX ^ 2))) L =
    [:of-int (int j - int c), of-int (- 1 * int p):] * [:of-int (int j + int c),
      of-int (1 * int p):]
  by (rule poly-eq-of-eval-nat) (rule e)
qed
have Q: pcompose ([:ε:] * pX ^ 5 * pcompose (∏ j ∈
  J. (pX + [:of-nat j ^ 2:]) ^ k j) (- (pX ^ 2))) L =
  [:ε:] * (L ^ 5 * (∏ j ∈
    J. ([:of-int (int j - int c), of-int (- 1 * int p):] *
      [:of-int (int j + int c), of-int (1 * int p):]) ^ k j))
  unfolding pcompose-mult pcompose-prod pcompose-power-left pcompose-pX pcom-
    pose-const fac
  by (simp only: mult.assoc)
have L5: Decomp p (5 * (if int p dvd int c then 1 else 0)) (5 *
  (if int p dvd int c then 1 else 0)) (L ^ 5)
  unfolding Leq by (rule Decomp-pow[OF p Decomp-linear[OF p, of 1 int c]])
simp
have Pr: Decomp p (∑ j ∈ J. k j * omg p c j) (∑ j ∈ J. k j * omg p c j)
  (∏ j ∈ J. ([:of-int (int j - int c), of-int (- 1 * int p):] *
    [:of-int (int j + int c), of-int (1 * int p):]) ^ k j)
proof (rule Decomp-prod[OF p J])
  fix j assume j ∈ J
  show Decomp p (k j * omg p c j) (k j * omg p c j)
    (([:of-int (int j - int c), of-int (- 1 * int p):] * [:of-int (int j + int c),
      of-int (1 * int p):]) ^ k j)
  unfolding omg-def by (rule Decomp-pow[OF p Decomp-mult[OF p De-
    comp-linear[OF p]
      Decomp-linear[OF p]]]) simp-all
qed
show ?thesis
  unfolding L-def[symmetric] Q eNum-def
  by (rule Decomp-C-unit[OF p - ε]) (rule Decomp-mult[OF p L5 Pr])
qed

lemma tauExt-tr-C-mul: tauExt-tr p Y ([:a:] * U) S M = [:a:] * tauExt-tr p Y U
  S M
proof -
  have d: ([:a:] * U) div qPoleProd S = Polynomial.smult a (U div qPoleProd S)
  by (simp add: div-smult-left)
  have t: tauK (Polynomial.smult a P) = a * tauK P for P by (simp add:
    tauK-eq-tau tau-smult)
  have r: qRes S ([:a:] * U) s = a * qRes S U s for s by (simp add: qRes-def)

```

show *?thesis* **unfolding** *tauExt-tr-def d t r* **by** (*simp add: smult-add-right smult-sum-right*)
qed

lemma *vpGge-Yp-tr: prime p $\implies \gamma \leq p \implies vpGge p (Yp-tr p M) 0$*

proof –

assume *p: prime p and p7: $\gamma \leq p$*
have *v: vge p (of-nat p) 0* **using** *vge-of-int[OF p, of int p]* **by** *simp*
have *a: vpGge p ([of-nat p ^ 5:] * pX) 0*
using *vpGge-mul[OF p vpGge-C[OF vge-pow[OF p v]] vpGge-pX]* **by** *simp*
have *b: vpGge p [Cp-tr p M:] 0* **by** (*rule vpGge-C[OF vge-mono[OF Cp-tr-vge[OF p p7]]]*) *simp*
show *?thesis* **unfolding** *Yp-tr-def* **by** (*rule vpGge-add[OF p a b]*)
qed

lemma *coeff-high-vge:*

assumes *prime p vpGge p N 0 FarGood p F degree N < d*
shows *vge p (Polynomial.coeff (N * F) d) 1*
proof –
have *N0: $\bigwedge i. vge p (Polynomial.coeff N i) 0$* **using** *assms(2)* **unfolding** *vpGge-iff*
by *blast*
have *F1: $\bigwedge i. 1 \leq i \implies vge p (Polynomial.coeff F i) 1$* **using** *assms(3)*
unfolding *FarGood-def* **by** *blast*
have *vge p ($\sum i \leq d. Polynomial.coeff N i * Polynomial.coeff F (d - i)$) 1*
proof (*rule vge-sum[OF assms(1)]*)
show *finite {..d}* **by** *simp*
next
fix *i* **assume** *i $\in \{..d\}$*
show *vge p (Polynomial.coeff N i * Polynomial.coeff F (d - i)) 1*
proof (*cases degree N < i*)
case *True*
thus *?thesis* **by** (*simp add: coeff-eq-0 vge-zero*)
next
case *False*
hence *1 $\leq d - i$* **using** *assms(4)* **by** *linarith*
from *vge-mul[OF assms(1) N0 F1[OF this]]* **show** *?thesis* **by** *simp*
qed
qed
thus *?thesis* **unfolding** *coeff-mult* .
qed

One summand of (3.7). If $P(c + pz) = p^e N F$ (*Decomp*) with $e \leq p + 1$, the class-*c* summand has valuation $\geq e - \#\{\text{near poles}\}$, for every truncation length $M \geq \deg P$.

theorem *summand-bound-tr:*

assumes *p: prime p and p7: $\gamma \leq p$ and K1: $2 * K < p^2$ and K2: $K + p < p^2$*
and *S: finite S and hS: $\bigwedge r. r \in S \implies$*
 $|r| \leq \text{int } K$ **and** *c: $c < p$*
and *D: Decomp p e e (pcompose P ([of-nat c:] + [of-nat p:] * pX)) and e: e*

$\leq p + 1$
and dP : $\text{degree } P \leq M$
shows $vpGge\ p\ (\tauExt\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (\text{distribNum}\ p\ S\ P\ c)\ (\text{distribPoles}\ p\ S\ c)\ M)$
 $(\text{of-int}\ (\text{int}\ e - \text{int}\ (\text{card}\ \{r \in S.\ \text{int}\ p\ \text{dvd}\ (r - \text{int}\ c)\})))$
proof –
define L **where** $L = [:of\text{-}nat\ c:] + [:of\text{-}nat\ p:] * (pX :: \text{rat}\ poly)$
obtain $N\ F$ **where** Q : $pcompose\ P\ L = [:of\text{-}nat\ p\ \wedge e:] * N * F$ **and** $N0$: $vpGge\ p\ N\ 0$
and Nd : $\text{degree}\ N \leq e$ **and** FG : $\text{FarGood}\ p\ F$
using D **unfolding** $Decomp\text{-}def\ L\text{-}def$ **by** $blast$
have $p0$: $(\text{of-nat}\ p :: \text{rat}) \neq 0$ **using** p **by** $(simp\ add:\ \text{prime-gt-0-nat})$
define S_n **where** $S_n = \{r \in S.\ \text{int}\ p\ \text{dvd}\ (r - \text{int}\ c)\}$
define S_f **where** $S_f = \{r \in S.\ \neg\ \text{int}\ p\ \text{dvd}\ (r - \text{int}\ c)\}$
define $near :: \text{int}\ \text{set}$ **where** $near = (\lambda r.\ (r - \text{int}\ c)\ \text{div}\ \text{int}\ p)\ 'S_n$
define $far :: \text{rat}\ \text{set}$ **where** $far = (\lambda r.\ (\text{of-int}\ r - \text{of-nat}\ c) / \text{of-nat}\ p)\ 'S_f$
have $finS_n$: $\text{finite}\ S_n$ **and** $finS_f$: $\text{finite}\ S_f$ **and** $finnear$: $\text{finite}\ near$ **and** $finfar$: $\text{finite}\ far$
by $(simp\text{-}all\ add:\ S_n\text{-}def\ S_f\text{-}def\ near\text{-}def\ far\text{-}def\ S)$
have inj : $inj\text{-}on\ (\lambda r.\ (\text{of-int}\ r - \text{of-nat}\ c) / (\text{of-nat}\ p :: \text{rat}))\ S$
by $(rule\ inj\text{-}on\ subset[OF\ distrib\text{-}inj\ subset\ UNIV])\ (simp\ add:\ \text{prime-gt-0-nat}\ p)$
have $cardfar$: $\text{card}\ far = \text{card}\ S_f$
unfolding $far\text{-}def$ **by** $(rule\ card\text{-}image)\ (use\ inj\ in\ \langle auto\ simp:\ S_f\text{-}def\ inj\text{-}on\text{-}def \rangle)$
have $cardS$: $\text{card}\ S = \text{card}\ S_n + \text{card}\ S_f$
proof –
have SU : $S = S_n \cup S_f$ **by** $(auto\ simp:\ S_n\text{-}def\ S_f\text{-}def)$
show $?thesis$ **unfolding** SU **by** $(rule\ card\text{-}Un\text{-}disjoint[OF\ finS_n\ finS_f])\ (auto\ simp:\ S_n\text{-}def\ S_f\text{-}def)$
qed
have $poles$: $\text{distribPoles}\ p\ S\ c = \text{of-int}\ 'near \cup far$
proof –
have SU : $S = S_n \cup S_f$ **by** $(auto\ simp:\ S_n\text{-}def\ S_f\text{-}def)$
have $e1$: $(\lambda r.\ (\text{of-int}\ r - \text{of-nat}\ c) / (\text{of-nat}\ p :: \text{rat}))\ 'S_n = \text{of-int}\ 'near$
unfolding $near\text{-}def\ image\text{-}image$
proof $(rule\ image\text{-}cong[OF\ refl])$
fix r **assume** $r \in S_n$
hence $\text{int}\ p\ \text{dvd}\ (r - \text{int}\ c)$ **by** $(simp\ add:\ S_n\text{-}def)$
then obtain q **where** q : $r - \text{int}\ c = \text{int}\ p * q$ **by** $(elim\ dvdE)$
have qd : $(r - \text{int}\ c)\ \text{div}\ \text{int}\ p = q$ **unfolding** q **using** p **by** $(simp\ add:\ \text{prime-gt-0-nat})$
have $(\text{of-int}\ r - \text{of-nat}\ c) / (\text{of-nat}\ p :: \text{rat}) = \text{of-int}\ (r - \text{int}\ c) / \text{of-nat}\ p$
by $simp$
also have $\dots = \text{of-int}\ q$ **unfolding** q **using** $p0$ **by** $simp$
finally show $(\text{of-int}\ r - \text{of-nat}\ c) / \text{of-nat}\ p = (\text{of-int}\ ((r - \text{int}\ c)\ \text{div}\ \text{int}\ p) :: \text{rat})$
unfolding qd .
qed
show $?thesis$ **unfolding** $distribPoles\text{-}def\ SU\ image\text{-}Un\ e1\ far\text{-}def$..

```

qed
have hT:  $\bigwedge r s. r \in \text{near} \implies s \in \text{near} \implies$ 
   $r \neq s \implies \neg \text{int } p \text{ dvd } (r - s)$ 
proof -
  fix r s assume r:  $r \in \text{near}$  and s:  $s \in \text{near}$  and rs:  $r \neq s$ 
  obtain r0 where r0:  $r0 \in S$   $r = (r0 - \text{int } c) \text{ div int } p$  using r unfolding
near-def
  by blast
  obtain s0 where s0:  $s0 \in S$   $s = (s0 - \text{int } c) \text{ div int } p$  using s unfolding
near-def
  by blast
  have r0S:  $r0 \in S$  and s0S:  $s0 \in S$  using r0(1) s0(1) by (simp-all add:
Sn-def)
  have int p dvd (r0 - int c) using r0(1) by (simp add: Sn-def)
  hence dr:  $r0 - \text{int } c = \text{int } p * r$  unfolding r0(2) by simp
  have int p dvd (s0 - int c) using s0(1) by (simp add: Sn-def)
  hence ds:  $s0 - \text{int } c = \text{int } p * s$  unfolding s0(2) by simp
  have int (2 * K) < int (p ^ 2) using K1 by (simp only: of-nat-less-iff)
  hence K1':  $2 * \text{int } K < \text{int } p ^ 2$  by simp
  show  $\neg \text{int } p \text{ dvd } (r - s)$ 
proof
  assume int p dvd (r - s)
  then obtain t where t:  $r - s = \text{int } p * t$  by (elim dvdE)
  have r0 - s0 = (r0 - int c) - (s0 - int c) by simp
  also have ... = int p * (r - s) unfolding dr ds by (simp add: right-diff-distrib)
  also have ... = int p ^ 2 * t unfolding t by (simp add: power2-eq-square)
  finally have e:  $r0 - s0 = \text{int } p ^ 2 * t$  .
  have dv:  $\text{int } p ^ 2 \text{ dvd } (r0 - s0)$  unfolding e by simp
  have |r0 - s0| ≤ |r0| + |s0| by (rule abs-triangle-ineq4)
  hence bnd:  $|r0 - s0| < \text{int } p ^ 2$  using hS[OF r0S] hS[OF s0S] K1' by
linarith
  have r0 - s0 = 0
  proof (rule ccontr)
    assume r0 - s0 ≠ 0
    from dvd-imp-le-int[OF this dv] bnd show False by simp
  qed
  hence r = s using r0(2) s0(2) by simp
  with rs show False ..
qed
qed
have hd:  $\bigwedge r. r \in \text{near} \implies dIdx r < p$ 
proof -
  fix r assume r:  $r \in \text{near}$ 
  obtain r0 where r0:  $r0 \in S$   $r = (r0 - \text{int } c) \text{ div int } p$  using r unfolding
near-def
  by blast
  have r0S:  $r0 \in S$  using r0(1) by (simp add: Sn-def)
  have int p dvd (r0 - int c) using r0(1) by (simp add: Sn-def)
  hence dr:  $r0 - \text{int } c = \text{int } p * r$  unfolding r0(2) by simp

```



```

have ab:  $|r0| \leq \text{int } K$  by (rule hS[OF r0S])
have int  $(K + p) < \text{int } (p \wedge 2)$  using K2 by (simp only: of-nat-less-iff)
hence K2':  $\text{int } K + \text{int } p < \text{int } p * \text{int } p$  by (simp add: power2-eq-square)
have pp:  $0 < \text{int } p$  using p by (simp add: prime-gt-0-nat)
have ci:  $\text{int } c < \text{int } p$  using c by simp
have up:  $\text{int } p * r < \text{int } p * \text{int } p$  using dr ab K2' ci unfolding abs-le-iff by
linarith
have lo:  $\text{int } p * (- \text{int } p) < \text{int } p * r$  using dr ab K2' ci unfolding abs-le-iff
by linarith
have  $r < \text{int } p$  using up pp by (simp add: mult-less-cancel-left)
moreover have  $- \text{int } p < r$ 
proof (rule ccontr)
  assume  $\neg - \text{int } p < r$ 
  hence  $r \leq - \text{int } p$  by simp
  hence  $\text{int } p * r \leq \text{int } p * (- \text{int } p)$  using pp by (intro mult-left-mono)
simp-all
  thus False using lo by linarith
qed
ultimately show  $dIdx\ r < p$  unfolding dIdx-def by (simp add: nat-less-iff)
qed
have hFv:  $\bigwedge s. s \in \text{far} \implies vp\ p\ s = -\ 1$ 
proof -
  fix s assume  $s \in \text{far}$ 
  then obtain r0 where  $r0 \in Sf\ s = (\text{of-int } r0 - \text{of-nat } c) / \text{of-nat } p$ 
  unfolding far-def by blast
  have nd:  $\neg \text{int } p \text{ dvd } (r0 - \text{int } c)$  using r0(1) by (simp add: Sf-def)
  have  $s = \text{of-int } (r0 - \text{int } c) / \text{of-nat } p$  unfolding r0(2) by simp
  thus  $vp\ p\ s = -\ 1$  using vp-int-div-p[OF p nd] by simp
qed
have U0:  $vpGge\ p\ (N * F)\ 0$  using FG
  by (intro vpGge-mul[OF p N0, of - 0, simplified]) (simp add: FarGood-def)
have U1:  $\bigwedge d. p + 1 < d \implies vge\ p\ (\text{Polynomial.coeff } (N * F)\ d)\ 1$ 
  by (rule coeff-high-vge[OF p N0 FG]) (use Nd e in simp)
have dU:  $\text{degree } (N * F) \leq M$ 
proof -
  have Le:  $L = [\text{of-nat } c, \text{of-nat } p:]$  unfolding L-def by (simp add: monom-Suc)
  have  $\text{degree } L = 1$  unfolding Le using p0 by simp
  hence  $\text{degree } (pcompose\ P\ L) = \text{degree } P$  by (simp add: degree-pcompose)
  moreover have  $\text{degree } (pcompose\ P\ L) = \text{degree } (N * F)$  unfolding Q using
p0
  by (simp add: mult.assoc)
  ultimately show ?thesis using dP by simp
qed
have num:  $\text{distribNum } p\ S\ P\ c = [\text{of-nat } p\ \text{powi } (\text{int } e - \text{int } (\text{card } Sn)) * (-\ 1) \\ \wedge \text{card } Sf:] * \\ ([(- \text{of-nat } p)\ \text{powi } (- \text{int } (\text{card } far)):] * (N * F))$ 
proof -
  have pp:  $(\text{of-nat } p :: \text{rat})\ \text{powi } (\text{int } e - \text{int } (\text{card } Sn)) * \text{of-nat } p\ \text{powi } (- \text{int } (\text{card } Sf)) =$ 

```

```

      of-nat p powi (− int (card S)) * of-nat p ^ e
    proof −
      have (of-nat p :: rat) powi (int e − int (card Sn)) * of-nat p powi (− int
(card Sf)) =
        of-nat p powi (int e − int (card Sn) + − int (card Sf))
      by (rule power-int-add[symmetric]) (use p0 in simp)
      also have ... = of-nat p powi (− int (card S) + int e)
      by (rule arg-cong[where f = λk. (of-nat p :: rat) powi k]) (use cardS in
simp)
      also have ... = of-nat p powi (− int (card S)) * of-nat p powi (int e)
      by (rule power-int-add) (use p0 in simp)
      also have ... = of-nat p powi (− int (card S)) * of-nat p ^ e
      by (simp add: power-int-of-nat)
      finally show ?thesis .
    qed
    have mm: ((− 1 :: rat) ^ card Sf) * (− 1) ^ card Sf = 1
      by (simp add: power-mult-distrib[symmetric])
    have ng: (− (of-nat p :: rat)) powi (− int (card Sf)) =
      (− 1) ^ card Sf * of-nat p powi (− int (card Sf))
    proof −
      have i1: inverse ((− 1 :: rat) ^ card Sf) = (− 1) ^ card Sf
      by (cases even (card Sf)) (simp-all add: neg-one-even-power neg-one-odd-power)
      show ?thesis unfolding power-int-minus power-int-of-nat power-minus[of
of-nat p :: rat]
        inverse-mult-distrib i1 ..
    qed
    have k: of-nat p powi (− int (card S)) * of-nat p ^ e =
      (of-nat p powi (int e − int (card Sn)) * (− 1) ^ card Sf) *
      ((− of-nat p) powi (− int (card far)) :: rat)
    proof −
      have (of-nat p powi (int e − int (card Sn)) * (− 1) ^ card Sf) *
      ((− of-nat p) powi (− int (card far)) :: rat) =
      ((− 1) ^ card Sf * (− 1) ^ card Sf) * (of-nat p powi (int e − int (card
Sn)) *
        of-nat p powi (− int (card Sf)))
      unfolding cardfar ng by (simp only: mult-ac)
      also have ... = of-nat p powi (− int (card S)) * of-nat p ^ e unfolding mm
pp by simp
      finally show ?thesis by (rule sym)
    qed
    have g1: [:x:] * ([:y:] * N * F) = Polynomial.smult (x * y) (N * F) for x y ::
rat
      by (simp add: mult-ac)
    have g2: [:x:] * ([:y:] * (N * F)) = Polynomial.smult (x * y) (N * F) for x y
:: rat
      by (simp add: mult commute)
    show ?thesis unfolding distribNum-def L-def[symmetric] Q g1 g2 k ..
  qed
  have ext: vpGge p (tauExt-tr p (Yp-tr p M) ([:(− of-nat p) powi (− int (card

```

```

far)):] * (N *
  F)) (of-int 'near  $\cup$  far) M) 0
  by (rule tauX-far-pole-valuation[OF p p7 finnear finfar hT hd hFv vpGge-Yp-tr[OF
p p7] U0 U1
    dU])
  have cv: vge p (of-nat p powi (int e - int (card Sn)) * (- 1) ^ card Sf) (of-int
(int e -
  int (card Sn)))
  proof -
    have v1: vge p (of-nat p powi (int e - int (card Sn))) (of-int (int e - int (card
Sn)))
    by (simp add: vge-def vp-def qmultiplicity-power-int[OF p])
    have v2: vge p ((- 1) ^ card Sf) 0 using vge-of-int[OF p, of (- 1) ^ card Sf]
  by simp
  from vge-mul[OF p v1 v2] show ?thesis by simp
qed
show ?thesis
  unfolding poles num tauExt-tr-C-mul[where a =
    of-nat p powi (int e - int (card Sn)) * (- 1) ^ card Sf] Sn-def[symmetric]
  using vpGge-mul[OF p vpGge-C[OF cv] ext] by simp
qed

```

lemma *finite-signedPoles*: $\text{finite } S \implies \text{finite } (\text{signedPoles } S)$
by (simp add: signedPoles-def)

lemma *signedPoles-abs-le*: $r \in \text{signedPoles } \{1..K\} \implies |r| \leq \text{int } K$
by (auto simp: signedPoles-def)

lemma *nearCount-eq-card*: $\text{nearCount } p \ K \ c = \text{card } \{r \in \text{signedPoles } \{1..K\}. \text{int } p \text{ dvd } (r - \text{int } c)\}$
by (simp add: nearCount-def)

The inner-range entry bound, generic form (Lean: *entry-generic*): if every class c gives $B + 4 \leq e-c - n-c$ with $e-c \leq p + 1$, then $v-p \wedge G(\mu X (\prod (t + j \wedge 2) \wedge \{k-j\} / D-K)) \geq B$. Uses (3.1), Lemma 3.2 and the summand bounds.

theorem *entry-generic*:

```

assumes p: prime p and p7: 7 ≤ p and K1: 2 * K < p ^ 2 and K2: K + p <
  p ^ 2 and J: finite J
  and hB:  $\bigwedge c. c < p \implies$ 
    B + 4 ≤ of-nat (eNum p c J k) - of-nat (nearCount p K c)
  and hdeg:  $\bigwedge c. c < p \implies \text{eNum } p \ c \ J \ k \leq p + 1$ 
shows vpGge p (muX {1..K} ( $\prod j \in J. (pX + [: \text{of-nat } j \wedge 2:] ) \wedge k \ j$ )) B
proof -
  define S where S = signedPoles {1..K}
  define P where P =  $[: (- 1) \wedge \text{card } \{1..K\}:] * pX \wedge 5 * pcompose (\prod j \in$ 
    J.  $(pX + [: \text{of-nat } j \wedge 2:] ) \wedge k \ j) (- (pX \wedge 2))$ 
  have finS: finite S by (simp add: S-def finite-signedPoles)
  have pb:  $\mu X \{1..K\} (\prod j \in J. (pX + [: \text{of-nat } j \wedge 2:] ) \wedge k \ j) = \text{tauX } S \ P$ 
  unfolding S-def P-def by (rule pullback) auto

```

```

have  $\varepsilon$ :  $vge\ p\ ((-1) \wedge card\ \{1..K\})\ 0$ 
using  $vge\text{-}of\text{-}int[OF\ p,\ of\ (-1) \wedge card\ \{1..K\}]$  by simp
obtain  $M0$  where  $M0$ :  $\bigwedge M. M0 \leq M \implies$ 
 $vpGge\ p\ (tauX\ S\ P - [:of\text{-}nat\ p\ powi\ (-4):] *$ 
 $(\sum a < p. tauExt\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ a)\ (distribPoles$ 
 $p\ S\ a)\ M))$ 
 $B$ 
using  $tauX\text{-}distribution\text{-}formula[OF\ p\ p7\ finS,\ of\ P\ B]$  by blast
define  $M$  where  $M = max\ M0\ (degree\ P)$ 
have  $summand$ :  $\bigwedge c. c < p \implies$ 
 $vpGge\ p\ (tauExt\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ c)\ (distribPoles\ p\ S$ 
 $M)\ (B + 4))$ 
proof –
fix  $c$  assume  $c$ :  $c < p$ 
have  $vpGge\ p\ (tauExt\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ c)\ (distribPoles\ p\ S$ 
 $c)\ M)$ 
 $(of\text{-}int\ (int\ (eNum\ p\ c\ J\ k) - int\ (card\ \{r \in S. int\ p\ dvd\ (r - int\ c)\})))$ 
unfolding  $P\text{-}def$ 
by  $(rule\ summand\text{-}bound\text{-}tr[OF\ p\ p7\ K1\ K2\ finS - c\ decomp\text{-}pullback[OF\ p\ J$ 
 $\varepsilon]\ hdeg[OF$ 
 $c]])$ 
 $(use\ signedPoles\text{-}abs\text{-}le\ \mathbf{in}\ \langle auto\ simp: S\text{-}def\ M\text{-}def\ P\text{-}def \rangle)$ 
thus  $vpGge\ p\ (tauExt\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ c)\ (distribPoles\ p\ S$ 
 $c)\ M)\ (B + 4)$ 
proof  $(rule\ vpGge\text{-}mono)$ 
show  $B + 4 \leq of\text{-}int\ (int\ (eNum\ p\ c\ J\ k) - int\ (card\ \{r \in S. int\ p\ dvd\ (r$ 
 $- int\ c)\})))$ 
using  $hB[OF\ c]$  unfolding  $nearCount\text{-}eq\text{-}card\ S\text{-}def$  by simp
qed
qed
have  $sumv$ :  $vpGge\ p\ ([:of\text{-}nat\ p\ powi\ (-4):] *$ 
 $(\sum a < p. tauExt\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ a)\ (distribPoles\ p\ S\ a)$ 
 $M))\ B$ 
proof –
have  $c4$ :  $vpGge\ p\ [:of\text{-}nat\ p\ powi\ (-4):]\ (-4)$ 
by  $(rule\ vpGge\text{-}C)\ (simp\ add: vge\text{-}def\ vp\text{-}def\ qmultiplicity\text{-}power\text{-}int[OF\ p])$ 
have  $vpGge\ p\ (\sum a < p. tauExt\text{-}tr\ p\ (Yp\text{-}tr\ p\ M)\ (distribNum\ p\ S\ P\ a)$ 
 $(distribPoles\ p\ S\ a)$ 
 $M)\ (B + 4)$ 
by  $(rule\ vpGge\text{-}sum[OF\ p])\ (use\ summand\ \mathbf{in}\ auto)$ 
from  $vpGge\text{-}mul[OF\ p\ c4\ this]$  show  $?thesis$  by simp
qed
have  $vpGge\ p\ (tauX\ S\ P)\ B$ 
using  $vpGge\text{-}add[OF\ p\ M0[of\ M]\ sumv]$  by  $(simp\ add: M\text{-}def)$ 
thus  $?thesis$  by  $(simp\ only: pb)$ 
qed

```

21.3 The inner entry bounds (4.2)/(4.3)

The class of a residue: $cls\ p\ j = \min(j \bmod p, p - j \bmod p)$.

definition $cls :: nat \Rightarrow nat \Rightarrow nat$ **where** $cls\ p\ j = \min\ (j \bmod p)\ (p - j \bmod p)$

Exponent of $t + j^2$ in the inner row E - x .

definition $nu :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \times nat \Rightarrow nat \Rightarrow nat$ **where**
 $nu\ K\ M\ p\ x\ j = (\text{if } j \leq mHalf\ p \text{ then } (\text{if } j = fst\ x \text{ then } snd\ x \text{ else } nat\ (Lin\ K\ M\ p\ j)) \text{ else } 0)$

The index set and the exponents of the numerator D - N^6 E - x E - y .

definition $Jset :: nat \Rightarrow nat \Rightarrow nat\ set$ **where** $Jset\ K\ p = \{1..Nof\ K\} \cup \{..mHalf\ p\}$

definition $kexp :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \times nat \Rightarrow nat \times nat \Rightarrow nat \Rightarrow nat$ **where**
 $kexp\ K\ M\ p\ x\ y\ j = 6 * (\text{if } j \in \{1..Nof\ K\} \text{ then } 1 \text{ else } 0) + nu\ K\ M\ p\ x\ j + nu\ K\ M\ p\ y\ j$

Multiplicity of a class: 2 for the zero class, 1 otherwise.

definition $mult :: nat \Rightarrow nat$ **where** $mult\ c = (\text{if } c = 0 \text{ then } 2 \text{ else } 1)$

lemma *mem-inner-idx*:

$x \in set\ (inner_idx\ K\ M\ p) \longleftrightarrow fst\ x \leq mHalf\ p \wedge snd\ x < nat\ (Lin\ K\ M\ p\ (fst\ x))$

by (cases x) (auto simp: inner-idx-def image-iff simp del: upt-Suc)

lemma *innerBasis-eq-prod*:

assumes $fst\ x \leq mHalf\ p$

shows $innerBasis\ K\ M\ p\ x = (\prod_{j \leq mHalf\ p} (pX + [:of-nat\ j\ ^2:])) \wedge nu\ K\ M\ p\ x\ j$

proof –

have $a: fst\ x \in \{..mHalf\ p\}$ **using** *assms* **by** *simp*

have $n: nu\ K\ M\ p\ x\ (fst\ x) = snd\ x$ **using** *assms* **by** (*simp add: nu-def*)

have $c: (\prod_{j \in \{..mHalf\ p\} - \{fst\ x\}} (pX + [:of-nat\ j\ ^2:])) \wedge nu\ K\ M\ p\ x\ j = (\prod_{c \in \{0..mHalf\ p\} - \{fst\ x\}} (pX + [:of-nat\ c\ ^2:])) \wedge nat\ (Lin\ K\ M\ p\ c))$

by (*rule prod.cong*) (*auto simp: nu-def*)

show *?thesis* **unfolding** *innerBasis-def prod.remove[OF finite-atMost a]* $n\ c$ **by** (*rule mult.commute*)

qed

lemma *numerator-eq*:

assumes $fst\ x \leq mHalf\ p$ $fst\ y \leq mHalf\ p$

shows $Dm\ (Nof\ K) \wedge 6 * innerBasis\ K\ M\ p\ x * innerBasis\ K\ M\ p\ y =$

$(\prod_{j \in Jset\ K\ p} (pX + [:of-nat\ j\ ^2:])) \wedge kexp\ K\ M\ p\ x\ y\ j$

proof –

```

have fin: finite (Jset K p) by (simp add: Jset-def)
have e1: ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ (6 * (if j \in \{1..Nof K\} then 1 else 0))) = Dm (Nof K) ^ 6$ )
proof -
  have ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ (6 * (if j \in \{1..Nof K\} then 1 else 0))) =$ 
    ( $\prod j = 1..Nof K. (pX + [:of-nat j ^ 2:]) ^ 6$ )
    by (rule prod.mono-neutral-cong-right[OF fin]) (auto simp: Jset-def)
  also have ... = Dm (Nof K) ^ 6 unfolding Dm-def by (rule prod-power-distrib[symmetric])
  finally show ?thesis .
qed
have e2: ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ nu K M p z j$ ) =
  innerBasis K M p z if fst z ≤ mHalf p for z
proof -
  have ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ nu K M p z j$ ) = ( $\prod j \leq mHalf p. (pX + [:of-nat j ^ 2:]) ^ nu K M p z j$ )
    by (rule prod.mono-neutral-cong-right[OF fin]) (auto simp: Jset-def nu-def)
  also have ... = innerBasis K M p z by (rule innerBasis-eq-prod[OF that, symmetric])
  finally show ?thesis .
qed
have ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ kexp K M p x y j$ ) =
  ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ (6 * (if j \in \{1..Nof K\} then 1 else 0))) *$ 
  ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ nu K M p x j$ ) * ( $\prod j \in Jset K p. (pX + [:of-nat j ^ 2:]) ^ nu K M p y j$ )
  unfolding kexp-def power-add prod.distrib ..
thus ?thesis unfolding e1 e2[OF assms(1)] e2[OF assms(2)] by (rule sym)
qed

lemma cls-le: odd p  $\implies$  cls p j ≤ mHalf p
proof -
  assume odd p
  then obtain m where pm: p = 2 * m + 1 by (elim oddE)
  have j mod p < p using pm by simp
  thus ?thesis unfolding cls-def mHalf-def pm by (auto simp: min-def)
qed

lemma cls-small: odd p  $\implies$  j ≤ mHalf p  $\implies$  cls p j = j
proof -
  assume odd p and j: j ≤ mHalf p
  then obtain m where pm: p = 2 * m + 1 by (elim oddE)
  have j ≤ m using j unfolding mHalf-def pm by simp
  thus ?thesis unfolding cls-def pm by (simp add: min-def)
qed

lemma mod-add-zero-iff-aux:
  fixes r c p :: nat
  assumes r < p c < p

```

```

    shows  $(r + c) \bmod p = 0 \iff r + c = 0 \vee r + c = p$ 
  proof (cases  $r + c < p$ )
    case True
      thus ?thesis by simp
    next
      case False
        hence  $(r + c) \bmod p = (r + c - p) \bmod p$  by (simp add: le-mod-geq)
        also have  $\dots = r + c - p$  using assms by simp
        finally show ?thesis using False by auto
  qed

lemma cls-eq-iff:
  assumes odd  $p$   $a \leq mHalf\ p$ 
  shows  $cls\ p\ j = a \iff j \bmod p = a \vee j \bmod p + a = p$ 
proof -
  obtain  $m$  where  $pm: p = 2 * m + 1$  using assms(1) by (elim oddE)
  have  $j \bmod p < p$  using pm by simp
  moreover have  $a \leq m$  using assms(2) unfolding mHalf-def pm by simp
  ultimately show ?thesis unfolding cls-def pm by (auto simp: min-def)
qed

lemma omg-eq:
  assumes odd  $p$   $c < p$ 
  shows  $omg\ p\ c\ j = (if\ cls\ p\ j = cls\ p\ c\ then\ mult\ c\ else\ 0)$ 
proof -
  have  $p0: 0 < p$  using assms(2) by simp
  define  $r$  where  $r = j \bmod p$ 
  have  $r: r < p$  using  $p0$  by (simp add: r-def)
  have  $d1: int\ p\ dvd\ (int\ j - int\ c) \iff r = c$ 
  proof -
    have  $int\ p\ dvd\ (int\ j - int\ c) \iff int\ j \bmod int\ p = int\ c \bmod int\ p$ 
      by (simp add: mod-eq-dvd-iff)
    also have  $\dots \iff j \bmod p = c \bmod p$ 
      by (simp only: of-nat-mod[symmetric] of-nat-eq-iff)
    finally show ?thesis using assms(2) by (simp add: r-def)
  qed
  have  $d2: int\ p\ dvd\ (int\ j + int\ c) \iff r + c = 0 \vee r + c = p$ 
  proof -
    have  $int\ p\ dvd\ (int\ j + int\ c) \iff p\ dvd\ (j + c)$ 
      by (metis int-dvd-int-iff of-nat-add)
    also have  $\dots \iff (r + c) \bmod p = 0$ 
      by (simp add: r-def dvd-eq-mod-eq-0 mod-add-left-eq)
    also have  $\dots \iff r + c = 0 \vee r + c = p$ 
      by (rule mod-add-zero-iff-aux[OF r assms(2)])
    finally show ?thesis .
  qed
  have  $np: c + c \neq p$  using assms(1) by presburger
  have  $cj: cls\ p\ j = min\ r\ (p - r)$  by (simp add: cls-def r-def)
  have  $cc: cls\ p\ c = min\ c\ (p - c)$  using assms(2) by (simp add: cls-def)

```

show *?thesis* **unfolding** *omg-def d1 d2 cj cc mult-def* **using** *r assms(2) np* **by**
(auto simp: min-def)

qed

lemma *sum-omg*:

assumes *odd p c < p*

shows $(\sum j = 1..A. \text{omg } p \ c \ j) = \text{mult } c * \text{ell } p \ A \ (\text{cls } p \ c)$

proof –

have *p0: 0 < p* **using** *assms(2)* **by** *simp*

define *a* **where** *a = cls p c*

have *am: a ≤ mHalf p* **unfolding** *a-def* **by** *(rule cls-le[OF assms(1)])*

have *ap: a < p* **using** *am p0* **by** *(simp add: mHalf-def)*

have *m2: (j + a) mod p = 0 ⟷ j mod p + a = 0 ∨ j mod p + a = p* **for** *j*
using *mod-add-zero-iff-aux[OF - ap, of j mod p] p0* **by** *(simp add: mod-add-left-eq)*

have *S: {j ∈ {1..A}. cls p j = a} = {j ∈ {1..A}. j mod p = a mod p ∨
(j + a) mod p = 0}*

using *cls-eq-iff[OF assms(1) am] m2 ap* **by** *auto*

have $(\sum j = 1..A. \text{omg } p \ c \ j) = (\sum j \in \{j \in \{1..A\}. \text{cls } p \ j = a\}. \text{mult } c)$

unfolding *omg-eq[OF assms] a-def sum.inter-filter[OF finite-atLeastAtMost]* **..**

also have $\dots = \text{mult } c * \text{card } \{j \in \{1..A\}. \text{cls } p \ j = a\}$ **by** *simp*

also have $\dots = \text{mult } c * \text{ell } p \ A \ (\text{cls } p \ c)$ **unfolding** *S* **unfolding** *ell-def a-def*

..

finally show *?thesis* **.**

qed

lemma *nearCount-eq*:

assumes *prime p*

shows $\text{nearCount } p \ K \ c = (\sum j = 1..K. \text{omg } p \ c \ j)$

proof –

have *n: int p dvd (– int j – int c) ⟷ int p dvd (int j + int c)* **for** *j*

proof –

have *eq: – int j – int c = – (int j + int c)* **by** *simp*

show *?thesis* **unfolding** *eq dvd-minus-iff* **..**

qed

have *U: {r ∈ int ‘ {1..K} ∪ (λj. – int j) ‘ {1..K}. int p dvd (r – int c)} =
int ‘ {j ∈ {1..K}. int p dvd (int j – int c)} ∪ (λj. – int j) ‘ {j ∈
{1..K}. int p dvd (int j + int c)}*

using *n* **by** *auto*

have *c1: card (int ‘ {j ∈ {1..K}. P j}) = card {j ∈ {1..K}. P j}* **for** *P*

by *(rule card-image) (simp add: inj-on-def)*

have *c2: card ((λj. – int j) ‘ {j ∈ {1..K}. P j}) = card {j ∈ {1..K}. P j}* **for**
P

by *(rule card-image) (simp add: inj-on-def)*

have *ind: (∑ j ∈ {1..K}. (if P j then 1 else 0 :: nat)) = card {j ∈ {1..K}. P j}*
for *P*

by *(simp add: sum.inter-filter[OF finite-atLeastAtMost, symmetric])*

have $\text{nearCount } p \ K \ c = \text{card } (\text{int ‘ } \{j \in \{1..K\}. \text{int } p \ \text{dvd } (\text{int } j - \text{int } c)\} \cup$
 $(\lambda j. - \text{int } j) ‘ \{j \in \{1..K\}. \text{int } p \ \text{dvd } (\text{int } j + \text{int } c)\})$

unfolding *nearCount-def signedPoles-def U* **..**

also have $\dots = \text{card } (\text{int } \{j \in \{1..K\}. \text{int } p \text{ dvd } (\text{int } j - \text{int } c)\}) +$
 $\text{card } ((\lambda j. - \text{int } j) \{j \in \{1..K\}. \text{int } p \text{ dvd } (\text{int } j + \text{int } c)\})$
by (rule card-Un-disjoint; force+)
also have $\dots = \text{card } \{j \in \{1..K\}. \text{int } p \text{ dvd } (\text{int } j - \text{int } c)\} + \text{card } \{j \in$
 $\{1..K\}. \text{int } p \text{ dvd } (\text{int } j + \text{int } c)\}$
unfolding c1 c2 ..
also have $\dots = (\sum j = 1..K. \text{omg } p \text{ c } j)$ **unfolding** omg-def sum.distrib ind ..
finally show ?thesis .
qed

lemma cls-zero: $\text{cls } p \ 0 = 0$
by (simp add: cls-def)

lemma cls-pos: $c \neq 0 \implies c < p \implies 1 \leq \text{cls } p \ c$
by (simp add: cls-def)

lemma ell-zero-eq: $0 < p \implies \text{ell } p \ A \ 0 = A \text{ div } p$
proof –
assume $p: 0 < p$
have $e: \{j \in \{1..A\}. j \bmod p = 0 \bmod p \vee (j + 0) \bmod p = 0\} = \{j \in \{1..A\}.$
 $p \text{ dvd } j\}$
by (auto simp: dvd-eq-mod-eq-0)
show ?thesis **unfolding** ell-def e **by** (rule card-multiples-upto[OF p])
qed

lemma eNum-eq:
assumes $\text{odd } p \ c < p$
shows $e\text{Num } p \ c \ (Jset \ K \ p) \ (kexp \ K \ M \ p \ x \ y) =$
 $5 * (\text{if } c = 0 \text{ then } 1 \text{ else } 0) +$
 $\text{mult } c * (6 * \text{ell } p \ (\text{Nof } K) \ (\text{cls } p \ c) + \text{nu } K \ M \ p \ x \ (\text{cls } p \ c) + \text{nu } K \ M \ p$
 $y \ (\text{cls } p \ c))$
proof –
have $\text{fin}: \text{finite } (Jset \ K \ p)$ **by** (simp add: Jset-def)
have $\text{dc}: \text{int } p \text{ dvd } \text{int } c \longleftrightarrow c = 0$ **using** assms(2) **by** (auto dest: dvd-imp-le)
have $s1: (\sum j \in Jset \ K \ p. (\text{if } j \in \{1..\text{Nof } K\} \text{ then } 1 \text{ else } 0) * \text{omg } p \ c \ j) =$
 $\text{mult } c * \text{ell } p \ (\text{Nof } K) \ (\text{cls } p \ c)$
proof –
have $(\sum j \in Jset \ K \ p. (\text{if } j \in \{1..\text{Nof } K\} \text{ then } 1 \text{ else } 0) * \text{omg } p \ c \ j) = (\sum j$
 $=$
 $1..\text{Nof } K. \text{omg } p \ c \ j)$
by (rule sum.mono-neutral-cong-right[OF fin]) (auto simp: Jset-def)
also have $\dots = \text{mult } c * \text{ell } p \ (\text{Nof } K) \ (\text{cls } p \ c)$ **by** (rule sum-omg[OF assms])
finally show ?thesis .
qed
have $s2: (\sum j \in Jset \ K \ p. \text{nu } K \ M \ p \ z \ j * \text{omg } p \ c \ j) = \text{nu } K \ M \ p \ z \ (\text{cls } p \ c) *$
 $\text{mult } c$
for z
proof –
have $(\sum j \in Jset \ K \ p. \text{nu } K \ M \ p \ z \ j * \text{omg } p \ c \ j) = (\sum j \leq m\text{Half } p.$

if $j = \text{cls } p \ c$ then $\text{nu } K \ M \ p \ z \ j * \text{mult } c$ else 0)
 by (rule *sum.mono-neutral-cong-right*[OF *fin*]) (auto simp: *Jset-def nu-def*
omg-eq[OF *assms*]
cls-small[OF *assms*(1)])
 also have ... = $\text{nu } K \ M \ p \ z \ (\text{cls } p \ c) * \text{mult } c$ using *cls-le*[OF *assms*(1), of
c] by *simp*
 finally show ?thesis .
 qed
 have $e\text{Num } p \ c \ (\text{Jset } K \ p) \ (\text{kexp } K \ M \ p \ x \ y) = 5 * (\text{if } \text{int } p \ \text{dvd } \text{int } c \text{ then } 1 \text{ else } 0) +$
 $(6 * (\sum j \in \text{Jset } K \ p. (\text{if } j \in \{1.. \text{Nof } K\} \text{ then } 1 \text{ else } 0) * \text{omg } p \ c \ j) +$
 $(\sum j \in \text{Jset } K \ p. \text{nu } K \ M \ p \ x \ j * \text{omg } p \ c \ j) + (\sum j \in$
 $\text{Jset } K \ p. \text{nu } K \ M \ p \ y \ j * \text{omg } p \ c \ j))$
 unfolding *eNum-def kexp-def* by (simp add: *sum.distrib sum-distrib-left alge-*
bra-simps)
 also have ... = $5 * (\text{if } c = 0 \text{ then } 1 \text{ else } 0) +$
 $\text{mult } c * (6 * \text{ell } p \ (\text{Nof } K) \ (\text{cls } p \ c) + \text{nu } K \ M \ p \ x \ (\text{cls } p \ c) + \text{nu } K \ M \ p \ y$
 $(\text{cls } p \ c))$
 unfolding *s1 s2 dc* by (simp add: *algebra-simps*)
 finally show ?thesis .
 qed

lemma *count-mod-ge*: $0 < p \implies 1 \leq r \implies r < p \implies$
 $A \ \text{div } p \leq \text{card } \{j \in \{1..A\}. j \bmod p = r\}$
proof –
 assume *p*: $0 < p$ and *r1*: $1 \leq r$ and *rp*: $r < p$
 have *sub*: $(\lambda t. t * p + r) ' \{.. < A \ \text{div } p\} \subseteq \{j \in \{1..A\}. j \bmod p = r\}$
proof
 fix *j* assume *j* $\in (\lambda t. t * p + r) ' \{.. < A \ \text{div } p\}$
 then obtain *t* where *t*: $t < A \ \text{div } p \ j = t * p + r$ by *blast*
 have $\text{Suc } t * p \leq A \ \text{div } p * p$ using *t*(1) by (intro *mult-le-mono1*) *simp*
 also have ... $\leq A$ by (rule *div-times-less-eq-dividend*)
 finally have $t * p + p \leq A$ by *simp*
 thus $j \in \{j \in \{1..A\}. j \bmod p = r\}$ using *t r1 rp* by *simp*
 qed
 have *inj*: *inj-on* $(\lambda t. t * p + r) \ \{.. < A \ \text{div } p\}$ using *p* by (auto intro: *inj-onI*)
 have $\text{card } ((\lambda t. t * p + r) ' \{.. < A \ \text{div } p\}) \leq \text{card } \{j \in \{1..A\}. j \bmod p = r\}$
 by (rule *card-mono*[OF *sub*]) *simp*
 thus ?thesis using *card-image*[OF *inj*] by *simp*
 qed

lemma *count-mod-le*: $0 < p \implies 1 \leq r \implies r < p \implies$
 $\text{card } \{j \in \{1..A\}. j \bmod p = r\} \leq A \ \text{div } p + 1$
proof –
 assume *p*: $0 < p$
 show ?thesis by (rule *card-cong-class-le*[OF *p*]) *auto*
 qed

lemma *ell-bounds*:

```

assumes  $odd\ p\ 1 \leq a\ a \leq mHalf\ p$ 
shows  $2 * (A\ div\ p) \leq ell\ p\ A\ a \wedge ell\ p\ A\ a \leq 2 * (A\ div\ p) + 2$ 
proof –
  have  $p0: 0 < p$  using  $assms(1)$  by  $(rule\ odd-pos)$ 
  have  $ap: a < p$  using  $assms(3)\ p0$  by  $(simp\ add: mHalf-def)$ 
  have  $np: a \neq p - a$  using  $assms(1)\ ap$  by  $presburger$ 
  have  $m: (j + a) \bmod p = 0 \iff j \bmod p = p - a$  for  $j$ 
  proof –
    have  $r: j \bmod p < p$  using  $p0$  by  $simp$ 
    have  $(j + a) \bmod p = 0 \iff j \bmod p + a = 0 \vee j \bmod p + a = p$ 
      using  $mod-add-zero-iff-aux[OF\ r\ ap]$  by  $(simp\ add: mod-add-left-eq)$ 
    thus  $?thesis$  using  $assms(2)\ ap\ r$  by  $auto$ 
  qed
  have  $e: \{j \in \{1..A\}. j \bmod p = a \bmod p \vee (j + a) \bmod p = 0\} =$ 
     $\{j \in \{1..A\}. j \bmod p = a\} \cup \{j \in \{1..A\}. j \bmod p = p - a\}$ 
    using  $m\ ap$  by  $auto$ 
  have  $ell\ p\ A\ a = card\ \{j \in \{1..A\}. j \bmod p = a\} + card\ \{j \in \{1..A\}. j \bmod p$ 
 $= p - a\}$ 
    unfolding  $ell-def\ e$  by  $(rule\ card-Un-disjoint)\ (use\ np\ in\ auto)$ 
  moreover have  $A\ div\ p \leq card\ \{j \in \{1..A\}. j \bmod p = a\}$ 
    by  $(rule\ count-mod-ge[OF\ p0\ assms(2)\ ap])$ 
  moreover have  $A\ div\ p \leq card\ \{j \in \{1..A\}. j \bmod p = p - a\}$ 
    by  $(rule\ count-mod-ge[OF\ p0])\ (use\ assms(2)\ ap\ in\ auto)$ 
  moreover have  $card\ \{j \in \{1..A\}. j \bmod p = a\} \leq A\ div\ p + 1$ 
    by  $(rule\ count-mod-le[OF\ p0\ assms(2)\ ap])$ 
  moreover have  $card\ \{j \in \{1..A\}. j \bmod p = p - a\} \leq A\ div\ p + 1$ 
    by  $(rule\ count-mod-le[OF\ p0])\ (use\ assms(2)\ ap\ in\ auto)$ 
  ultimately show  $?thesis$  by  $linarith$ 
qed

lemma  $hyp-facts$ :
  assumes  $prime\ p\ InnerHyp\ K\ M\ p$ 
  shows  $7 \leq p \wedge odd\ p \wedge 2 * mHalf\ p + 1 = p \wedge K\ div\ p < M \wedge$ 
 $Nof\ K\ div\ p < M$ 
  proof –
    have  $b: 200 * M < p$  using  $innerHyp-bounds[OF\ assms(2)]$  by  $blast$ 
    have  $H: 40 \leq M\ K < p * M$  using  $assms(2)$  by  $(auto\ simp: InnerHyp-def)$ 
    have  $p7: 7 \leq p$  using  $b\ H$  by  $linarith$ 
    have  $o: odd\ p$  using  $prime-odd-nat[OF\ assms(1)]\ p7$  by  $simp$ 
    have  $mp: 2 * mHalf\ p + 1 = p$  using  $o$  unfolding  $mHalf-def$  by  $(auto\ elim: oddE)$ 
    have  $kd: K\ div\ p < M$  using  $H(2)\ p7$  by  $(simp\ add: div-less-iff-less-mult\ mult.commute)$ 
    have  $Nof\ K\ div\ p \leq K\ div\ p$  by  $(rule\ div-le-mono)\ (simp\ add: Nof-def)$ 
    hence  $Nof\ K\ div\ p < M$  using  $kd$  by  $linarith$ 
    with  $p7\ o\ mp\ kd$  show  $?thesis$  by  $simp$ 
  qed

lemma  $Tin-le$ :

```

```

assumes prime p InnerHyp K M p
shows  $Tin\ K\ M\ p \leq 4 * int\ M$ 
proof -
  have  $T0: 0 \leq Ein\ K\ M\ p$  and  $Em: int\ (mHalf\ p) * Tin\ K\ M\ p + Ein\ K\ M\ p =$ 
 $rhs44\ K\ M\ p$ 
  using TE-spec[OF assms] by blast+
  have  $H: 40 \leq M\ 40\ dvd\ K\ 200 * M^2 \leq K\ K < p * M$  using assms(2)
  by (auto simp: InnerHyp-def)
  obtain  $k$  where  $Kk: K = 40 * k$  using  $H(2)$  by (elim dvdE)
  have  $hk: hof\ K = 37 * k$  and  $Nk: Nof\ K = 3 * k$  unfolding  $Kk\ hof-def\ Nof-def$ 
by simp-all
  have  $r44: rhs44\ K\ M\ p \leq 46 * int\ k$  unfolding  $rhs44-def\ hk\ Nk\ L0-def$  by simp
  have  $mp: 2 * mHalf\ p + 1 = p$  using hyp-facts[OF assms] by blast
  define  $m$  where  $m = int\ (mHalf\ p)$ 
  have  $m0: 0 \leq m$  by (simp add: m-def)
  have  $ip: int\ p = 2 * m + 1$  unfolding  $m-def$  using  $mp$  by linarith
  have  $pM: int\ p * int\ M = 2 * (m * int\ M) + int\ M$  unfolding  $ip$  by (simp
add: algebra-simps)
  have  $int\ K < int\ (p * M)$  using  $H(4)$  by (simp only: of-nat-less-iff)
  hence  $KpM: 40 * int\ k < int\ p * int\ M$  unfolding  $Kk$  by simp
  have  $M \leq M * M$  by (rule le-square)
  hence  $M \leq k$  using  $H(3)\ Kk$  unfolding power2-eq-square by linarith
  hence  $kM: int\ M \leq int\ k$  by simp
  show ?thesis
  proof (rule ccontr)
    assume  $\neg Tin\ K\ M\ p \leq 4 * int\ M$ 
    hence  $4 * int\ M + 1 \leq Tin\ K\ M\ p$  by simp
    hence  $m * (4 * int\ M + 1) \leq m * Tin\ K\ M\ p$  using  $m0$  by (rule mult-left-mono)
    hence  $4 * (m * int\ M) + m \leq m * Tin\ K\ M\ p$  by (simp add: algebra-simps)
    moreover have  $m * Tin\ K\ M\ p \leq 46 * int\ k$  using  $Em\ T0\ r44$  unfolding
m-def by linarith
    ultimately show False using  $pM\ KpM\ kM\ m0$  by linarith
  qed
qed

lemma Lin-le:
  assumes prime p InnerHyp K M p a ≤ mHalf p
  shows  $Lin\ K\ M\ p\ a \leq 4 * int\ M + 10$ 
proof (cases a = 0)
  case True
    thus ?thesis by (simp add: Lin-def L0-def)
  next
  case False
    have  $epsIn\ K\ M\ p\ a \leq 1$  by (simp add: epsIn-def)
    moreover have  $0 \leq bIn\ K\ p\ a$  by (simp add: bIn-def)
    moreover have  $Lin\ K\ M\ p\ a = Tin\ K\ M\ p - bIn\ K\ p\ a + epsIn\ K\ M\ p\ a$  using
False
    by (simp add: Lin-def)
    ultimately show ?thesis using Tin-le[OF assms(1,2)] by linarith

```

qed

lemma *nu-le*:

assumes *prime p InnerHyp K M p x ∈ set (inner-idx K M p)*

shows $\text{nu } K \ M \ p \ x \ j \leq 4 * M + 10$

proof –

have $m: \text{fst } x \leq m\text{Half } p \ \text{snd } x < \text{nat } (\text{Lin } K \ M \ p \ (\text{fst } x))$ using *assms(3)*

by (*simp-all add: mem-inner-idx*)

have $l: \text{nat } (\text{Lin } K \ M \ p \ a) \leq 4 * M + 10$ if $a \leq m\text{Half } p$ for a

using *Lin-le[OF assms(1,2) that]* by (*simp add: nat-le-iff*)

show *?thesis* unfolding *nu-def* using $m \ l[\text{of } \text{fst } x] \ l[\text{of } j]$ by *auto*

qed

lemma *ellN-le*:

assumes *prime p InnerHyp K M p 1 ≤ a a ≤ mHalf p*

shows $\text{ell } p \ (\text{Nof } K) \ a \leq M$

proof –

have $o: \text{odd } p$ using *hyp-facts[OF assms(1,2)]* by *blast*

have $b: \text{ell } p \ (\text{Nof } K) \ a \leq 2 * (\text{Nof } K \ \text{div } p) + 2$

using *ell-bounds[OF o assms(3,4), of Nof K]* by *blast*

have $H: 40 \leq M \ K < p * M$ using *assms(2)* by (*auto simp: InnerHyp-def*)

have $N: 40 * \text{Nof } K \leq 3 * K$ by (*simp add: Nof-def*)

have $q: \text{Nof } K \ \text{div } p * p \leq \text{Nof } K$ by (*rule div-times-less-eq-dividend*)

have $p * (40 * (\text{Nof } K \ \text{div } p)) < p * (3 * M)$

proof –

have $p * (40 * (\text{Nof } K \ \text{div } p)) = 40 * (\text{Nof } K \ \text{div } p * p)$ by (*simp add: ac-simps*)

also have $\dots \leq 40 * \text{Nof } K$ using q by *simp*

also have $\dots \leq 3 * K$ by (*rule N*)

also have $\dots < 3 * (p * M)$ using $H(2)$ by *simp*

also have $\dots = p * (3 * M)$ by (*simp add: ac-simps*)

finally show *?thesis* .

qed

hence $0 < p \wedge 40 * (\text{Nof } K \ \text{div } p) < 3 * M$ by (*simp only: mult-less-cancel1*)

hence $40 * (\text{Nof } K \ \text{div } p) < 3 * M$ by *simp*

thus *?thesis* using $b \ H(1)$ by *linarith*

qed

lemma *eps-key*:

assumes *prime p InnerHyp K M p a ∈ {1..mHalf p} b ∈ {1..mHalf p}*

shows $2 * (\text{epsIn } K \ M \ p \ a - \text{epsIn } K \ M \ p \ b) \leq 2 + \text{int } (\text{ell } p \ K \ a) - \text{int } (\text{ell } p \ K \ b)$

proof –

define r where $r = (\lambda x \ y. \text{ell } p \ K \ y < \text{ell } p \ K \ x \vee (\text{ell } p \ K \ x = \text{ell } p \ K \ y \wedge x < y))$

have $\text{rk}: \text{defaultPos } K \ p \ x = \text{card } \{c \in \{1..m\text{Half } p\}. r \ c \ x\}$ for x

by (*simp add: defaultPos-def r-def*)

have *irr*: $\bigwedge x. \neg r \ x \ x$ by (*simp add: r-def*)

have *tr*: $\bigwedge x \ y \ z. r \ x \ y \implies r \ y \ z \implies r \ x \ z$ unfolding *r-def*

by *auto*

```

have tot:  $\bigwedge x y. x \in \{1..mHalf\ p\} \implies$ 
   $y \in \{1..mHalf\ p\} \implies x \neq y \implies r\ x\ y \vee r\ y\ x$ 
  unfolding r-def by auto
have iff: defaultPos K p x < defaultPos K p y  $\longleftrightarrow r\ x\ y$ 
  if  $x \in \{1..mHalf\ p\}$   $y \in \{1..mHalf\ p\}$  for x y
  unfolding rk using rank-props(1)[OF finite-atLeastAtMost irr tr tot that] .
have cnt: card {c  $\in \{1..mHalf\ p\}. defaultPos K p c < defaultPos K p x$ } =
  defaultPos K p x
  if  $x \in \{1..mHalf\ p\}$  for x
proof -
  have {c  $\in \{1..mHalf\ p\}. defaultPos K p c < defaultPos K p x$ } = {c  $\in$ 
    {1..mHalf p}. r c x} using iff that by auto
  thus ?thesis using rk[of x] by simp
qed
have ea: epsIn K M p a = (if int (defaultPos K p a) < Ein K M p then 1 else 0)
  unfolding epsIn-def cnt[OF assms(3)] ..
have eb: epsIn K M p b = (if int (defaultPos K p b) < Ein K M p then 1 else 0)
  unfolding epsIn-def cnt[OF assms(4)] ..
have o: odd p using hyp-facts[OF assms(1,2)] by blast
have ba:  $2 * (K \text{ div } p) \leq ell\ p\ K\ a \wedge ell\ p\ K\ a \leq 2 * (K \text{ div } p) + 2$ 
  using ell-bounds[OF o, of a K] assms(3) by auto
have bb:  $2 * (K \text{ div } p) \leq ell\ p\ K\ b \wedge ell\ p\ K\ b \leq 2 * (K \text{ div } p) + 2$ 
  using ell-bounds[OF o, of b K] assms(4) by auto
show ?thesis
proof (cases int (defaultPos K p a) < Ein K M p  $\wedge$ 
 $\neg int (defaultPos K p b) < Ein K M p$ )
  case True
  hence defaultPos K p a < defaultPos K p b by linarith
  hence r a b using iff assms(3,4) by blast
  hence  $ell\ p\ K\ b \leq ell\ p\ K\ a$  unfolding r-def by auto
  hence int (ell p K b)  $\leq$  int (ell p K a) by simp
  thus ?thesis unfolding ea eb using True by simp
next
  case False
  thus ?thesis unfolding ea eb using ba bb by auto
qed
qed

```

Half-weight at an ordinary source b .

lemma half-ord:

```

assumes prime p InnerHyp K M p x  $\in set (inner-idx\ K\ M\ p)$   $1 \leq b$   $b \leq mHalf$ 
p
shows  $wInIdx\ K\ M\ p\ x \leq of-nat (nu\ K\ M\ p\ x\ b) + 3 * of-nat (ell\ p\ (Nof\ K)\ b)$ 
-
  ( $of-nat (ell\ p\ K\ b) + 4$ ) / 2
proof -
  obtain a i where  $x = (a, i)$  by (cases x)
  have m:  $a \leq mHalf\ p\ i < nat (Lin\ K\ M\ p\ a)$  using assms(3) unfolding x
  by (simp-all add: mem-inner-idx)

```

```

have bnz:  $b \neq 0$  using assms(4) by simp
have  $\text{Lin } K \ M \ p \ b = \text{Tin } K \ M \ p - \text{bIn } K \ p \ b + \text{epsIn } K \ M \ p \ b$  using bnz by
(simp add: Lin-def)
moreover have  $0 \leq \text{Lin } K \ M \ p \ b$  by (rule Lin-nonneg[OF assms(1,2,5)])
ultimately have ratLb:  $(\text{of-nat } (\text{nat } (\text{Lin } K \ M \ p \ b)) :: \text{rat}) =$ 
 $\text{of-int } (\text{Tin } K \ M \ p) - 3 * \text{of-nat } (\text{ell } p \ (\text{Nof } K) \ b) + \text{of-int } (\text{epsIn } K \ M \ p \ b)$ 
by (simp add: bIn-def)
show ?thesis
proof (cases  $a = 0$ )
case True
have  $w\text{InIdx } K \ M \ p \ x \leq \text{of-int } (\text{Zin } K \ M \ p \ b) - (\text{of-nat } (\text{ell } p \ K \ b) + 4) / 2$ 
unfolding wInIdx-def x fst-conv snd-conv wIn-def if-P[OF True]
by (rule Min-le) (use assms(4,5) in auto)
moreover have  $(\text{of-nat } (\text{nu } K \ M \ p \ x \ b) :: \text{rat}) =$ 
 $\text{of-int } (\text{Tin } K \ M \ p) - 3 * \text{of-nat } (\text{ell } p \ (\text{Nof } K) \ b) + \text{of-int } (\text{epsIn } K \ M \ p \ b)$ 
unfolding x nu-def using True assms(5) bnz ratLb by simp
moreover have  $(\text{of-int } (\text{Zin } K \ M \ p \ b) :: \text{rat}) = \text{of-int } (\text{Tin } K \ M \ p) + \text{of-int}$ 
 $(\text{epsIn } K \ M \ p \ b)$ 
by (simp add: Zin-def)
ultimately show ?thesis by linarith
next
case False
show ?thesis
proof (cases  $a = b$ )
case True
show ?thesis unfolding wInIdx-def x fst-conv snd-conv wIn-def using True
bnz
by (simp add: bIn-def nu-def assms(5))
next
case ab: False
have a1:  $a \in \{1..m\text{Half } p\}$  using False m(1) by simp
have b1:  $b \in \{1..m\text{Half } p\}$  using assms(4,5) by simp
have La:  $\text{Lin } K \ M \ p \ a = \text{Tin } K \ M \ p - \text{bIn } K \ p \ a + \text{epsIn } K \ M \ p \ a$  using
False
by (simp add: Lin-def)
have  $\text{int } i + 1 \leq \text{Lin } K \ M \ p \ a$  using m(2) by arith
hence r1:  $\text{rat-of-int } (\text{int } i + 1) \leq \text{rat-of-int } (\text{Tin } K \ M \ p - \text{bIn } K \ p \ a + \text{epsIn}$ 
 $K \ M \ p \ a)$ 
unfolding La by (simp only: of-int-le-iff)
have  $2 * (\text{epsIn } K \ M \ p \ a - \text{epsIn } K \ M \ p \ b) \leq 2 + \text{int } (\text{ell } p \ K \ a) - \text{int } (\text{ell}$ 
 $p \ K \ b)$ 
by (rule eps-key[OF assms(1,2) a1 b1])
hence r2:  $\text{rat-of-int } (2 * (\text{epsIn } K \ M \ p \ a - \text{epsIn } K \ M \ p \ b)) \leq$ 
 $\text{rat-of-int } (2 + \text{int } (\text{ell } p \ K \ a) - \text{int } (\text{ell } p \ K \ b))$ 
by (simp only: of-int-le-iff)
have w:  $w\text{InIdx } K \ M \ p \ x = \text{of-nat } i + 3 * \text{of-nat } (\text{ell } p \ (\text{Nof } K) \ a) - (\text{of-nat}$ 
 $(\text{ell } p \ K \ a) + 4) /$ 
 $2$ 
unfolding wInIdx-def x fst-conv snd-conv wIn-def using False by (simp

```

```

add: bIn-def)
  have nb: (of-nat (nu K M p x b) :: rat) =
    of-int (Tin K M p) - 3 * of-nat (ell p (Nof K) b) + of-int (epsIn K M p b)
  unfolding x nu-def using ab assms(5) ratLb by simp
  have of-nat i + 1 ≤ (of-int (Tin K M p) - 3 * of-nat (ell p (Nof K) a) +
    of-int (epsIn K M p a) :: rat)
  using r1 by (simp add: bIn-def)
  moreover have 2 * (of-int (epsIn K M p a) - of-int (epsIn K M p b)) ≤
    (2 + of-nat (ell p K a) - of-nat (ell p K b) :: rat)
  using r2 by simp
  ultimately show ?thesis unfolding w nb
  by (simp only: distrib-left right-diff-distrib add-divide-distrib diff-divide-distrib;
    linarith)
qed
qed
qed

Half-weight at the zero source.

lemma half-zero:
  assumes prime p InnerHyp K M p x ∈ set (inner-idx K M p)
  shows wInIdx K M p x ≤ 2 * of-nat (nu K M p x 0) + 6 * of-nat (Nof K div
p) -
  of-nat (K div p) + 1 / 2
proof -
  obtain a i where x: x = (a, i) by (cases x)
  have m: a ≤ mHalf p i < nat (Lin K M p a) using assms(3) unfolding x
  by (simp-all add: mem-inner-idx)
  show ?thesis
  proof (cases a = 0)
    case True
      have w: wInIdx K M p x ≤ 2 * of-nat i + 6 * of-nat (Nof K div p) - of-nat
(K div p) + 1 /
      2
      unfolding wInIdx-def x fst-conv snd-conv wIn-def if-P[OF True] by (rule
Min-le) auto
      have n: nu K M p x 0 = i unfolding x nu-def using True by simp
      show ?thesis using w unfolding n .
    next
      case False
      have Tl: Tin K M p ≤ 4 * int M by (rule Tin-le[OF assms(1,2)])
      have e: epsIn K M p a ≤ 1 by (simp add: epsIn-def)
      have L: Lin K M p a = Tin K M p - bIn K p a + epsIn K M p a using False
by (simp add: Lin-def)
      have iL: int i + 1 ≤ Lin K M p a using m(2) by arith
      have n0: nu K M p x 0 = 4 * M + 10 unfolding x nu-def using False
      by (simp add: Lin-def L0-def)
      have Kq: K div p < M using hyp-facts[OF assms(1,2)] by blast
      have w: wInIdx K M p x = of-nat i + of-int (bIn K p a) - (of-nat (ell p K a)
+ 4) / 2

```



```

    unfolding wInIdx-def x fst-conv snd-conv wIn-def using False by simp
    have r1: rat-of-int (int i + 1) ≤ rat-of-int (Tin K M p - bIn K p a + epsIn
K M p a)
    using iL unfolding L by (simp only: of-int-le-iff)
    have r2: rat-of-int (Tin K M p) ≤ rat-of-int (4 * int M) using Tl
    by (simp only: of-int-le-iff)
    have r3: rat-of-int (epsIn K M p a) ≤ rat-of-int 1 using e by (simp only:
of-int-le-iff)
    have r4: (of-nat (K div p) :: rat) ≤ of-nat M using Kq by simp
    have r5: (0 :: rat) ≤ of-nat (ell p K a) (0 :: rat) ≤ of-nat (Nof K div p)
    by simp-all
    have c4: (of-nat (4 * M + 10) :: rat) = 4 * of-nat M + 10 by simp
    have r6: (0 :: rat) ≤ of-nat M (0 :: rat) ≤ of-nat i by simp-all
    show ?thesis unfolding w n0 c4 using r1 r2 r3 r4 r5 r6
    by (simp only: of-int-add of-int-diff of-int-mult of-int-of-nat-eq of-int-1 of-int-numeral
distrib-left right-diff-distrib add-divide-distrib diff-divide-distrib; linarith)
qed
qed

```

(4.2)/(4.3): every entry of the inner Gram matrix satisfies the weights.

theorem *innerGram-entry-bound*:

```

    assumes p: prime p and H: InnerHyp K M p and x: x < length (inner-idx K
M p) and y: y <
length (inner-idx K M p)
    shows vpGge p (innerGram K M p $$ (x, y))
    (wInIdx K M p (inner-idx K M p ! x) + wInIdx K M p (inner-idx K M p
! y))

```

proof –

```

    define ix where ix = inner-idx K M p ! x
    define iy where iy = inner-idx K M p ! y
    have ixs: ix ∈ set (inner-idx K M p) and iys: iy ∈ set (inner-idx K M p)
    by (simp-all add: ix-def iy-def x y)
    obtain p7: 7 ≤ p and oddp: odd p and NpM: Nof K div p < M
    using hyp-facts[OF p H] by blast
    have K1: 2 * K < p ^ 2 and K2: K + p < p ^ 2
    proof –
        have b: 200 * M < p ∧ 5 * K < p ^ 2 by (rule innerHyp-bounds[OF H])
        have KM: K < p * M using H by (simp add: InnerHyp-def)
        have p * (M + 1) ≤ p * p using b by (intro mult-le-mono2) linarith
        moreover have p * (M + 1) = p * M + p by simp
        ultimately have K + p < p * p using KM by linarith
        thus K + p < p ^ 2 by (simp add: power2-eq-square)
        show 2 * K < p ^ 2 using b by linarith
    qed
    have entry: innerGram K M p $$ (x, y) = muX {1..K} (∏ j ∈
Jset K p. (pX + [:of-nat j ^ 2:]) ^ kexp K M p ix iy j)
    using x y numerator-eq[of ix p iy K M] mem-inner-idx[of ix K M p] mem-inner-idx[of
iy K M p] ix
iys

```

```

    by (simp add: innerGram-def Let-def ix-def iy-def mult.assoc)
  define B where B = wInIdx K M p ix + wInIdx K M p iy
  have hB:  $\bigwedge c. c < p \implies$ 
     $B + 4 \leq \text{of-nat } (eNum \ p \ c \ (Jset \ K \ p) \ (kexp \ K \ M \ p \ ix \ iy)) - \text{of-nat } (\text{nearCount}$ 
     $p \ K \ c)$ 
  proof -
    fix c assume c:  $c < p$ 
    have nc:  $\text{nearCount } p \ K \ c = \text{mult } c * \text{ell } p \ K \ (\text{cls } p \ c)$ 
    by (simp only: nearCount-eq[OF p] sum-omg[OF oddp c])
    show  $B + 4 \leq \text{of-nat } (eNum \ p \ c \ (Jset \ K \ p) \ (kexp \ K \ M \ p \ ix \ iy)) - \text{of-nat}$ 
     $(\text{nearCount } p \ K \ c)$ 
    proof (cases c = 0)
      case True
        have hx:  $wInIdx \ K \ M \ p \ ix \leq 2 * \text{of-nat } (\text{nu } K \ M \ p \ ix \ 0) + 6 * \text{of-nat } (\text{Nof}$ 
         $K \ \text{div } p) -$ 
         $\text{of-nat } (K \ \text{div } p) + 1 / 2$ 
        and hy:  $wInIdx \ K \ M \ p \ iy \leq 2 * \text{of-nat } (\text{nu } K \ M \ p \ iy \ 0) + 6 * \text{of-nat } (\text{Nof}$ 
         $K \ \text{div } p) -$ 
         $\text{of-nat } (K \ \text{div } p) + 1 / 2$ 
        by (rule half-zero[OF p H ix]) (rule half-zero[OF p H iys])
        have e:  $\text{of-nat } (eNum \ p \ c \ (Jset \ K \ p) \ (kexp \ K \ M \ p \ ix \ iy)) - \text{of-nat } (\text{nearCount}$ 
         $p \ K \ c) =$ 
         $(5 + 2 * (6 * \text{of-nat } (\text{Nof } K \ \text{div } p) + \text{of-nat } (\text{nu } K \ M \ p \ ix \ 0) + \text{of-nat}$ 
         $(\text{nu } K \ M \ p \ iy \ 0)) -$ 
         $2 * \text{of-nat } (K \ \text{div } p) :: \text{rat})$ 
        unfolding eNum-eq[OF oddp c] nc using True
        by (simp add: cls-zero mult-def ell-zero-eq[OF prime-gt-0-nat[OF p]])
        show ?thesis unfolding B-def e using hx hy
        by (simp only: distrib-left right-diff-distrib add-divide-distrib diff-divide-distrib;
            linarith)
      case False
        next
        case False
          have cls-ge1:  $1 \leq \text{cls } p \ c$  using cls-pos[OF False c] .
          have cls-le-half:  $\text{cls } p \ c \leq mHalf \ p$  using cls-le[OF oddp] .
          have hx:  $wInIdx \ K \ M \ p \ ix \leq \text{of-nat } (\text{nu } K \ M \ p \ ix \ (\text{cls } p \ c)) + 3 *$ 
           $\text{of-nat } (\text{ell } p \ (\text{Nof } K) \ (\text{cls } p \ c)) - (\text{of-nat } (\text{ell } p \ K \ (\text{cls } p \ c)) + 4) / 2$ 
          and hy:  $wInIdx \ K \ M \ p \ iy \leq \text{of-nat } (\text{nu } K \ M \ p \ iy \ (\text{cls } p \ c)) + 3 *$ 
           $\text{of-nat } (\text{ell } p \ (\text{Nof } K) \ (\text{cls } p \ c)) - (\text{of-nat } (\text{ell } p \ K \ (\text{cls } p \ c)) + 4) / 2$ 
          by (rule half-ord[OF p H ix cls-ge1 cls-le-half]) (rule half-ord[OF p H iys
            cls-ge1
            cls-le-half])
          have e:  $\text{of-nat } (eNum \ p \ c \ (Jset \ K \ p) \ (kexp \ K \ M \ p \ ix \ iy)) - \text{of-nat } (\text{nearCount}$ 
           $p \ K \ c) =$ 
           $(6 * \text{of-nat } (\text{ell } p \ (\text{Nof } K) \ (\text{cls } p \ c)) + \text{of-nat } (\text{nu } K \ M \ p \ ix \ (\text{cls } p \ c)) +$ 
           $\text{of-nat } (\text{nu } K \ M \ p \ iy \ (\text{cls } p \ c))$ 
           $- \text{of-nat } (\text{ell } p \ K \ (\text{cls } p \ c)) :: \text{rat})$ 
          unfolding eNum-eq[OF oddp c] nc using False by (simp add: mult-def)
          show ?thesis unfolding B-def e using hx hy
          by (simp only: distrib-left right-diff-distrib add-divide-distrib diff-divide-distrib;

```

```

    linarith)
  qed
qed
have hdeg:  $\bigwedge c. c < p \implies eNum\ p\ c\ (Jset\ K\ p)\ (kexp\ K\ M\ p\ ix\ iy) \leq p + 1$ 
proof -
  fix c assume c:  $c < p$ 
  have b:  $200 * M < p$  using innerHyp-bounds[OF H] by blast
  have M40:  $40 \leq M$  using H by (simp add: InnerHyp-def)
  have nx:  $nu\ K\ M\ p\ ix\ (cls\ p\ c) \leq 4 * M + 10$  by (rule nu-le[OF p H ix])
  have ny:  $nu\ K\ M\ p\ iy\ (cls\ p\ c) \leq 4 * M + 10$  by (rule nu-le[OF p H iy])
  have el:  $ell\ p\ (Nof\ K)\ (cls\ p\ c) \leq M$ 
  proof (cases c = 0)
    case True
    thus ?thesis using NpM by (simp add: cls-zero ell-zero-eq[OF prime-gt-0-nat[OF p]])
  next
    case False
    show ?thesis by (rule ellN-le[OF p H cls-pos[OF False c] cls-le[OF oddp]])
  qed
  show  $eNum\ p\ c\ (Jset\ K\ p)\ (kexp\ K\ M\ p\ ix\ iy) \leq p + 1$ 
  proof (cases c = 0)
    case True
    have  $eNum\ p\ c\ (Jset\ K\ p)\ (kexp\ K\ M\ p\ ix\ iy) =$ 
       $5 + 2 * (6 * ell\ p\ (Nof\ K)\ (cls\ p\ c) + nu\ K\ M\ p\ ix\ (cls\ p\ c) + nu\ K\ M\ p\ iy\ (cls\ p\ c))$ 
    unfolding eNum-eq[OF oddp c] using True by (simp add: mult-def)
    thus ?thesis using nx ny el b M40 by fastforce
  next
    case False
    have  $eNum\ p\ c\ (Jset\ K\ p)\ (kexp\ K\ M\ p\ ix\ iy) =$ 
       $6 * ell\ p\ (Nof\ K)\ (cls\ p\ c) + nu\ K\ M\ p\ ix\ (cls\ p\ c) + nu\ K\ M\ p\ iy\ (cls\ p\ c)$ 
    unfolding eNum-eq[OF oddp c] using False by (simp add: mult-def)
    thus ?thesis using nx ny el b M40 by arith
  qed
qed
qed
show ?thesis
  unfolding entry ix-def[symmetric] iy-def[symmetric] B-def[symmetric]
  by (rule entry-generic[OF p p7 K1 K2 - hB hdeg]) (simp add: Jset-def)
qed

```

Multiplying by a p -adic unit does not change Gauss-valuation bounds.

lemma *vpGge-of-unit-mul*:

assumes $prime\ p\ u \neq 0\ vp\ p\ u = 0\ vpGge\ p\ ([u] * P)\ b$
shows $vpGge\ p\ P\ b$

proof -

have $prime\ (int\ p)$ **using** *assms*(1) **by** (simp add: prime-nat-int-transfer)
hence $pr: prime\ elem\ (int\ p)$ **by** (rule prime-imp-prime-elem)
show ?thesis **unfolding** *vpGge-def*

```

proof (intro allI impI)
  fix i assume c: Polynomial.coeff P i ≠ 0
  have ne: u * Polynomial.coeff P i ≠ 0 using assms(2) c by simp
  have co: Polynomial.coeff ([:u:] * P) i = u * Polynomial.coeff P i by simp
  have l: b ≤ of-int (vp p (u * Polynomial.coeff P i)) using assms(4) ne
    unfolding vpGge-def co[symmetric] by blast
  have e: vp p (u * Polynomial.coeff P i) = vp p (Polynomial.coeff P i)
    using qmultiplicity-mult[OF pr assms(2) c] assms(3) by (simp add: vp-def)
  show b ≤ of-int (vp p (Polynomial.coeff P i)) using l unfolding e .
qed
qed

```

$\sum -i (w (\sigma i) + w i) = 2 \sum -i w i$ for a permutation σ of $\{0..<n\}$.

```

lemma sum-permute-add:
  assumes  $\sigma$  permutes  $\{0..<n\}$ 
  shows  $(\sum i = 0..<n. w (\sigma i) + w i) = 2 * (\sum i = 0..<n. (w i :: rat))$ 
proof -
  have  $(\sum i = 0..<n. w (\sigma i)) = (\sum i = 0..<n. w i)$ 
    using sum.permute[OF assms, of w] unfolding o-def by (rule sym)
  thus ?thesis unfolding sum.distrib mult-2 by simp
qed

```

The Gram matrix of a basis B (rows = coefficient vectors) is $B G B^T$, and its determinant is $\det(B)^2 \det(G)$; a Gauss bound on the Gram determinant transfers to $\det G$ when $\det B$ is a p -adic unit.

```

lemma det-gram-transfer:
  assumes p: prime p and G:  $G \in \text{carrier-mat } n \ n$  and B:  $B \in \text{carrier-mat } n \ n$ 
  and dB: Determinant.det B ≠ 0 vp p (Determinant.det B) = 0
  and gram: vpGge p (Determinant.det (map-mat ( $\lambda q. [:q:]$ ) B * G *
    transpose-mat (map-mat ( $\lambda q. [:q:]$ ) B))) b
  shows vpGge p (Determinant.det G) b
proof -
  have Determinant.det (map-mat ( $\lambda q. [:q:]$ ) B * G *
    transpose-mat (map-mat ( $\lambda q. [:q:]$ ) B)) =
    [:Determinant.det B ^ 2:] * Determinant.det G
  proof -
    have Bp: map-mat ( $\lambda q. [:q:]$ ) B ∈ carrier-mat n n using B by simp
    have Tc: transpose-mat (map-mat ( $\lambda q. [:q:]$ ) B) ∈ carrier-mat n n using Bp
by simp
    have BG: map-mat ( $\lambda q. [:q:]$ ) B * G ∈ carrier-mat n n using Bp G
      by (rule mult-carrier-mat)
    show ?thesis unfolding det-mult[OF BG Tc] det-mult[OF Bp G] det-transpose[OF
      Bp]
      det-map-mat-C[OF B]
      by (simp add: power2-eq-square mult-ac)
  qed
  thus ?thesis using gram vpGge-of-unit-mul[OF p, of Determinant.det B ^ 2] dB
    by (metis power-not-zero dB(1) dB(2) vp-power-rat[OF p dB(1), of 2] mult-zero-right)
qed

```

$\gamma\text{-}p\hat{in} = 2 \sum \{-x \in idx\} w\text{-}x$, the double sum written over the index list.

lemma *sum-list-concat-aux*: *sum-list* (*concat xss*) =
sum-list (*map sum-list* (*xss* :: 'a :: monoid-add list list))
by (*induction xss*) *simp-all*

lemma *gammaIn-eq-sum-idx*:

gammaIn K M p = 2 * ($\sum x < \text{length } (\text{inner-idx } K M p). wInIdx K M p$
(inner-idx K M p ! x))

proof –

have *sum-list* (*map* (*wInIdx K M p*) (*inner-idx K M p*)) =
 $(\sum x < \text{length } (\text{inner-idx } K M p). wInIdx K M p (\text{inner-idx } K M p ! x))$
by (*simp add: sum-list-sum-nth atLeast0LessThan*)

moreover have *sum-list* (*map* (*wInIdx K M p*) (*inner-idx K M p*)) = $(\sum a \leq$
mHalf p. $\sum i < \text{nat } (\text{Lin } K M p a). wIn K M p a i$)

unfolding *inner-idx-def map-concat sum-list-concat-aux map-map*

by (*simp add: o-def wInIdx-def interv-sum-list-conv-sum-set-nat atLeast0LessThan*
lessThan-Suc-atMost del: upt-Suc)

ultimately show ?thesis **unfolding** *gammaIn-def* **by** *simp*

qed

lemma *degree-innerBasis-lt*:

assumes *prime p InnerHyp K M p x* $\in \text{set } (\text{inner-idx } K M p)$

shows *degree* (*innerBasis K M p x*) < *hof K*

proof –

obtain *a i* **where** *x*: *x* = (*a*, *i*) **by** (*cases x*)

have *a*: *a* ≤ *mHalf p* **and** *i*: *i* < *nat* (*Lin K M p a*) **using** *assms(3)* **unfolding**

x

by (*simp-all add: mem-inner-idx*)

have *tot*: $(\sum c \leq mHalf p. \text{nat } (\text{Lin } K M p c)) = \text{hof } K$

proof –

have *int* $(\sum c \leq mHalf p. \text{nat } (\text{Lin } K M p c)) = (\sum c \leq mHalf p. \text{Lin } K M$
p c)

unfolding *of-nat-sum* **by** (*rule sum.cong*) (*use Lin-nonneg[OF assms(1,2)]*)

in *auto*)

also have ... = *int* (*hof K*) **by** (*rule sum-Lin[OF assms(1,2)]*)

finally show ?thesis **by** (*simp only: of-nat-eq-iff*)

qed

have *rem*: $(\sum c \leq mHalf p. \text{nat } (\text{Lin } K M p c)) = \text{nat } (\text{Lin } K M p a) + (\sum c$
 $\in$

$\{0..mHalf p\} - \{a\}. \text{nat } (\text{Lin } K M p c))$

proof –

have *e*: $\{..mHalf p\} = \{0..mHalf p\}$ **by** *auto*

show ?thesis **unfolding** *e* **using** *a* **by** (*simp add: sum.remove*)

qed

have *d1*: *degree* $(\prod c \in \{0..mHalf p\} - \{a\}. (pX +$
 $[:\text{of-nat } c \wedge 2:]) \wedge \text{nat } (\text{Lin } K M p c)) \leq$

$(\sum c \in \{0..mHalf p\} - \{a\}. \text{nat } (\text{Lin } K M p c))$

proof –

have *degree* $(\prod c \in \{0..mHalf p\} - \{a\}. (pX +$

```

    ([:of-nat c ^ 2:]) ^ nat (Lin K M p c)) ≤
    (∑ c ∈ {0..mHalf p} - {a}. degree ((pX + [:of-nat c ^ 2:]) ^ nat (Lin K
M p c)))
    using degree-prod-sum-le[of {0..mHalf p} - {a} λc. (pX +
    [:of-nat c ^ 2:]) ^ nat (Lin K M p c)]
    by (simp add: o-def)
    also have ... ≤ (∑ c ∈ {0..mHalf p} - {a}. nat (Lin K M p c))
    proof (rule sum-mono)
      fix c
      show degree ((pX + [:of-nat c ^ 2:]) ^ nat (Lin K M p c)) ≤ nat (Lin K M
p c)
        using degree-power-le[of pX + [:of-nat c ^ 2:] nat (Lin K M p c)]
        unfolding degree-pX-plus-const by simp
    qed
    finally show ?thesis .
  qed
  have d2: degree ((pX + [:of-nat a ^ 2:]) ^ i) ≤ i
    using degree-power-le[of pX + [:of-nat a ^ 2:] i] unfolding degree-pX-plus-const
  by simp
  have degree (innerBasis K M p x) ≤ (∑ c ∈
    {0..mHalf p} - {a}. nat (Lin K M p c)) + i
    unfolding innerBasis-def x fst-conv snd-conv using degree-mult-le d1 d2
    by (meson add-le-mono order-trans)
  thus ?thesis using tot rem i by linarith
qed

```

The inner Gram matrix is $B \cdot G \cdot K \cdot B^T$ for the coefficient matrix B of the inner basis.

lemma *innerGram-eq-conj*:

```

  assumes p: prime p and H: InnerHyp K M p
  defines B ≡ Matrix.mat (hof K) (hof K) (λ(x,
    d). Polynomial.coeff (innerBasis K M p (inner-idx K M p ! x)) d)
  shows innerGram K M p = map-mat (λq. [:q:]) B * GK K *
    transpose-mat (map-mat (λq. [:q:]) B)
  proof -
    let ?h = hof K and ?idx = inner-idx K M p
    let ?Q = map-mat (λq. [:q:]) B
    have len: length ?idx = ?h by (rule length-inner-idx[OF p H])
    have dB: dim-row B = ?h dim-col B = ?h by (simp-all add: B-def)
    have dG: dim-row (GK K) = ?h dim-col (GK K) = ?h by (simp-all add: GK-def)
    have sm: muX S (Polynomial.smult a P) = Polynomial.smult a (muX S P) for
    S a P
      using muX-C-mul[of S a P] by simp
    have qe: innerBasis K M p (?idx ! x) = (∑ k < ?h. Polynomial.smult (B $$
    (x, k)) (pX ^ k))
      if x: x < ?h for x
    proof -
      have dg: degree (innerBasis K M p (?idx ! x)) ≤ ?h - 1
        using degree-innerBasis-lt[OF p H, of ?idx ! x] x len by simp

```

```

have  $e: \{..?h - 1\} = \{..<?h\}$  using  $x$  by auto
have  $innerBasis\ K\ M\ p\ ( ?idx\ !\ x) =$ 
   $(\sum\ k \leq ?h - 1. Polynomial.monom\ (Polynomial.coeff\ (innerBasis\ K\ M\ p$ 
 $( ?idx\ !\ x))\ k)$ 
   $k)$ 
  by  $(rule\ sym[OF\ poly-as-sum-of-monom's[OF\ dg]])$ 
also have  $\dots = (\sum\ k < ?h. Polynomial.monom\ (Polynomial.coeff\ (innerBasis$ 
 $K\ M\ p\ ( ?idx$ 
 $!\ x))\ k)\ k)$ 
  unfolding  $e\ ..$ 
also have  $\dots = (\sum\ k < ?h. Polynomial.smult\ (B\ \$\$ (x, k))\ (pX \wedge k))$ 
  by  $(rule\ sum.cong[OF\ refl])\ (use\ x\ in\ \langle unfold\ pX-pow-monom\ smult-monom,$ 
 $simp\ add: B-def \rangle)$ 
finally show  $?thesis\ .$ 
qed
have  $pq: Dm\ (Nof\ K) \wedge 6 * innerBasis\ K\ M\ p\ ( ?idx\ !\ x) * innerBasis\ K\ M\ p$ 
 $( ?idx\ !\ y) =$ 
 $(\sum\ k < ?h. \sum\ l < ?h. Polynomial.smult\ (B\ \$\$ (x, k) * B\ \$\$ (y,$ 
 $l))\ (Dm\ (Nof\ K) \wedge 6 * pX \wedge (k + l)))$ 
  if  $x: x < ?h$  and  $y: y < ?h$  for  $x\ y$ 
proof  $-$ 
  have  $pq0: innerBasis\ K\ M\ p\ ( ?idx\ !\ x) * innerBasis\ K\ M\ p\ ( ?idx\ !\ y) = (\sum\ k$ 
 $<$ 
 $?h. \sum\ l < ?h.$ 
 $Polynomial.smult\ (B\ \$\$ (x, k))\ (pX \wedge k) * Polynomial.smult\ (B\ \$\$ (y, l))$ 
 $(pX \wedge l))$ 
  unfolding  $qe[OF\ x]\ qe[OF\ y]$  by  $(rule\ sum-product)$ 
  have  $Dm\ (Nof\ K) \wedge 6 * innerBasis\ K\ M\ p\ ( ?idx\ !\ x) * innerBasis\ K\ M\ p\ ( ?idx$ 
 $!\ y) =$ 
 $Dm\ (Nof\ K) \wedge 6 * (innerBasis\ K\ M\ p\ ( ?idx\ !\ x) * innerBasis\ K\ M\ p\ ( ?idx$ 
 $!\ y))$ 
  by  $(simp\ only: mult.assoc)$ 
  also have  $\dots = (\sum\ k < ?h. \sum\ l <$ 
 $?h. Polynomial.smult\ (B\ \$\$ (x, k) * B\ \$\$ (y, l))\ (Dm\ (Nof\ K) \wedge 6 * pX \wedge (k$ 
 $+ l)))$ 
  unfolding  $pq0\ sum-distrib-left$  by  $(intro\ sum.cong\ refl)\ (simp\ add: power-add$ 
 $mult-ac)$ 
finally show  $?thesis\ .$ 
qed
have  $entry: innerGram\ K\ M\ p\ \$\$ (x, y) = (?Q * GK\ K * transpose-mat\ ?Q)\ \$\$$ 
 $(x, y)$ 
  if  $x: x < ?h$  and  $y: y < ?h$  for  $x\ y$ 
proof  $-$ 
  have  $innerGram\ K\ M\ p\ \$\$ (x, y) =$ 
 $muX\ \{1..K\}\ (Dm\ (Nof\ K) \wedge 6 * innerBasis\ K\ M\ p\ ( ?idx\ !\ x) * innerBasis$ 
 $K\ M\ p\ ( ?idx\ !\ y))$ 
  using  $x\ y\ len$  by  $(simp\ add: innerGram-def\ Let-def)$ 
  also have  $\dots = (\sum\ k < ?h. \sum\ l <$ 
 $?h. [:B\ \$\$ (x, k):] * GK\ K\ \$\$ (k, l) * [:B\ \$\$ (y, l):])$ 

```

```

    unfolding pq[OF x y] muX-sum sm by (intro sum.cong refl) (simp add:
GK-def mult-ac)
    also have ... = (?Q * GK K * transpose-mat ?Q) $$ (x, y)
    proof -
      have m: (?Q * GK K * transpose-mat ?Q) $$ (x, y) =
        (∑ l < ?h. (∑ k < ?h. [:B $$ (x, k):] * GK K $$ (k, l)) * [:B $$ (y, l):])
      using x y by (simp add: scalar-prod-def dB dG atLeast0LessThan)
      show ?thesis unfolding m sum-distrib-right by (rule sum.swap)
    qed
    finally show ?thesis .
  qed
show ?thesis
proof (rule eq-matI)
  fix x y
  assume x: x < dim-row (?Q * GK K * transpose-mat ?Q) and y: y <
    dim-col (?Q * GK K * transpose-mat ?Q)
  have xh: x < ?h and yh: y < ?h using x y by (simp-all add: dB)
  show innerGram K M p $$ (x, y) = (?Q * GK K * transpose-mat ?Q) $$ (x,
y)
    by (rule entry[OF xh yh])
  qed (simp-all add: innerGram-def Let-def len dB dG)
qed

```

(4.5): the inner basis is $\mathbb{Z}$ - p -unimodular: the coefficient matrix of the basis polynomials has a determinant that is a p -adic unit. Proof by the Taylor-matrix criterion (*unimodular-criterion*): expanding row $E-\{a, j\}$ at the point $-c \wedge 2$ to order i gives an upper triangular matrix with the units $\prod -\{c \neq a\} (c \wedge 2 - a \wedge 2) \wedge \{L-c\}$ on the diagonal.

theorem *innerBasis-unimodular*:

```

assumes p: prime p and H: InnerHyp K M p
defines B ≡ Matrix.mat (hof K) (hof K) (λ(x,
d). Polynomial.coeff (innerBasis K M p (inner-idx K M p ! x)) d)
shows Determinant.det B ≠ 0 ∧ ∀p p (Determinant.det B) = 0
proof -
  define h where h = hof K
  define idx where idx = inner-idx K M p
  define rt where rt = (λy. - (of-nat (fst (idx ! y)) ^ 2) :: rat)
  define od where od = (λy. snd (idx ! y))
  define V where V = Matrix.mat h h (λ(d, y). tay (pX ^ d) (rt y) (od y))
  define T where T = B * V
  have len: length idx = h by (simp add: idx-def h-def length-inner-idx[OF p H])
  have Bc: B ∈ carrier-mat h h and Vc: V ∈ carrier-mat h h
    by (simp-all add: B-def V-def h-def)
  have mp: 2 * mHalf p + 1 = p using hyp-facts[OF p H] by blast
  have mem: ∧x. x < h ⇒ idx ! x ∈ set idx by (simp add: len)
  have cls-le: ∧x. x < h ⇒ fst (idx ! x) ≤ mHalf p
    using mem-inner-idx mem by (auto simp: idx-def)
  have Tentry: ∧x y. x < h ⇒ y < h ⇒
    T $$ (x, y) = tay (innerBasis K M p (idx ! x)) (rt y) (od y)

```



```

proof –
  fix  $x\ y$  assume  $x: x < h$  and  $y: y < h$ 
  have  $dg: \text{degree } (\text{innerBasis } K\ M\ p\ (idx\ !\ x)) < h$ 
  using  $\text{degree-innerBasis-lt}[OF\ p\ H, \text{ of } idx\ !\ x]\ \text{mem}[OF\ x]$  by ( $\text{simp add:}$ 
 $idx\text{-def } h\text{-def}$ )
  have  $\text{dim-row } B = h\ \text{dim-col } B = h\ \text{dim-row } V = h\ \text{dim-col } V = h$  using  $Bc$ 
 $Vc$  by auto
  with  $x\ y$  have  $T\ \$\$ (x, y) = (\sum d \in \{0..<h\}. B\ \$\$ (x, d) * V\ \$\$ (d, y))$ 
  by ( $\text{simp add: } T\text{-def scalar-prod-def}$ )
  also have  $\dots = (\sum d < h. \text{Polynomial.coeff } (\text{innerBasis } K\ M\ p\ (idx\ !\ x))\ d * \text{tay } (pX\ ^d)\ (rt\ y)\ (od\ y))$ 
  by ( $\text{rule sum.cong}$ ) ( $\text{use } x\ y$  in  $\langle \text{auto simp: } B\text{-def } V\text{-def } idx\text{-def } h\text{-def} \rangle$ )
  also have  $\dots = \text{tay } (\text{innerBasis } K\ M\ p\ (idx\ !\ x))\ (rt\ y)\ (od\ y)$ 
  by ( $\text{rule tay-eq-sum}[OF\ dg, \text{ symmetric}]$ )
  finally show  $T\ \$\$ (x, y) = \text{tay } (\text{innerBasis } K\ M\ p\ (idx\ !\ x))\ (rt\ y)\ (od\ y) .$ 
qed
have below:  $\bigwedge x\ y. y < x \implies x < h \implies T\ \$\$ (x, y) = 0$ 
proof –
  fix  $x\ y$  assume  $yx: y < x$  and  $x: x < h$ 
  have  $y: y < h$  using  $yx\ x$  by simp
  obtain  $a\ j$  where  $ax: idx\ !\ x = (a, j)$  by ( $\text{cases } idx\ !\ x$ )
  obtain  $c\ i$  where  $cy: idx\ !\ y = (c, i)$  by ( $\text{cases } idx\ !\ y$ )
  have  $my: c \leq mHalf\ p\ i < nat\ (Lin\ K\ M\ p\ c)$  using  $\text{mem}[OF\ y]\ cy$ 
  by ( $\text{simp-all add: } idx\text{-def mem-inner-idx}$ )
  have  $mx: a \leq mHalf\ p$  using  $\text{mem}[OF\ x]\ ax$  by ( $\text{simp add: } idx\text{-def mem-inner-idx}$ )
  have  $lex: c < a \vee (c = a \wedge i < j)$ 
  using  $\text{inner-idx-lex}[of\ y\ x\ K\ M\ p]\ yx\ x\ len\ ax\ cy$  by ( $\text{simp add: } idx\text{-def}$ )
  have  $rr: pX + [:of\text{-nat } c\ ^2:] = pX - [:rt\ y:]$  unfolding  $rt\text{-def } cy$  by simp
  have  $T: T\ \$\$ (x, y) = \text{tay } (\text{innerBasis } K\ M\ p\ (a, j))\ (rt\ y)\ i$ 
  using  $\text{Tentry}[OF\ x\ y]$  unfolding  $ax$  by ( $\text{simp add: } od\text{-def } cy$ )
  show  $T\ \$\$ (x, y) = 0$ 
  proof ( $\text{cases } c < a$ )
  case True
  have  $cin: c \in \{0..mHalf\ p\} - \{a\}$  using  $True\ mx$  by simp
  have  $E: \text{innerBasis } K\ M\ p\ (a, j) = (pX - [:rt\ y:])\ ^{nat\ (Lin\ K\ M\ p\ c)} * \left( \left( \prod_{c' \in \{0..mHalf\ p\} - \{a\} - \{c\}. (pX + [:of\text{-nat } c'\ ^2:])\ ^{nat\ (Lin\ K\ M\ p\ c')}} \right) * (pX + [:of\text{-nat } a\ ^2:])\ ^j \right)$ 
  unfolding  $\text{innerBasis-def fst-conv snd-conv prod.remove}[OF\ \text{finite-Diff}[OF\ \text{finite-atLeastAtMost}]\ cin]\ rr$ 
  by ( $\text{simp only: mult.assoc}$ )
  show  $?thesis$  unfolding  $T\ E$  by ( $\text{rule tay-pow-mult-lt}[OF\ my(2)]$ )
  next
  case False
  hence  $ca: c = a$  and  $ij: i < j$  using  $lex$  by auto
  have  $E: \text{innerBasis } K\ M\ p\ (a, j) = (pX - [:rt\ y:])\ ^j * \left( \prod_{c' \in \{0..mHalf\ p\} - \{a\}. (pX + [:of\text{-nat } c'\ ^2:])\ ^{nat\ (Lin\ K\ M\ p\ c')}} \right)$ 
  unfolding  $\text{innerBasis-def fst-conv snd-conv rr}[unfolding\ ca]$  by ( $\text{rule mult.commute}$ )
  show  $?thesis$  unfolding  $T\ E$  by ( $\text{rule tay-pow-mult-lt}[OF\ ij]$ )

```

```

qed
qed
have diag:  $\bigwedge x. x < h \implies T \$\$ (x, x) =$ 
   $(\prod c \in \{0..mHalf\ p\} - \{fst\ (idx\ !\ x)\}. (of\_nat\ c\ ^2 -$ 
     $of\_nat\ (fst\ (idx\ !\ x))\ ^2) \wedge nat\ (Lin\ K\ M\ p\ c))$ 
proof -
  fix x assume x:  $x < h$ 
  obtain a j where ax:  $idx\ !\ x = (a, j)$  by (cases  $idx\ !\ x$ )
  have rr:  $pX + [:of\_nat\ a\ ^2:] = pX - [:rt\ x:]$  unfolding rt-def ax by simp
  have E:  $innerBasis\ K\ M\ p\ (a, j) = (pX - [:rt\ x:])\ ^j *$ 
     $(\prod c \in \{0..mHalf\ p\} - \{a\}. (pX + [:of\_nat\ c\ ^2:])\ ^{nat\ (Lin\ K\ M\ p\ c)})$ 
    unfolding innerBasis-def fst-conv snd-conv rr by (rule mult.commute)
  have T  $\$\$ (x, x) = tay\ (innerBasis\ K\ M\ p\ (a, j))\ (rt\ x)\ j$ 
    using Tentry[OF x x] unfolding ax by (simp add: od-def ax)
  also have  $\dots = tay\ (\prod c \in$ 
     $\{0..mHalf\ p\} - \{a\}. (pX + [:of\_nat\ c\ ^2:])\ ^{nat\ (Lin\ K\ M\ p\ c)})\ (rt\ x)\ 0$ 
    unfolding E using tay-pow-mult[of rt x j  $\prod c \in$ 
       $\{0..mHalf\ p\} - \{a\}. (pX + [:of\_nat\ c\ ^2:])\ ^{nat\ (Lin\ K\ M\ p\ c)}\ 0]$ 
    by simp
  also have  $\dots = poly\ (\prod c \in$ 
     $\{0..mHalf\ p\} - \{a\}. (pX + [:of\_nat\ c\ ^2:])\ ^{nat\ (Lin\ K\ M\ p\ c)})\ (rt\ x)$ 
    by (rule tay-0-eq-poly)
  also have  $\dots = (\prod c \in$ 
     $\{0..mHalf\ p\} - \{fst\ (idx\ !\ x)\}. (of\_nat\ c\ ^2 -$ 
     $of\_nat\ (fst\ (idx\ !\ x))\ ^2) \wedge nat\ (Lin\ K\ M\ p\ c))$ 
    unfolding ax rt-def by (simp add: poly-prod poly-monom)
  finally show  $T \$\$ (x, x) =$ 
     $(\prod c \in \{0..mHalf\ p\} - \{fst\ (idx\ !\ x)\}. (of\_nat\ c\ ^2 -$ 
     $of\_nat\ (fst\ (idx\ !\ x))\ ^2) \wedge nat\ (Lin\ K\ M\ p\ c)) .$ 
qed
have unit:  $\bigwedge x. x < h \implies T \$\$ (x, x) \neq 0 \wedge vp\ p\ (T \$\$ (x, x)) = 0$ 
proof -
  fix x assume x:  $x < h$ 
  define a where  $a = fst\ (idx\ !\ x)$ 
  have am:  $a \leq mHalf\ p$  using cls-le[OF x] by (simp add: a-def)
  have pr:  $prime\ (int\ p)$  using p by (simp add: prime-nat-int-transfer)
  have fac:  $(of\_nat\ c\ ^2 - of\_nat\ a\ ^2 :: rat) \neq 0 \wedge$ 
     $vp\ p\ (of\_nat\ c\ ^2 - of\_nat\ a\ ^2) = 0$ 
    if c:  $c \in \{0..mHalf\ p\} - \{a\}$  for c
  proof -
    have cm:  $c \leq mHalf\ p$   $c \neq a$  using c by auto
    define z :: int where  $z = (int\ c - int\ a) * (int\ c + int\ a)$ 
    have zq:  $(of\_nat\ c\ ^2 - of\_nat\ a\ ^2 :: rat) = of\_int\ z$ 
      by (simp add: z-def power2-eq-square algebra-simps)
    have  $\neg int\ p\ dvd\ (int\ c - int\ a)$ 
  proof
    assume d:  $int\ p\ dvd\ (int\ c - int\ a)$ 
    have  $int\ c - int\ a \neq 0$  using cm by simp
    from dvd-imp-le-int[OF this d] have  $int\ p \leq |int\ c - int\ a|$  by simp

```

```

      thus False using cm am mp by arith
    qed
  moreover have  $\neg \text{int } p \text{ dvd } (\text{int } c + \text{int } a)$ 
  proof
    assume  $d: \text{int } p \text{ dvd } (\text{int } c + \text{int } a)$ 
    have  $\text{int } c + \text{int } a \neq 0$  using cm by simp
    from dvd-imp-le-int[OF this d] have  $\text{int } p \leq |\text{int } c + \text{int } a|$  by simp
    thus False using cm am mp by arith
  qed
  ultimately have  $\text{nz}: \neg \text{int } p \text{ dvd } z$  unfolding z-def using prime-dvd-mult-iff[OF
pr]
    by blast
  hence  $z \neq 0$  by auto
  thus ?thesis unfolding zq using nz
    by (simp add: vp-def qmultiplicity-of-int not-dvd-imp-multiplicity-0)
  qed
  have  $d: T \text{ } \$\$ (x, x) = (\prod c \in$ 
     $\{0..m\text{Half } p\} - \{a\}. (\text{of-nat } c \wedge 2 - \text{of-nat } a \wedge 2) \wedge \text{nat } (\text{Lin } K \text{ } M \text{ } p \text{ } c))$ 
    using diag[OF x] by (simp add: a-def)
  have  $\text{ne}: (\prod c \in \{0..m\text{Half } p\} - \{a\}. (\text{of-nat } c \wedge 2 -$ 
     $\text{of-nat } a \wedge 2 :: \text{rat}) \wedge \text{nat } (\text{Lin } K \text{ } M \text{ } p \text{ } c)) \neq 0$ 
    using fac by (simp add: prod-zero-iff)
  have  $\text{vp } p (\prod c \in \{0..m\text{Half } p\} - \{a\}. (\text{of-nat } c \wedge 2 -$ 
     $\text{of-nat } a \wedge 2 :: \text{rat}) \wedge \text{nat } (\text{Lin } K \text{ } M \text{ } p \text{ } c)) =$ 
     $(\sum c \in \{0..m\text{Half } p\} - \{a\}. \text{vp } p ((\text{of-nat } c \wedge 2 -$ 
     $\text{of-nat } a \wedge 2) \wedge \text{nat } (\text{Lin } K \text{ } M \text{ } p \text{ } c)))$ 
    by (rule vp-prod-rat[OF p]) (use fac in auto)
  also have  $\dots = 0$  using fac by (simp add: vp-power-rat[OF p])
  finally show  $T \text{ } \$\$ (x, x) \neq 0 \wedge \text{vp } p (T \text{ } \$\$ (x, x)) = 0$  unfolding d using
ne by simp
  qed
  have  $\text{Bi}: \bigwedge x \text{ } d. x < h \implies d < h \implies \text{vge } p (B \text{ } \$\$ (x, d)) \text{ } 0$ 
    using vpGge-innerBasis[OF p] by (auto simp: B-def h-def vpGge-iff)
  have  $\text{Vi}: \bigwedge d \text{ } y. d < h \implies y < h \implies \text{vge } p (V \text{ } \$\$ (d, y)) \text{ } 0$ 
  proof -
    fix  $d \text{ } y$  assume  $d: d < h$  and  $y: y < h$ 
    have  $P: \text{vpGge } p (pX \wedge d) \text{ } 0$  using vpGge-pow[OF p vpGge-pX, of d] by simp
    have  $\text{vge } p (\text{of-nat } (\text{fst } (\text{idx } ! \text{ } y)) \wedge 2) \text{ } 0$ 
      using vge-pow[OF p vge-of-int[OF p, of int (fst (idx ! y))]] by simp
    hence  $r: \text{vge } p (\text{rt } y) \text{ } 0$  unfolding rt-def by (rule vge-neg)
    show  $\text{vge } p (V \text{ } \$\$ (d, y)) \text{ } 0$  using vge-tay[OF p P r, of od y] d y by (simp
add: V-def)
  qed
  show ?thesis
    unfolding h-def[symmetric]
    by (rule unimodular-criterion[OF p Bc Vc Bi Vi])
      (use below unit in ⟨simp-all add: T-def[symmetric] vge-zero⟩)
  qed

```

Proposition 4.1 from the entry bounds (unimodular change of basis + $\text{vpGge-det-of-terms}$).

theorem *DeltaK-vpGge-inner-of-entries*:
assumes p : prime p **and** H : *InnerHyp* K M p
and ent: $\bigwedge x y. x < \text{length } (\text{inner-idx } K \ M \ p) \implies$
 $y < \text{length } (\text{inner-idx } K \ M \ p) \implies$
 $\text{vpGge } p \ (\text{innerGram } K \ M \ p \ \$\$ (x, y))$
 $(w\text{InIdx } K \ M \ p \ (\text{inner-idx } K \ M \ p \ ! \ x) + w\text{InIdx } K \ M \ p \ (\text{inner-idx } K \ M$
 $p \ ! \ y))$
shows $\text{vpGge } p \ (\text{DeltaK } K) \ (\text{gammaIn } K \ M \ p)$
proof –
define h **where** $h = \text{hof } K$
define B **where** $B = \text{Matrix.mat } (\text{hof } K) \ (\text{hof } K) \ (\lambda(x,$
 $d). \text{Polynomial.coeff } (\text{innerBasis } K \ M \ p \ (\text{inner-idx } K \ M \ p \ ! \ x)) \ d)$
define w **where** $w = (\lambda x. w\text{InIdx } K \ M \ p \ (\text{inner-idx } K \ M \ p \ ! \ x))$
have len : $\text{length } (\text{inner-idx } K \ M \ p) = h$ **by** (*simp add*: $h\text{-def length-inner-idx[OF } p \ H]$)
have Bc : $B \in \text{carrier-mat } h \ h$ **and** Gc : $GK \ K \in$
 $\text{carrier-mat } h \ h$ **and** IGc : $\text{innerGram } K \ M \ p \in \text{carrier-mat } h \ h$
by (*simp-all add*: $B\text{-def } GK\text{-def innerGram-def } h\text{-def len Let-def}$)
have uni : $\text{Determinant.det } B \neq 0 \wedge \text{vp } p \ (\text{Determinant.det } B) = 0$
unfolding $B\text{-def}$ **by** (*rule innerBasis-unimodular[OF } p \ H])
have gramv : $\text{vpGge } p \ (\text{Determinant.det } (\text{innerGram } K \ M \ p)) \ (2 * (\sum i = 0..<h.$
 $w \ i))$
proof (*rule vpGge-det-of-terms[OF } p \ IGc]*)
fix σ **assume** σ : σ *permutes* $\{0..<h\}$
have $\text{vpGge } p \ (\prod i = 0..<h. \text{innerGram } K \ M \ p \ \$\$ (i, \sigma \ i)) \ (\sum i =$
 $0..<h. w \ i + w \ (\sigma \ i))$
proof (*rule vpGge-prod[OF } p]*)
show *finite* $\{0..<h\}$ **by** *simp*
next
fix i **assume** i : $i \in \{0..<h\}$
have si : $\sigma \ i < h$ **using** *permutes-in-image[OF } σ]* i **by** *simp*
show $\text{vpGge } p \ (\text{innerGram } K \ M \ p \ \$\$ (i, \sigma \ i)) \ (w \ i + w \ (\sigma \ i))$
unfolding $w\text{-def}$ **by** (*rule ent*) (*use* $i \ si \ \text{len}$ **in** *auto*)
qed
thus $\text{vpGge } p \ (\prod i = 0..<h. \text{innerGram } K \ M \ p \ \$\$ (i, \sigma \ i)) \ (2 * (\sum i =$
 $0..<h. w \ i))$
using *sum-permute-add[OF } σ , of w]* **by** (*simp add*: *add commute*)
qed
have $\text{vpGge } p \ (\text{DeltaK } K) \ (2 * (\sum i = 0..<h. w \ i))$
unfolding DeltaK-def
by (*rule det-gram-transfer[OF } $p \ Gc \ Bc$ conjunct1[OF uni] conjunct2[OF uni]]*)
(use $\text{gramv innerGram-eq-conj[OF } p \ H]$ **in** *simp add*: $B\text{-def}$)
thus *?thesis*
using $\text{gammaIn-eq-sum-idx[of } K \ M \ p]$ **by** (*simp add*: $w\text{-def len atLeast0LessThan}$)
qed*

Proposition 4.1: under (4.1), $v\text{-p}^{\wedge}G(\Delta\text{-}K) \geq \gamma\text{-p}^{\wedge}in$.

theorem *DeltaK-vpGge-inner*: prime $p \implies \text{InnerHyp } K \ M \ p \implies$
 $\text{vpGge } p \ (\text{DeltaK } K) \ (\text{gammaIn } K \ M \ p)$

```

    by (rule DeltaK-vpGge-inner-of-entries, assumption+, rule innerGram-entry-bound)
end

theory Zeta5-Outer-Range
  imports Zeta5-Inner-Range BenOr-Kozen-Reif.More-Matrix
begin

```

22 The outer range: Lemma 4.2 and Proposition 4.3

Section 4.2 of the paper: the decomposition (4.10), the outer basis (4.11) and its unimodularity, the arithmetic of the pole classes, the entry bound (4.12), the closed form (4.14) of γ_p^{out} , Lemma 4.2 (*det-vpGge-of-weighted-entries*) and Proposition 4.3 (*DeltaK-vpGge-outer*, *DeltaK-vpGge-large-prime*). Lean theories *MuMod*, *CRTBasis*, *OuterCount*, *OuterEntries*, *Lemma42*, *Prop43*, *Prop43Large*.

22.1 Section 4.2 The outer range

(4.9) holds when $K \geq 200M^2$ and $K/3 < p \leq K$.

lemma *outer-hyp-of-range*:

```

  assumes  $40 \leq M$   $40 \text{ dvd } K$   $200 * M^2 \leq K$  prime  $p$   $K < 3 * p$   $p \leq K$ 
  shows OuterHyp  $K$   $p$ 
proof -
  have  $200 * 40^2 \leq 200 * M^2$  using assms(1) by (intro mult-le-mono2
power-mono) simp-all
  hence  $K: 320000 \leq K$  using assms(3) by simp
  have  $N: 40 * \text{Nof } K \leq 3 * K$  by (simp add: Nof-def)
  have  $p: 106666 < p$  using  $K$  assms(5) by simp
  have  $2 * K < 3 * p * 2$  using assms(5) by simp
  also have  $\dots \leq p * p$  using  $p$  by (auto intro: mult-le-mono)
  finally have  $2 * K < p^2$  by (simp add: power2-eq-square)
  show ?thesis unfolding OuterHyp-def
proof (intro conjI)
  show  $7 \leq p$  using  $p$  by simp
  show  $p \leq K$  by fact
  show  $K < 3 * p$  by fact
  show  $2 * K < p^2$  by fact
  show  $2 * \text{Nof } K < p$  using  $N$  assms(5) by linarith
  show  $5 * \text{Nof } K \leq 2 * p - 2$  using  $N$  assms(5)  $p$  by linarith
qed
qed

```

(4.10): $G-K = A + p^{-1}L$ with L constant, integral, of rank $\leq r-p$ (rank over $\mathbb{Q}$, *vec-space.rank* of *Jordan_Normal_Form*).

lemma *cCorr-vge*: $\text{prime } p \implies \text{vge } p \text{ (cCorr } p \text{ e) } 0$
by (*auto simp*: *cCorr-def vge-zero vge-of-int*)

lemma *cCorr-of-lt*: $e < 2 * p - 3 \implies \text{cCorr } p \text{ e} = 0$
by (*simp add*: *cCorr-def*)

$\text{mu-X} = \text{mu}^0\text{-X} + p^{-1} \sum -e \text{ q-e c-e}$: the two functionals differ by the correction of the polynomial part.

lemma *muX-eq-muXmod-add*:
 $\text{prime } p \implies \text{muX } S \text{ P} = \text{muXmod } p \text{ S P} +$
 $[(\sum e \leq \text{degree } (P \text{ div sqPoleProd } S). \text{Polynomial.coeff } (P \text{ div sqPoleProd } S)$
 $e * \text{cCorr } p \text{ e}) / \text{of-nat } p:]$

proof –
assume *prime p*
define *Q* **where** $Q = P \text{ div sqPoleProd } S$
define *Sg* **where** $Sg = (\sum j \in S. [: \text{sqResidue } S \text{ P } j:] * \text{muPole } j)$
define *X* **where** $X = (\sum e \leq \text{degree } Q. \text{Polynomial.coeff } Q \text{ e} * (\text{muMono } e -$
 $\text{cCorr } p \text{ e} /$
 $\text{of-nat } p))$
define *Y* **where** $Y = (\sum e \leq \text{degree } Q. \text{Polynomial.coeff } Q \text{ e} * \text{cCorr } p \text{ e}) /$
 $\text{of-nat } p$
have *e*: $\text{muPoly } Q = X + Y$
by (*simp add*: *muPoly-def X-def Y-def right-diff-distrib sum-subtractf sum-divide-distrib*)
have $\text{muX } S \text{ P} = [:X + Y:] + Sg$ **by** (*simp only*: *muX-def Q-def[symmetric]*
Sg-def[symmetric] e)
also have $\dots = ([:X:] + Sg) + [:Y:]$ **by** (*simp add*: *add-ac*)
also have $[:X:] + Sg = \text{muXmod } p \text{ S P}$ **by** (*simp add*: *muXmod-def Q-def Sg-def*
X-def)
finally show *?thesis* **by** (*simp add*: *Y-def Q-def*)
qed

lemma *pX-plus-const-pCons*: $pX + [:c:] = [:c, 1:]$
by (*simp add*: *monom-Suc*)

lemma *vpGge-mul0'*: $\text{prime } p \implies \text{vpGge } p \text{ A } 0 \implies$
 $\text{vpGge } p \text{ B } 0 \implies \text{vpGge } p \text{ (A * B) } 0$
using *vpGge-mul[of p A 0 B 0]* **by** *simp*

lemma *vpGge-pow0'*: $\text{prime } p \implies \text{vpGge } p \text{ A } 0 \implies \text{vpGge } p \text{ (A } ^n \text{) } 0$
by (*induction n*) (*auto simp*: *vpGge-one vpGge-mul0'*)

lemma *vpGge-pX-plus-sq0*: $\text{prime } p \implies \text{vpGge } p \text{ (pX} + [: \text{of-nat } n ^2:] \text{) } 0$
proof –
assume *p*: *prime p*
show *?thesis* **by** (*rule* *vpGge-add[OF p vpGge-pX vpGge-C[OF vge-pow[OF p*
vge-of-nat[OF p]]]])
qed

```

lemma lead-coeff-sqPoleProd0: lead-coeff (sqPoleProd S) = 1
  unfolding sqPoleProd-def lead-coeff-prod pX-plus-const-pCons by simp

lemma vpGge-sqPoleProd0: prime p  $\implies$  vpGge p (sqPoleProd S) 0
proof –
  assume p: prime p
  show ?thesis
  proof (cases finite S)
    case True
    have vpGge p ( $\prod j \in S. pX + [:of-nat j \wedge 2:]$ ) ( $\sum j \in S. 0$ )
      by (rule vpGge-prod[OF p True]) (rule vpGge-pX-plus-sq0[OF p])
    thus ?thesis by (simp add: sqPoleProd-def)
  next
    case False
    thus ?thesis by (simp add: sqPoleProd-def vpGge-one)
  qed
qed

lemma vpGge-Dm0: prime p  $\implies$  vpGge p (Dm m) 0
proof –
  assume p: prime p
  have Dm m = sqPoleProd {1..m} by (simp add: Dm-def sqPoleProd-def)
  thus ?thesis using vpGge-sqPoleProd0[OF p] by simp
qed

lemma degree-div-sub:
  fixes P Q :: rat poly
  assumes Q  $\neq$  0
  shows degree (P div Q)  $\leq$  degree P – degree Q
proof (cases P div Q = 0)
  case True
    thus ?thesis by simp
  next
    case False
    have dm: degree (Q * (P div Q)) = degree Q + degree (P div Q) using assms
    False
      by (rule degree-mult-eq)
    have P mod Q = 0  $\vee$  degree (P mod Q) < degree (Q * (P div Q))
      using degree-mod-less[OF assms, of P] dm by auto
    hence degree (Q * (P div Q) + P mod Q) = degree (Q * (P div Q))
    proof
      assume P mod Q = 0
      thus ?thesis by (simp only: add-0-right)
    next
      assume degree (P mod Q) < degree (Q * (P div Q))
      thus ?thesis by (rule degree-add-eq-left)
    qed
    hence degree P = degree Q + degree (P div Q) using dm by simp
    thus ?thesis by simp
  qed

```

```

lemma blocks-lex:
  fixes  $f :: \text{nat} \Rightarrow \text{nat}$ 
  defines  $B\ n \equiv \text{concat } (\text{map } (\lambda a. \text{map } (\lambda i. (a, i)) [0..<f\ a]) [0..<n])$ 
  shows  $y < x \implies x < \text{length } (B\ n) \implies$ 
     $\text{fst } (B\ n\ !\ y) < \text{fst } (B\ n\ !\ x) \vee (\text{fst } (B\ n\ !\ y) = \text{fst } (B\ n\ !\ x) \wedge$ 
     $\text{snd } (B\ n\ !\ y) < \text{snd } (B\ n\ !\ x))$ 
proof (induction n arbitrary: x y)
  case 0
  thus ?case by (simp add: B-def)
next
  case (Suc n)
  define  $C$  where  $C = \text{map } (\lambda i. (n, i)) [0..<f\ n]$ 
  have  $BS: B\ (Suc\ n) = B\ n\ @\ C$  by (simp add: B-def C-def)
  have  $\text{mem}: \text{fst } z < n$  if  $z \in \text{set } (B\ n)$  for  $z$  using that by (auto simp: B-def)
  have  $Cn: C\ !\ k = (n, k)$  if  $k < \text{length } C$  for  $k$  using that by (simp add: C-def)
  show ?case
  proof (cases x < length (B n))
  case True
  thus ?thesis using  $Suc.IH[OF\ Suc.prem\ s(1)\ True]\ Suc.prem\ s(1)$  by (simp
add: BS nth-append)
  next
  case False
  have  $xC: x - \text{length } (B\ n) < \text{length } C$  using  $Suc.prem\ s(2)\ False$  by (simp
add: BS)
  show ?thesis
  proof (cases y < length (B n))
  case True
  have  $\text{fst } (B\ n\ !\ y) < n$  using  $\text{mem}[OF\ nth\ mem[OF\ True]]$  .
  thus ?thesis using  $True\ False\ xC$  by (simp add: BS nth-append Cn)
  next
  case False2: False
  have  $y - \text{length } (B\ n) < x - \text{length } (B\ n)$  using  $Suc.prem\ s(1)\ False2$  by
simp
  thus ?thesis using  $False\ False2\ xC\ Suc.prem\ s$  by (simp add: BS nth-append
Cn)
  qed
  qed
  qed

```

The polynomial quotients $Q\{-ij\} = D\text{-}N^6\ t^{\{i+j\}} \text{ div } D\text{-}K$ are integral.

```

lemma vpGge-quot:  $\text{prime } p \implies$ 
   $\text{vpGge } p\ ((Dm\ (Nof\ K) \wedge^6 * pX \wedge n) \text{ div } sqPoleProd\ \{1..K\})\ 0$ 
proof -
  assume  $p: \text{prime } p$ 
  have  $\text{num}: \text{vpGge } p\ (Dm\ (Nof\ K) \wedge^6 * pX \wedge n)\ 0$ 
  by (rule vpGge-mul0'[OF p vpGge-pow0'[OF p vpGge-Dm0[OF p]] vpGge-pow0'[OF
p vpGge-pX]])
  show ?thesis by (rule vpGge-div-monic[OF p num lead-coeff-sqPoleProd0 vpGge-sqPoleProd0[OF

```


$p]]$)
qed

A matrix whose rows outside a set of k indices vanish has rank $\leq k$.

lemma *rank-le-card-nonzero-rows*:

assumes $L \in \text{carrier-mat } n \ n \ T \subseteq \{0..<n\} \wedge i \ j. \ i < n \implies$

$i \notin T \implies j < n \implies L \ \$\$ (i, j) = 0$

shows $\text{vec-space.rank } n \ L \leq \text{card } T$

proof –

have finT : *finite* T **using** *finite-subset*[*OF* *assms*(2)] **by** *blast*

define xs **where** $xs = \text{sorted-list-of-set } T$

define k **where** $k = \text{card } T$

have dxs : *distinct* xs **unfolding** $xs\text{-def}$ **by** *auto*

have sxs : *set* $xs = T$ **unfolding** $xs\text{-def}$ **using** finT **by** *force*

have lxs : *length* $xs = k$ **unfolding** $xs\text{-def}$ $k\text{-def}$ **using** finT **by** *force*

define $E :: 'a \text{ mat}$ **where** $E = \text{Matrix.mat } n \ k \ (\lambda(i, t). \text{ if } xs ! t = i \text{ then } 1 \text{ else } 0)$

define G **where** $G = \text{Matrix.mat } k \ n \ (\lambda(t, j). \ L \ \$\$ (xs ! t, j))$

have Ec : $E \in \text{carrier-mat } n \ k$ **unfolding** $E\text{-def}$ **by** *force*

have Gc : $G \in \text{carrier-mat } k \ n$ **unfolding** $G\text{-def}$ **by** *simp*

have EG : $E * G = L$

proof (*rule eq-matI*)

fix $i \ j$ **assume** $i: i < \text{dim-row } L$ **and** $j: j < \text{dim-col } L$

have i' : $i < n$ **and** j' : $j < n$ **using** $i \ j$ *assms*(1) **by** *auto*

have e : $(E * G) \ \$\$ (i, j) = (\sum t = 0..<k. (\text{if } xs ! t = i \text{ then } 1 \text{ else } 0) * L \ \$\$ (xs ! t, j))$

using $i' \ j' \ Ec \ Gc$ **unfolding** $E\text{-def}$ $G\text{-def}$ **by** (*auto simp: scalar-prod-def index-mult-mat*)

show $(E * G) \ \$\$ (i, j) = L \ \$\$ (i, j)$

proof (*cases* $i \in T$)

case *True*

then obtain $t0$ **where** $t0: t0 < k \ xs ! t0 = i$ **using** $sxs \ lxs$ **by** (*meson in-set-conv-nth*)

have u : $xs ! t = i \iff t = t0$ **if** $t < k$ **for** t

using *that* $t0(1) \ t0(2)$ [*symmetric*] $dxs \ lxs$ **by** (*auto simp: nth-eq-iff-index-eq*)

have $(\sum t = 0..<k. (\text{if } xs ! t = i \text{ then } 1 \text{ else } 0) * L \ \$\$ (xs ! t, j)) = (\sum t = 0..<k. \text{ if } t = t0 \text{ then } L \ \$\$ (i, j) \text{ else } 0)$

proof (*rule sum.cong*[*OF refl*])

fix t **assume** $t: t \in \{0..<k\}$

show $(\text{if } xs ! t = i \text{ then } 1 \text{ else } 0) * L \ \$\$ (xs ! t, j) = (\text{if } t = t0 \text{ then } L \ \$\$ (i, j) \text{ else } 0)$

using $u[\text{of } t] \ t \ t0$ **by** *auto*

qed

also have $\dots = L \ \$\$ (i, j)$ **using** $t0$ **by** *fastforce*

finally show *?thesis* **using** e **by** *argo*

next

case *False*

have $xs ! t \neq i$ **if** $t < k$ **for** t **using** *that False* $sxs \ lxs$ **by** *force*

hence $(\sum t = 0..<k. (\text{if } xs ! t = i \text{ then } 1 \text{ else } 0) * L \ \$\$ (xs ! t, j)) = 0$ **by**

```

force
  thus ?thesis using e assms(3)[OF i' False j'] by argo
qed
next
  show dim-row (E * G) = dim-row L using Ec Gc assms(1) by fastforce
  show dim-col (E * G) = dim-col L using Ec Gc assms(1) by fastforce
qed
  have vec-space.rank n L = vec-space.rank n (E * G) using EG by meson
  also have ... ≤ vec-space.rank n E by (rule vec-space.rank-mat-mul-right[OF Ec Gc])
  also have ... ≤ k by (rule vec-space.rank-le-nc[OF Ec])
  finally show ?thesis unfolding k-def .
qed

```

(4.10): $G \cdot K = A + p \wedge \neg 1L$ with L constant, integral, of rank $\leq r \cdot p$ (rank over $\mathbb{Q}$, *vec-space.rank* of *Jordan_Normal_Form*).

theorem *GK-outer-decomposition*:

```

assumes p: prime p and K: 40 dvd K and H: OuterHyp K p
shows ∃ L :: rat mat. L ∈ carrier-mat (hof K) (hof K) ∧
      GK K = Aout K p + map-mat (λc. [:c / of-nat p:]) L ∧
      (∀ i < hof K. ∀ j < hof K. L $$ (i, j) ≠ 0 ⟶
        0 ≤ vp p (L $$ (i, j))) ∧
      vec-space.rank (hof K) L ≤ rOut K p
proof -
  define h where h = hof K
  define Q where Q = (λi j. (Dm (Nof K) ^ 6 * pX ^ (i + j)) div sqPoleProd {1..K})
  define L where L = Matrix.mat h h (λ(i, j). ∑ e ≤ degree (Q i j). Polynomial.coeff (Q i j) e * cCorr p e)
  obtain n where Kn: K = 40 * n using K by (elim dvdE)
  have hN: Nof K = 3 * n and hh: h = 37 * n by (simp-all add: Kn Nof-def hof-def h-def)
  have Lc: L ∈ carrier-mat h h by (simp add: L-def)
  have GL: GK K = Aout K p + map-mat (λc. [:c / of-nat p:]) L
  proof (rule eq-matI)
    fix i j assume i: i < dim-row (Aout K p + map-mat (λc. [:c / of-nat p:]) L)
      and j: j < dim-col (Aout K p + map-mat (λc. [:c / of-nat p:]) L)
    show GK K $$ (i, j) = (Aout K p + map-mat (λc. [:c / of-nat p:]) L) $$ (i, j)
      using i j by (simp add: GK-def Aout-def L-def Q-def h-def muX-eq-muXmod-add[OF p])
  qed (simp-all add: GK-def Aout-def L-def h-def)
  have Lint: ⋀ i j. i < h ⟹ j < h ⟹
    L $$ (i, j) ≠ 0 ⟹ 0 ≤ vp p (L $$ (i, j))
  proof -
    fix i j assume i: i < h and j: j < h and nz: L $$ (i, j) ≠ 0
    have Qi: vpGge p (Q i j) 0 unfolding Q-def by (rule vpGge-quot[OF p])
    have vge p (∑ e ≤ degree (Q i j). Polynomial.coeff (Q i j) e * cCorr p e) 0
    proof (rule vge-sum[OF p])
      fix e

```

```

    show vge p (Polynomial.coeff (Q i j) e * cCorr p e) 0
    using vge-mul[OF p Qi[unfolded vpGge-iff, rule-format] cCorr-vge[OF p]]
  by simp
  qed simp
  hence vge p (L $$ (i, j)) 0 using i j by (simp add: L-def)
  thus 0 ≤ vp p (L $$ (i, j)) using nz by (simp add: vge-def)
  qed
  define m0 where m0 = 2 * p - 2 - 5 * (3 * n)
  have zero-rows:  $\bigwedge i j. i < h \implies i < m0 \implies j < h \implies L \text{ } \$\$ (i, j) = 0$ 
  proof -
    fix i j assume i:  $i < h$  and im:  $i < m0$  and j:  $j < h$ 
    have p7:  $7 \leq p$  and N5:  $5 * \text{Nof } K \leq 2 * p - 2$  using H by (simp-all add:
  OuterHyp-def)
    have dS:  $\text{degree } (\text{sqPoleProd } \{1..K\}) = K$ 
    proof -
      have  $\text{sqPoleProd } \{1..K\} = Dm \text{ } K$  by (simp add: Dm-def sqPoleProd-def)
      thus ?thesis by (simp add: degree-Dm)
    qed
    have dN:  $\text{degree } (Dm (\text{Nof } K) \wedge 6 * pX \wedge (i + j)) \leq 6 * \text{Nof } K + (i + j)$ 
    proof -
      have  $\text{degree } (Dm (\text{Nof } K) \wedge 6 * pX \wedge (i + j)) \leq$ 
         $\text{degree } (Dm (\text{Nof } K) \wedge 6) + \text{degree } (pX \wedge (i + j))$  by (rule degree-mult-le)
      also have  $\text{degree } (Dm (\text{Nof } K) \wedge 6) \leq \text{degree } (Dm (\text{Nof } K)) * 6$  by (rule
  degree-power-le)
      also have  $\text{degree } ((pX :: \text{rat poly}) \wedge (i + j)) \leq \text{degree } (pX :: \text{rat poly}) * (i +$ 
  j)
      by (rule degree-power-le)
    finally show ?thesis by (simp add: degree-Dm degree-monom-eq)
    qed
    have dq:  $\text{degree } (Q i j) \leq \text{degree } (Dm (\text{Nof } K) \wedge 6 * pX \wedge (i + j)) -$ 
       $\text{degree } (\text{sqPoleProd } \{1..K\})$ 
    unfolding Q-def by (rule degree-div-sub[OF sqPoleProd-nonzero]) simp
    have  $\text{degree } (Q i j) < 2 * p - 3$  using dq dN dS im j hN hh Kn p7 N5 by
  (simp add: m0-def h-def)
    hence z:  $\bigwedge e. e \in \{\dots \text{degree } (Q i j)\} \implies$ 
       $\text{Polynomial.coeff } (Q i j) e * cCorr p e = 0$ 
    by (auto simp: cCorr-of-lt)
    have  $L \text{ } \$\$ (i, j) = (\sum e \leq \text{degree } (Q i j). \text{Polynomial.coeff } (Q i j) e * cCorr$ 
  p e)
    using i j by (simp add: L-def)
    also have  $\dots = 0$  by (rule sum.neutral) (use z in blast)
    finally show  $L \text{ } \$\$ (i, j) = 0$  .
  qed
  have rank:  $\text{vec-space.rank } h \text{ } L \leq \text{rOut } K \text{ } p$ 
  proof -
    have N5:  $5 * \text{Nof } K \leq 2 * p - 2$  and p7:  $7 \leq p$  using H by (simp-all add:
  OuterHyp-def)
    have  $\text{vec-space.rank } h \text{ } L \leq \text{card } \{m0..<h\}$ 

```

```

      by (rule rank-le-card-nonzero-rows[OF Lc]) (auto intro: zero-rows)
    also have card {m0.. $h$ }  $\leq$  rOut K p using N5 p7 hN hh Kn by (simp add:
rOut-def m0-def)
    finally show ?thesis .
  qed
  show ?thesis unfolding h-def[symmetric] using Lc GL Lint rank by blast
qed

```

lemma *mod-add-eq-0-iff*:

```

  fixes j a p :: nat
  assumes 0 < a a < p
  shows (j + a) mod p = 0  $\longleftrightarrow$  j mod p = p - a
proof -
  from assms have 0 < p by simp
  hence r: j mod p < p by simp
  have (j + a) mod p = (j mod p + a) mod p by (simp add: mod-add-left-eq)
  also have ... = 0  $\longleftrightarrow$  j mod p + a = p
proof
  assume h: (j mod p + a) mod p = 0
  have j mod p + a < 2 * p using r assms by linarith
  moreover have p dvd j mod p + a using h by (simp add: mod-eq-0-iff-dvd)
  then obtain t where t: j mod p + a = p * t by (elim dvdE)
  ultimately have t < 2 by simp
  moreover have t  $\neq$  0
proof
  assume t = 0
  hence j mod p + a = 0 using t by simp
  thus False using assms by simp
qed
  ultimately show j mod p + a = p using t by (cases t) auto
qed simp
finally show ?thesis using assms by linarith
qed

```

lemma *sum-classes-card*:

```

  assumes prime p 2 < p finite X
  shows ( $\sum$  a  $\leq$  mHalf p. card {j  $\in$  X. j mod p = a mod p  $\vee$ 
(j + a) mod p = 0}) = card X
proof -
  define m where m = mHalf p
  have p0: 0 < p using assms by simp
  from assms have odd p by (metis prime-odd-nat)
  hence m2: 2 * m = p - 1 unfolding m-def mHalf-def by (auto elim!: oddE)
  define C where C a = {j  $\in$  X. j mod p = a  $\vee$  j mod p = p - a} for a
  have Ceq: {j  $\in$  X. j mod p = a mod p  $\vee$  (j + a) mod p = 0} = C a if a: a  $\in$ 
{..m}
  for a
proof (cases a = 0)
  case True

```

```

    have  $j \bmod p \neq p$  for  $j$  using  $p0$  by (metis mod-less-divisor less-irrefl)
    thus ?thesis using True by (auto simp: C-def)
next
  case False
  hence  $ap: 0 < a < p$  using  $a m2$  by auto
  thus ?thesis using mod-add-eq-0-iff[OF ap] by (auto simp: C-def)
qed
have  $(\sum a \leq m. \text{card } \{j \in X. j \bmod p = a \bmod p \vee (j + a) \bmod p = 0\}) = (\sum a \leq m. \text{card } (C a))$ 
proof (rule sum.cong)
  fix  $a$  assume  $a: a \in \{..m\}$ 
  show  $\text{card } \{j \in X. j \bmod p = a \bmod p \vee (j + a) \bmod p = 0\} = \text{card } (C a)$ 
    unfolding Ceq[OF a] ..
qed simp
also have  $\dots = \text{card } (\bigcup a \in \{..m\}. C a)$ 
proof (rule card-UN-disjoint[symmetric])
  show  $\forall i \in \{..m\}. \forall j \in \{..m\}. i \neq j \longrightarrow C i \cap C j = \{\}$ 
    using  $m2$  by (auto simp: C-def)
qed (use assms(3) in <auto simp: C-def>)
also have  $(\bigcup a \in \{..m\}. C a) = X$ 
proof (intro equalityI subsetI)
  fix  $j$  assume  $j \in (\bigcup a \in \{..m\}. C a)$ 
  thus  $j \in X$  by (auto simp: C-def)
next
  fix  $j$  assume  $j: j \in X$ 
  have  $rp: j \bmod p < p$  using  $p0$  by simp
  show  $j \in (\bigcup a \in \{..m\}. C a)$ 
  proof (cases  $j \bmod p \leq m$ )
    case True
    thus ?thesis using  $j$  by (auto simp: C-def intro!: bexI[of -  $j \bmod p$ ])
  next
    case False
    hence  $p - j \bmod p \in \{..m\}$  using  $rp m2$  by auto
    moreover have  $j \in C (p - j \bmod p)$  using  $j rp$  by (auto simp: C-def)
    ultimately show ?thesis by blast
  qed
qed
finally show ?thesis by (simp add: m-def)
qed

lemma length-outer-idx: prime  $p \implies 40 \text{ dvd } K \implies \text{OuterHyp } K p \implies \text{length } (\text{outer-idx } K p) = \text{hof } K$ 
proof -
  assume  $p$ : prime  $p$  and  $K$ :  $40 \text{ dvd } K$  and  $H$ : OuterHyp  $K p$ 
  have  $p2$ :  $2 < p$  using  $H$  by (simp add: OuterHyp-def)
  obtain  $k$  where  $k$ :  $K = 40 * k$  using  $K$  by (elim dvdE)
  have  $hk$ :  $\text{hof } K = 37 * k$  and  $Nk$ :  $\text{Nof } K = 3 * k$  unfolding  $k$  hof-def Nof-def
  by simp-all

```

have $\text{length } (\text{outer-idx } K \ p) = (\sum \ a \in \text{set } [0..<mHalf \ p + 1]. \text{outerCount } K \ p$
 $a)$
unfolding $\text{outer-idx-def length-concat map-map}$
by $(\text{simp add: o-def sum-list-distinct-conv-sum-set})$
also have $\text{set } [0..<mHalf \ p + 1] = \{..mHalf \ p\}$ **by** auto
also have $(\sum \ a \leq mHalf \ p. \text{outerCount } K \ p \ a) = \text{card } \{Nof \ K + 1..K\}$
unfolding $\text{outerCount-def outerClassPoles-def}$ **by** $(\text{rule sum-classes-card}[OF \ p$
 $p2]) \text{ simp}$
also have $\dots = \text{hof } K$ **using** $hk \ Nk \ k$ **by** simp
finally show $?thesis$.
qed

(4.11): the outer basis is $\mathbb{Z}$ - p -unimodular.

lemma outer-idx-lex :

assumes $y < x \ x < \text{length } (\text{outer-idx } K \ p)$
shows $\text{fst } (\text{outer-idx } K \ p \ ! \ y) < \text{fst } (\text{outer-idx } K \ p \ ! \ x) \vee$
 $(\text{fst } (\text{outer-idx } K \ p \ ! \ y) = \text{fst } (\text{outer-idx } K \ p \ ! \ x) \wedge$
 $\text{snd } (\text{outer-idx } K \ p \ ! \ y) < \text{snd } (\text{outer-idx } K \ p \ ! \ x))$
using blocks-lex **where** $f = \text{outerCount } K \ p$ **and** $n =$
 $mHalf \ p + 1, \ OF \ \text{assms}[\text{unfolded outer-idx-def}]$
unfolding outer-idx-def .

lemma mem-outer-idx : $x \in \text{set } (\text{outer-idx } K \ p) \longleftrightarrow \text{fst } x \leq mHalf \ p \wedge$
 $\text{snd } x < \text{outerCount } K \ p \ (\text{fst } x)$
by $(\text{cases } x) \ (\text{auto simp: outer-idx-def image-iff simp del: upt-Suc})$

lemma vpGge-outerBasis : $\text{prime } p \implies \text{vpGge } p \ (\text{outerBasis } K \ p \ x) \ 0$

proof –

assume p : $\text{prime } p$
have a : $\text{vpGge } p \ (\text{sqPoleProd } S * (pX + [:of\text{-nat } n \wedge 2:]) \wedge i) \ 0$ **for** $S \ n \ i$
by $(\text{rule vpGge-mul0}[OF \ p \ \text{vpGge-sqPoleProd0}[OF \ p] \ \text{vpGge-pow0}[OF \ p \ \text{vpGge-pX-plus-sq0}[OF$
 $p]])$
have b : $\text{vpGge } p \ (\text{sqPoleProd } S * \text{sqPoleProd } T * (pX + [:of\text{-nat } n \wedge 2:]) \wedge i) \ 0$
for $S \ T \ n \ i$
by $(\text{rule vpGge-mul0}[OF \ p \ \text{vpGge-mul0}[OF \ p \ \text{vpGge-sqPoleProd0}[OF \ p] \ \text{vpGge-sqPoleProd0}[OF$
 $p]] \ \text{vpGge-pow0}[OF \ p \ \text{vpGge-pX-plus-sq0}[OF \ p]])$
show $?thesis$ **unfolding** $\text{outerBasis-def Let-def}$ **by** $(\text{simp only: } a \ b \ \text{if-cancel split:}$
 $\text{if-split}) \text{ simp}$
qed

In the outer range every class $a \geq 1$ contains exactly the two poles $a, p - a$ below p , so $\#\{j \in C\text{-}a. \ p < j\} + 2 = \text{ell } p \ K \ a$ (Lean: card-E-add-two).

lemma res-class0 :

assumes $a \leq mHalf \ p \ 2 * mHalf \ p + 1 = p \ j \bmod p = a \bmod p \vee (j + a) \bmod$
 $p = 0$
shows $j \bmod p = a \vee (1 \leq a \wedge j \bmod p = p - a)$
proof $(\text{cases } a = 0)$
case True

```

    thus ?thesis using assms(3) by auto
next
  case False
  hence ap:  $0 < a < p$  using assms(1,2) by auto
  thus ?thesis using assms(3) mod-add-eq-0-iff[OF ap, of j] by auto
qed
lemma card-class-high-add-two:
  assumes prime p OuterHyp K  $p \mid 1 \leq a \leq mHalf\ p$ 
  shows card  $\{j \in \text{outerClassPoles } K\ p\ a.\ p < j\} + 2 = \text{ell } p\ K\ a$ 
proof -
  have h:  $7 \leq p \leq K\ 2 * \text{Nof } K < p$  using assms(2) by (auto simp: Outer-
Hyp-def)
  have odd p using prime-odd-nat[OF assms(1)] h(1) by simp
  then obtain t where  $t: p = 2 * t + 1$  by (elim oddE)
  have m:  $2 * mHalf\ p + 1 = p$  by (simp add: mHalf-def t)
  have ap:  $a < p \mid 2 * a < p$  using m assms(4) by linarith+
  have ane:  $a \neq p - a$  using ap by linarith
  define P where  $P\ j \longleftrightarrow j \bmod p = a \bmod p \vee (j + a) \bmod p = 0$  for j
  define X where  $X = \{j \in \text{outerClassPoles } K\ p\ a.\ p < j\}$ 
  have E:  $\{j \in \{1..K\}.\ P\ j\} = \{a, p - a\} \cup X$ 
  proof (intro equalityI subsetI)
    fix j assume j:  $j \in \{j \in \{1..K\}.\ P\ j\}$ 
    have r:  $j \bmod p = a \vee (1 \leq a \wedge j \bmod p = p - a)$ 
      by (rule res-class0[OF assms(4) m]) (use j in <simp add: P-def>)
    show  $j \in \{a, p - a\} \cup X$ 
  proof (cases j < p)
    case True
    thus ?thesis using r by auto
  next
    case False
    have j  $\neq p$ 
  proof
    assume j = p
    hence  $j \bmod p = 0$  by simp
    thus False using r ap assms(3) by auto
  qed
  hence  $p < j$  using False by simp
  thus ?thesis using j h by (auto simp: X-def outerClassPoles-def P-def)
  qed
next
  fix j assume j  $\in \{a, p - a\} \cup X$ 
  thus  $j \in \{j \in \{1..K\}.\ P\ j\}$  using assms(3) ap h
    by (auto simp: X-def outerClassPoles-def P-def)
  qed
  have fin: finite X by (simp add: X-def outerClassPoles-def)
  have  $\text{ell } p\ K\ a = \text{card } (\{a, p - a\} \cup X)$  unfolding ell-def P-def[symmetric] E
  ..
  also have  $\dots = \text{card } \{a, p - a\} + \text{card } X$ 
    by (rule card-Un-disjoint) (use ap fin in <auto simp: X-def>)

```

also have $\text{card } \{a, p - a\} = 2$ using *ane* by *simp*
 finally show *?thesis* unfolding *X-def* by *simp*
 qed

Modulo p , a pole $j \in C\text{-}a$ with $p < j$ satisfies $j^2 \equiv a^2$: the factor $t + j^2$ is $(t + a^2) + (j^2 - a^2)$ with $p \mid j^2 - a^2$.

lemma *class-sq-cong*:

assumes *prime* p $1 \leq a \leq m\text{Half } p$ $j \in \text{outerClassPoles } K$ $p \nmid a$
 shows $\text{vge } p (\text{of-nat } j^2 - \text{of-nat } a^2) \ 1$

proof –

have p : *prime* p using *assms*(1) .

have j : $j \bmod p = a \bmod p \vee (j + a) \bmod p = 0$ using *assms*(4)
 by (*simp* add: *outerClassPoles-def*)

have $\text{int } p \text{ dvd } (\text{int } j - \text{int } a) * (\text{int } j + \text{int } a)$

proof (*cases* $j \bmod p = a \bmod p$)

case *True*

hence $\text{int } j \bmod \text{int } p = \text{int } a \bmod \text{int } p$ by (*metis* *of-nat-mod*)

hence $\text{int } p \text{ dvd } \text{int } j - \text{int } a$ by (*simp* add: *mod-eq-dvd-iff*)

thus *?thesis* by *simp*

next

case *False*

hence $(j + a) \bmod p = 0$ using j by *simp*

hence $p \text{ dvd } j + a$ by (*rule* *mod-0-imp-dvd*)

hence $\text{int } p \text{ dvd } \text{int } j + \text{int } a$ by (*simp* *flip*: *of-nat-add*)

thus *?thesis* by *simp*

qed

then obtain w where w : $(\text{int } j - \text{int } a) * (\text{int } j + \text{int } a) = \text{int } p * w$ by (*elim* *dvdE*)

have e : $(\text{of-nat } j^2 - \text{of-nat } a^2 :: \text{rat}) = \text{of-nat } p * \text{of-int } w$

proof –

have $(\text{of-nat } j^2 - \text{of-nat } a^2 :: \text{rat}) = \text{of-int } ((\text{int } j - \text{int } a) * (\text{int } j + \text{int } a))$

by (*simp* add: *power2-eq-square algebra-simps*)

thus *?thesis* unfolding w by *simp*

qed

have $\text{vge } p (\text{of-nat } p * \text{of-int } w) (1 + 0)$

by (*rule* *vge-mul*[*OF* p *vge-p-self*[*OF* p] *vge-of-int*[*OF* p]])

thus *?thesis* unfolding e by *simp*

qed

22.2 The modified functional $\mu^0\text{-}X$ and the structure of the outer basis

definition *tailPoles* :: $\text{nat} \Rightarrow \text{nat set}$ where *tailPoles* $K = \{\text{Nof } K + 1..K\}$

μ^0 on polynomials.

definition *muModPoly* :: $\text{nat} \Rightarrow \text{rat poly} \Rightarrow \text{rat}$ where

$\text{muModPoly } p \ Q = (\sum e \leq \text{degree } Q. \text{Polynomial.coeff } Q \ e * (\text{muMono } e - \text{cCorr } p \ e) /$

$of\text{-}nat\ p))$

lemma $muXmod\text{-}eq$:

$muXmod\ p\ S\ P = [:muModPoly\ p\ (P\ div\ sqPoleProd\ S):] + (\sum\ j \in S. [:sqResidue\ S\ P\ j:] * muPole\ j)$
by ($simp\ add$: $muXmod\text{-}def\ muModPoly\text{-}def$)

lemma $muModPoly\text{-}vge$: $prime\ p \implies \gamma \leq p \implies$

$vpGge\ p\ Q\ 0 \implies vge\ p\ (muModPoly\ p\ Q)\ 0$

proof –

assume p : $prime\ p$ **and** $p\gamma$: $\gamma \leq p$ **and** Q : $vpGge\ p\ Q\ 0$

show $?thesis\ unfolding\ muModPoly\text{-}def$

proof ($rule\ vge\text{-}sum[OF\ p]$)

fix e

have $vge\ p\ (Polynomial.coeff\ Q\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))\ (0 + 0)$

by ($rule\ vge\text{-}mul[OF\ p\ Q[unfolded\ vpGge\text{-}iff,\ rule\text{-}format]\ muMod\text{-}vge[OF\ p\ p\gamma]]$)

thus $vge\ p\ (Polynomial.coeff\ Q\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))\ 0$ **by** $simp$

qed $simp$

qed

lemma $muModPoly\text{-}sub\text{-}muPoly$:

$muModPoly\ p\ Q = muPoly\ Q - (\sum\ e \leq degree\ Q. Polynomial.coeff\ Q\ e * (cCorr\ p\ e / of\text{-}nat\ p))$
by ($simp\ add$: $muModPoly\text{-}def\ muPoly\text{-}def\ right\text{-}diff\text{-}distrib\ sum\text{-}subtractf$)

lemma $muXmod\text{-}as\text{-}muX$:

$muXmod\ p\ S\ P = muX\ S\ P - [: \sum\ e \leq degree\ (P\ div\ sqPoleProd\ S). Polynomial.coeff\ (P\ div\ sqPoleProd\ S)\ e * (cCorr\ p\ e / of\text{-}nat\ p):]$

proof –

define Q **where** $Q = P\ div\ sqPoleProd\ S$

define Sg **where** $Sg = (\sum\ j \in S. [:sqResidue\ S\ P\ j:] * muPole\ j)$

define Y **where** $Y = (\sum\ e \leq degree\ Q. Polynomial.coeff\ Q\ e * (cCorr\ p\ e / of\text{-}nat\ p))$

have $muXmod\ p\ S\ P = [:muPoly\ Q - Y:] + Sg$

by ($simp\ add$: $muXmod\text{-}eq\ muModPoly\text{-}sub\text{-}muPoly\ Q\text{-}def\ Sg\text{-}def\ Y\text{-}def$)

also have $\dots = ([:muPoly\ Q:] + Sg) - [:Y:]$ **by** ($simp\ add$: $algebra\text{-}simps$)

also have $[:muPoly\ Q:] + Sg = muX\ S\ P$ **by** ($simp\ add$: $muX\text{-}def\ Q\text{-}def\ Sg\text{-}def$)

finally show $?thesis$ **by** ($simp\ add$: $Y\text{-}def\ Q\text{-}def$)

qed

lemma $lead\text{-}coeff\text{-}sqPoleProd$: $finite\ S \implies lead\text{-}coeff\ (sqPoleProd\ S) = 1$

by ($rule\ lead\text{-}coeff\text{-}sqPoleProd0$)

lemma $vpGge\text{-}sqPoleProd$: $prime\ p \implies vpGge\ p\ (sqPoleProd\ S)\ 0$

by ($rule\ vpGge\text{-}sqPoleProd0$)

```

lemma vpGge-mul0: prime p  $\implies$  vpGge p A 0  $\implies$ 
  vpGge p B 0  $\implies$  vpGge p (A * B) 0
  using vpGge-mul[of p A 0 B 0] by simp

lemma vpGge-pow0: prime p  $\implies$  vpGge p A 0  $\implies$  vpGge p (A ^ n) 0
  by (induction n) (auto simp: vpGge-one vpGge-mul0)

lemma vpGge-pX-plus-sq: prime p  $\implies$  vpGge p (pX + [:of-nat n ^ 2:]) 0
  by (rule vpGge-pX-plus-sq0)

lemma vpGge-Dm: prime p  $\implies$  vpGge p (Dm m) 0
  by (rule vpGge-Dm0)

lemma wOut-nonpos: wOut K p x  $\leq 0$ 
  by (auto simp: wOut-def Let-def)

lemma wOut-halfInt:  $\exists k::\text{int}. 2 * wOut K p x = \text{of-int } k$ 
proof –
  have m:  $2 * \min 0 t = \min 0 (2 * t)$  for t :: rat by (simp add: min-def)
  have mi:  $\min 0 t \in \mathbb{Z}$  if  $t \in \mathbb{Z}$  for t :: rat using that by (simp add: min-def)
  have  $2 * wOut K p x \in \mathbb{Z}$ 
  proof (cases fst x = 0)
    case True
    thus ?thesis by (auto simp: wOut-def Let-def)
  next
    case False
    define l where  $l = \text{ell } p K (\text{fst } x)$ 
    define d where  $d = (\text{if } \text{fst } x \leq \text{Nof } K \text{ then } 1 \text{ else } 0 :: \text{rat})$ 
    have d:  $d \in \mathbb{Z}$  by (simp add: d-def)
    have w:  $wOut K p x = (\text{if } \text{snd } x < l - 2 \text{ then } \min 0 (\text{of-nat } (\text{snd } x) + 3 * d$ 
     $- (\text{of-nat } l + 4) /$ 
     $2) \text{ else } 0)$ 
    using False by (simp add: wOut-def Let-def l-def d-def)
    have e:  $2 * (\text{of-nat } (\text{snd } x) + 3 * d - (\text{of-nat } l + 4) / 2) =$ 
     $2 * \text{of-nat } (\text{snd } x) + 6 * d - \text{of-nat } l - (4 :: \text{rat})$ 
    by (simp add: field-simps)
    have  $2 * \text{of-nat } (\text{snd } x) + 6 * d - \text{of-nat } l - (4 :: \text{rat}) \in \mathbb{Z}$ 
    using d by (intro Ints-diff Ints-add Ints-mult) simp-all
    hence mm:  $\min 0 (2 * (\text{of-nat } (\text{snd } x) + 3 * d - (\text{of-nat } l + 4) / 2)) \in \mathbb{Z}$ 
unfolding e
    by (rule mi)
    show ?thesis
    proof (cases snd x < l - 2)
      case True
      have  $2 * wOut K p x = \min 0 (2 * (\text{of-nat } (\text{snd } x) + 3 * d - (\text{of-nat } l +$ 
       $4) / 2))$ 
      by (simp only: w if-P[OF True] m)
      thus ?thesis using mm by simp

```

```

next
  case False
  thus ?thesis by (simp add: w)
qed
qed
thus ?thesis by (auto elim: Ints-cases)
qed

```

μ^0 - X also depends only on the rational function.

lemma *muXmod-mul-cancel-set*:

```

finite T  $\implies$  finite S  $\implies$   $T \cap S = \{\}$   $\implies$ 
  muXmod p ( $T \cup S$ ) ( $P * \text{sqPoleProd } T$ ) = muXmod p S P
proof -
  assume T: finite T and S: finite S and d:  $T \cap S = \{\}$ 
  have u: sqPoleProd ( $T \cup S$ ) = sqPoleProd T * sqPoleProd S using T S d
    by (simp add: sqPoleProd-def prod.union-disjoint)
  have sqPoleProd T * P div (sqPoleProd T * sqPoleProd S) = P div sqPoleProd S
    by (rule div-mult-mult1[OF sqPoleProd-nonzero[OF T]])
  hence q: ( $P * \text{sqPoleProd } T$ ) div sqPoleProd ( $T \cup S$ ) = P div sqPoleProd S
  unfolding u
    by (simp add: mult.commute)
  show ?thesis by (simp only: muXmod-as-muX q muX-mul-cancel-set[OF S T d])
qed

```

lemma *muXmod-empty*: muXmod *p* $\{\}$ *P* = [:muModPoly *p* *P*:]
 by (simp add: muXmod-eq sqPoleProd-def)

lemma *sqPoleProd-union*: finite *s* $\implies$ finite *t* $\implies$
 $s \cap t = \{\} \implies \text{sqPoleProd } (s \cup t) = \text{sqPoleProd } s * \text{sqPoleProd } t$
 by (simp add: sqPoleProd-def prod.union-disjoint)

lemma *sqPoleProd-Icc-split*: sqPoleProd $\{1..K\} = \text{Dm } (\text{Nof } K) * \text{sqPoleProd } (\text{tailPoles } K)$

```

proof -
  have Nof K  $\leq$  K by (simp add: Nof-def)
  hence u:  $\{1..K\} = \{1..\text{Nof } K\} \cup \text{tailPoles } K$  by (auto simp: tailPoles-def)
  have sqPoleProd ( $\{1..\text{Nof } K\} \cup \text{tailPoles } K$ ) =
    sqPoleProd  $\{1..\text{Nof } K\} * \text{sqPoleProd } (\text{tailPoles } K)$ 
    by (rule sqPoleProd-union) (auto simp: tailPoles-def)
  moreover have Dm ( $\text{Nof } K$ ) = sqPoleProd  $\{1..\text{Nof } K\}$  by (simp add: Dm-def sqPoleProd-def)
  ultimately show ?thesis unfolding u by simp
qed

```

The q -part of an outer basis row: $\text{outerBasis} = P \cdot a \cdot oq$.

definition *oq* :: $\text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \times \text{nat} \Rightarrow \text{rat poly}$ **where**

```

oq K p x = (let a = fst x; i = snd x in
  if a = 0 then (pX + [:of-nat p ^ 2:]) ^ i

```

else if $i < \text{ell } p \ K \ a - 2$ then $(pX + [:of\text{-}nat \ a \ ^2:]) \wedge i$
 else $\text{sqPoleProd } \{j \in \text{outerClassPoles } K \ p \ a. \ p <$
 $j\} * (pX + [:of\text{-}nat \ a \ ^2:]) \wedge (i - (\text{ell } p \ K \ a - 2))$

lemma *outerBasis-eq*: $\text{outerBasis } K \ p \ x =$
 $\text{sqPoleProd } (\text{tailPoles } K - \text{outerClassPoles } K \ p \ (\text{fst } x)) * \text{oq } K \ p \ x$
by (*simp add: outerBasis-def oq-def tailPoles-def Let-def mult.assoc*)

lemma *oq-vge*: $\text{prime } p \implies \text{vpGge } p \ (\text{oq } K \ p \ x) \ 0$

proof –

assume p : *prime* p
 have a : $\text{vpGge } p \ ((pX + [:of\text{-}nat \ n \ ^2:]) \wedge i) \ 0$ **for** $n \ i$
by (*rule vpGge-pow0[OF p vpGge-pX-plus-sq[OF p]]*)
 have b : $\text{vpGge } p \ (\text{sqPoleProd } S * (pX + [:of\text{-}nat \ n \ ^2:]) \wedge i) \ 0$ **for** $S \ n \ i$
by (*rule vpGge-mul0[OF p vpGge-sqPoleProd[OF p] a]*)
show ?thesis **unfolding** *oq-def Let-def* **by** (*simp only: a b if-cancel split: if-split*)
simp
qed

lemma *outerGram-apply*:

$x < \text{length } (\text{outer-idx } K \ p) \implies y < \text{length } (\text{outer-idx } K \ p) \implies$
 $\text{outerGram } K \ p \ \$\$ (x, y) = \text{muXmod } p \ \{1..K\} \ (\text{Dm } (\text{Nof } K) \wedge 6 * \text{outerBasis}$
 $K \ p \ (\text{outer-idx } K \ p \ ! \ x) *$
 $\text{outerBasis } K \ p \ (\text{outer-idx } K \ p \ ! \ y))$
by (*simp add: outerGram-def Let-def*)

22.3 Arithmetic of the pole classes

lemma *mem-class*: $j \in \text{outerClassPoles } K \ p \ a \iff \text{Nof } K + 1 \leq j \wedge$
 $j \leq K \wedge (j \bmod p = a \bmod p \vee (j + a) \bmod p = 0)$
by (*auto simp: outerClassPoles-def*)

Facts implied by (4.9).

lemma *outer-facts*:

assumes *prime* p *OuterHyp* $K \ p$
shows $7 \leq p \wedge \text{odd } p \wedge 2 * \text{mHalf } p + 1 = p \wedge p \leq K \wedge K < 3 * p \wedge$
 $2 * \text{Nof } K < p \wedge$
 $(K < 2 * p \longrightarrow K \bmod p = K - p) \wedge (2 * p \leq K \longrightarrow$
 $K \bmod p = K - 2 * p)$

proof –

have h : $7 \leq p \wedge p \leq K \wedge K < 3 * p \wedge 2 * \text{Nof } K < p$ **using** *assms(2)*
by (*auto simp: OuterHyp-def*)
 have o : $\text{odd } p$ **using** *prime-odd-nat[OF assms(1)] h(1)* **by** *simp*
then obtain k **where** k : $p = 2 * k + 1$ **by** (*elim oddE*)
 have m : $2 * \text{mHalf } p + 1 = p$ **by** (*simp add: mHalf-def k*)
 have $k1$: $K < 2 * p \longrightarrow K \bmod p = K - p$
proof
 assume $K < 2 * p$
 hence lt : $K - p < p$ **using** h **by** *linarith*

```

    have  $K \bmod p = (K - p) \bmod p$  using  $h(2)$  by (simp add: le-mod-geq)
    also have  $\dots = K - p$  using  $lt$  by simp
    finally show  $K \bmod p = K - p$  .
  qed
  have  $k2: 2 * p \leq K \longrightarrow K \bmod p = K - 2 * p$ 
  proof
    assume  $a: 2 * p \leq K$ 
    hence  $lt: K - 2 * p < p$  using  $h$  by linarith
    have  $K \bmod p = (K - p) \bmod p$  using  $h(2)$  by (simp add: le-mod-geq)
    also have  $\dots = (K - p - p) \bmod p$  using  $a$  by (simp add: le-mod-geq)
    also have  $K - p - p = K - 2 * p$  by simp
    also have  $(K - 2 * p) \bmod p = K - 2 * p$  using  $lt$  by simp
    finally show  $K \bmod p = K - 2 * p$  .
  qed
  show ?thesis using  $h$   $o$   $m$   $k1$   $k2$  by blast
qed

lemma res-class:
  assumes  $a \leq mHalf\ p\ 2 * mHalf\ p + 1 = p\ j \bmod p = a \bmod p \vee (j + a) \bmod p = 0$ 
  shows  $j \bmod p = a \vee (1 \leq a \wedge j \bmod p = p - a)$ 
  proof (cases  $a = 0$ )
    case True
    thus ?thesis using  $assms(3)$  by auto
  next
    case False
    hence  $ap: 0 < a \wedge a < p$  using  $assms(1,2)$  by auto
    thus ?thesis using  $assms(3)$  mod-add-eq-0-iff[OF  $ap$ , of  $j$ ] by auto
  qed

lemma class-disjoint:
  assumes  $prime\ p$   $OuterHyp\ K\ p\ a \leq mHalf\ p\ c \leq mHalf\ p\ a \neq c$ 
  shows  $outerClassPoles\ K\ p\ a \cap outerClassPoles\ K\ p\ c = \{\}$ 
  proof -
    have  $m: 2 * mHalf\ p + 1 = p$  using  $outer-facts[OF\ assms(1,2)]$  by blast
    { fix  $j$  assume  $ja: j \in outerClassPoles\ K\ p\ a$  and  $jc: j \in outerClassPoles\ K\ p\ c$ 
      have  $j \bmod p = a \vee (1 \leq a \wedge j \bmod p = p - a)$ 
        by (rule res-class[OF  $assms(3)$   $m$ ]) (use  $ja$  in <simp add: mem-class>)
      moreover have  $j \bmod p = c \vee (1 \leq c \wedge j \bmod p = p - c)$ 
        by (rule res-class[OF  $assms(4)$   $m$ ]) (use  $jc$  in <simp add: mem-class>)
      ultimately have  $False$  using  $assms(3,4,5)$   $m$  by (elim disjE conjE; linarith)
    }
    thus ?thesis by blast
  qed

lemma class-small:
  assumes  $prime\ p$   $OuterHyp\ K\ p\ 1 \leq a \wedge a \leq mHalf\ p\ j \in outerClassPoles\ K\ p\ a\ j < p$ 
  shows  $j = a \vee j = p - a$ 
  proof -

```

have $m: 2 * mHalf\ p + 1 = p$ **using** $outer-facts[OF\ assms(1,2)]$ **by** $blast$
have $j\ mod\ p = a \vee (1 \leq a \wedge j\ mod\ p = p - a)$
by $(rule\ res-class[OF\ assms(4)\ m])$ $(use\ assms(5))$ **in** $\langle simp\ add: mem-class \rangle$
thus $?thesis$ **using** $assms(6)$ **by** $auto$
qed

lemma $class-not-dvd$:

assumes $prime\ p\ OuterHyp\ K\ p\ 1 \leq a\ a \leq mHalf\ p\ j \in outerClassPoles\ K\ p\ a$
shows $\neg p\ dvd\ j$
proof
assume $d: p\ dvd\ j$
have $m: 2 * mHalf\ p + 1 = p$ **using** $outer-facts[OF\ assms(1,2)]$ **by** $blast$
have $j\ mod\ p = a \vee (1 \leq a \wedge j\ mod\ p = p - a)$
by $(rule\ res-class[OF\ assms(4)\ m])$ $(use\ assms(5))$ **in** $\langle simp\ add: mem-class \rangle$
moreover **have** $j\ mod\ p = 0$ **using** d **by** $simp$
ultimately **show** $False$ **using** $assms(3,4)\ m$ **by** $auto$
qed

lemma $pa-mem$:

assumes $prime\ p\ OuterHyp\ K\ p\ 1 \leq a\ a \leq mHalf\ p$
shows $p - a \in outerClassPoles\ K\ p\ a$
proof $-$
obtain $m: 2 * mHalf\ p + 1 = p$ **and** $pK: p \leq K$ **and** $N: 2 * Nof\ K < p$
using $outer-facts[OF\ assms(1,2)]$ **by** $blast$
have $Nof\ K + 1 \leq p - a$ **using** $N\ m\ assms(4)$ **by** $linarith$
moreover **have** $p - a \leq K$ **using** pK **by** $simp$
moreover **have** $(p - a + a)\ mod\ p = 0$ **using** $assms(4)\ m$ **by** $simp$
ultimately **show** $?thesis$ **by** $(simp\ add: mem-class)$
qed

lemma $a-mem-iff$:

assumes $prime\ p\ OuterHyp\ K\ p\ 1 \leq a\ a \leq mHalf\ p$
shows $a \in outerClassPoles\ K\ p\ a \longleftrightarrow Nof\ K < a$
proof $-$
obtain $m: 2 * mHalf\ p + 1 = p$ **and** $pK: p \leq K$ **using** $outer-facts[OF\ assms(1,2)]$ **by** $blast$
have $a \leq K$ **using** $m\ pK\ assms(4)$ **by** $linarith$
thus $?thesis$ **by** $(auto\ simp: mem-class)$
qed

lemma $card-outerClass$:

assumes $prime\ p\ OuterHyp\ K\ p\ 1 \leq a\ a \leq mHalf\ p$
shows $card\ (outerClassPoles\ K\ p\ a) + (if\ a \leq Nof\ K\ then\ 1\ else\ 0) = ell\ p\ K\ a$
proof $-$
obtain $m: 2 * mHalf\ p + 1 = p$ **and** $N: 2 * Nof\ K < p$ **using** $outer-facts[OF\ assms(1,2)]$ **by** $blast$
define P **where** $P\ j \longleftrightarrow j\ mod\ p = a\ mod\ p \vee (j + a)\ mod\ p = 0$ **for** j
have $NK: Nof\ K \leq K$ **by** $(simp\ add: Nof-def)$
have $E: \{j \in \{1..K\}. P\ j\} = outerClassPoles\ K\ p\ a \cup \{j \in \{1..Nof\ K\}. P\ j\}$

```

    using NK by (auto simp: outerClassPoles-def P-def)
  have L1:  $j = a$  if  $j \in \{1..Nof\ K\}$   $P\ j$  for  $j$ 
  proof -
    have  $jp: j < p$  using  $j\ N$  by auto
    have  $j\ mod\ p = a \vee (1 \leq a \wedge j\ mod\ p = p - a)$ 
      by (rule res-class[OF assms(4) m]) (use  $j$  in <simp add: P-def>)
    hence  $j = a \vee j = p - a$  using  $jp$  by auto
    moreover have  $j \neq p - a$  using  $j\ N\ m\ assms(4)$  by auto
    ultimately show ?thesis by blast
  qed
  have L2:  $P\ a$  by (simp add: P-def)
  have L:  $\{j \in \{1..Nof\ K\}. P\ j\} = (if\ a \leq Nof\ K\ then\ \{a\}\ else\ \{\})$ 
  proof (cases  $a \leq Nof\ K$ )
    case True
      have  $\{j \in \{1..Nof\ K\}. P\ j\} \subseteq \{a\}$  using L1 by blast
      moreover have  $a \in \{j \in \{1..Nof\ K\}. P\ j\}$  using True assms(3) L2 by simp
      ultimately have  $\{j \in \{1..Nof\ K\}. P\ j\} = \{a\}$  by blast
      thus ?thesis using True by simp
    next
      case False
      have  $\{j \in \{1..Nof\ K\}. P\ j\} = \{\}$ 
      proof (rule equals0I)
        fix  $j$  assume  $j: j \in \{j \in \{1..Nof\ K\}. P\ j\}$ 
        hence  $j = a$  using L1 by blast
        moreover have  $j \leq Nof\ K$  using  $j$  by simp
        ultimately show False using False by simp
      qed
      thus ?thesis using False by simp
    qed
  have  $ell\ p\ K\ a = card\ (outerClassPoles\ K\ p\ a) + card\ \{j \in \{1..Nof\ K\}. P\ j\}$ 
    unfolding ell-def P-def[symmetric] E by (rule card-Un-disjoint) (auto simp:
outerClassPoles-def)
  thus ?thesis unfolding L by simp
  qed

lemma zero-class:
  assumes prime  $p$  OuterHyp  $K\ p$ 
  shows  $outerClassPoles\ K\ p\ 0 = (if\ 2 * p \leq K\ then\ \{p, 2 * p\}\ else\ \{p\})$ 
  proof -
    obtain  $pK: p \leq K$  and  $K3: K < 3 * p$  and  $N: 2 * Nof\ K < p$  using
outer-facts[OF assms]
    by blast
    have  $j \in outerClassPoles\ K\ p\ 0 \longleftrightarrow j = p \vee (2 * p \leq K \wedge$ 
 $j = 2 * p)$  for  $j$ 
    proof
      assume  $j \in outerClassPoles\ K\ p\ 0$ 
      hence  $j: Nof\ K + 1 \leq j \leq K\ p\ dvd\ j$  by (auto simp: mem-class)
      then obtain  $t$  where  $t: j = p * t$  by (elim dvdE)
      have  $p * t < p * 3$  using  $t\ j\ K3$  by linarith

```

```

hence  $t < 3$  by simp
moreover have  $t \neq 0$ 
proof
  assume  $t = 0$ 
  thus False using  $t\ j(1)$  by simp
qed
ultimately have  $t = 1 \vee t = 2$  by auto
thus  $j = p \vee (2 * p \leq K \wedge j = 2 * p)$  using  $t\ j$  by (auto simp: mult.commute)
next
  assume  $j = p \vee (2 * p \leq K \wedge j = 2 * p)$ 
  thus  $j \in \text{outerClassPoles } K\ p\ 0$  using  $pK\ N$  by (auto simp: mem-class)
qed
thus ?thesis by auto
qed

```

22.4 Valuations at the poles

```

lemma vge-inv-two-mul: prime  $p \implies \neg p \text{ dvd } j \implies$ 
   $p \neq 2 \implies \text{vge } p\ (1 / (2 * \text{of-nat } j))\ 0$ 
proof -
  assume  $p$ : prime  $p$  and  $nd$ :  $\neg p \text{ dvd } j$  and  $p2$ :  $p \neq 2$ 
  have  $\neg p \text{ dvd } 2$ 
  proof
    assume  $p \text{ dvd } 2$ 
    hence  $p \leq 2$  by (rule dvd-imp-le) simp
    thus False using prime-ge-2-nat[OF  $p$ ]  $p2$  by simp
  qed
  hence  $\neg p \text{ dvd } 2 * j$  using  $nd$  prime-dvd-mult-iff[OF  $p$ ] by auto
  from vge-inverse-unit[OF  $p$  this] show ?thesis by (simp add: inverse-eq-divide)
qed

```

$\mu\text{-}X(1/(t+j^2))$ has coefficients of valuation $\geq \beta$ when $j^4 \nmid H\text{-}j$ and $1/(2j)$ do.

```

lemma muPole-coeff0: Polynomial.coeff (muPole  $j$ )  $n =$ 
  (if  $n = 1$  then  $\text{of-nat } j^4$  else if  $n =$ 
    0 then  $-(\text{of-nat } j^4 * H5\ j) - 1 / 4 + 1 / (2 * \text{of-nat } j)$  else 0)
unfolding muPole-def  $pX\text{-minus-const}$  by (auto simp: coeff-pCons split: nat.split)

```

```

lemma vge-pow-mul: prime  $p \implies \text{vge } p\ x\ b \implies$ 
   $\text{vge } p\ (x^{\wedge} n)\ (\text{of-nat } n * b)$ 
proof -
  assume  $p$ : prime  $p$  and  $h$ :  $\text{vge } p\ x\ b$ 
  show ?thesis
  proof (induction  $n$ )
    case 0
    show ?case by (simp add: vge-one)
  next
    case (Suc  $n$ )
    have  $\text{vge } p\ (x * x^{\wedge} n)\ (b + \text{of-nat } n * b)$  by (rule vge-mul[OF  $p$   $h$  Suc.IH])
  qed

```


thus ?case by (simp add: algebra-simps)
 qed
 qed

lemma nat-log-le-one-sq: prime p $\implies K < p^2 \implies \text{nat-log } p \ K \leq 1$
proof –
 assume p: prime p and K: K < p²
 have floorlog p K ≤ 2 by (rule floorlog-leI[OF K]) (use prime-gt-1-nat[OF p] in simp-all)
 thus ?thesis by (simp add: nat-log-def)
 qed

lemma vge-of-int-dvd: prime p $\implies \text{int } p \ \text{dvd } z \implies \text{vge } p \ (\text{of-int } z) \ 1$
proof –
 assume p: prime p and int p dvd z
 then obtain w where w: z = int p * w by (elim dvdE)
 have vge p (of-nat p * of-int w) (1 + 0)
 by (rule vge-mul[OF p vge-p-self[OF p] vge-of-int[OF p]])
 thus ?thesis by (simp add: w)
 qed

lemma muPole-vge:
 assumes prime p β ≤ 0 p ≠ 2 vge p (of-nat j⁴ * H5 j) β vge p (1 / (2 * of-nat j)) β
 shows vpGge p (muPole j) β
proof –
 have p: prime p and b0: β ≤ 0 and p2: p ≠ 2 using assms(1–3) .
 have n4: ¬ p dvd 4
proof
 assume p dvd 4
 hence p dvd 2 * 2 by simp
 hence p dvd 2 using prime-dvd-mult-iff[OF p, of 2 2] by simp
 hence p ≤ 2 by (rule dvd-imp-le) simp
 thus False using prime-ge-2-nat[OF p] p2 by simp
 qed
 have vge p (inverse (of-nat 4)) 0 by (rule vge-inverse-unit[OF p n4])
 hence vge p (1 / 4) 0 by (simp add: inverse-eq-divide)
 hence q: vge p (1 / 4) β by (rule vge-mono) (rule b0)
 have c0v: vge p (– (of-nat j⁴ * H5 j) – 1 / 4 + 1 / (2 * of-nat j)) β
 by (rule vge-add[OF p vge-diff[OF p vge-neg[OF assms(4)] q] assms(5)])
 have j4: vge p (of-nat j⁴) β
 by (rule vge-mono[OF vge-pow[OF p vge-of-nat[OF p]]]) (use b0 in simp)
 show ?thesis unfolding vpGge-iff muPole-coeff0 by (simp add: c0v j4 vge-zero)
 qed

lemma muPole-vge-small: prime p $\implies 1 \leq j \implies j < p \implies p \neq 2 \implies \text{vpGge } p \ (\text{muPole } j) \ 0$
proof –
 assume p: prime p and j1: 1 ≤ j and jp: j < p and p2: p ≠ 2
 have nd: ¬ p dvd j using j1 jp by (auto dest: dvd-imp-le)

have a : $vge\ p\ (of\text{-}nat\ j\ ^4 * H5\ j)\ (0 + 0)$
by (rule $vge\text{-}mul[OF\ p\ vge\text{-}pow[OF\ p\ vge\text{-}of\text{-}nat[OF\ p]]\ vge\text{-}H5\text{-}small[OF\ p\ jp]]$)
show $?thesis$ **by** (rule $muPole\text{-}vge[OF\ p\ order.refl\ p2]$) (use a $vge\text{-}inv\text{-}two\text{-}mul[OF\ p\ nd\ p2]$ **in** $simp\text{-}all$)
qed

lemma $muPole\text{-}vge\text{-}mult$: $prime\ p \implies 7 \leq p \implies 1 \leq t \implies t \leq 2 \implies vpGge\ p\ (muPole\ (t * p))\ (-\ 1)$
proof –
assume p : $prime\ p$ **and** $p7$: $7 \leq p$ **and** $t1$: $1 \leq t$ **and** $t2$: $t \leq 2$
have $p2$: $p \neq 2$ **using** $p7$ **by** $simp$
have tp : $t * p < p^2$ **using** $t2\ p7$ **by** ($simp\ add$: $power2\text{-}eq\text{-}square$)
have $v1$: $vge\ p\ (of\text{-}nat\ (t * p))\ 1$ **by** (rule $vge\text{-}dvd\text{-}nat[OF\ p]$) $simp$
have $v4$: $vge\ p\ (of\text{-}nat\ (t * p)^4)\ (of\text{-}nat\ 4 * 1)$ **by** (rule $vge\text{-}pow\text{-}mul[OF\ p\ v1]$)
have h : $vge\ p\ (H5\ (t * p))\ (-\ 5 * of\text{-}nat\ (nat\text{-}log\ p\ (t * p)))$ **by** (rule $vge\text{-}H5[OF\ p\ order.refl]$)
have nl : $nat\text{-}log\ p\ (t * p) \leq 1$ **by** (rule $nat\text{-}log\text{-}le\text{-}one\text{-}sq[OF\ p\ tp]$)
have h' : $vge\ p\ (H5\ (t * p))\ (-\ 5)$ **by** (rule $vge\text{-}mono[OF\ h]$) (use nl **in** $simp$)
have a : $vge\ p\ (of\text{-}nat\ (t * p)^4 * H5\ (t * p))\ (of\text{-}nat\ 4 * 1 + -\ 5)$
by (rule $vge\text{-}mul[OF\ p\ v4\ h']$)
have $z0$: $int\ (2 * t * p) \neq 0$ **using** $t1\ p7$ **by** $simp$
have nd : $\neg\ int\ p^{\wedge}(1 + 1)\ dvd\ int\ (2 * t * p)$
proof
assume $int\ p^{\wedge}(1 + 1)\ dvd\ int\ (2 * t * p)$
hence $p * p\ dvd\ (2 * t) * p$ **by** ($simp\ add$: $power2\text{-}eq\text{-}square\ mult\text{-}ac\ flip$:
 $of\text{-}nat\text{-}mult$
 $of\text{-}nat\text{-}power$)
hence $p\ dvd\ 2 * t$ **using** $p7$ **by** $simp$
hence $p \leq 2 * t$ **by** (rule $dvd\text{-}imp\text{-}le$) (use $t1$ **in** $simp$)
thus $False$ **using** $t2\ p7$ **by** $simp$
qed
have b : $vge\ p\ (1 / of\text{-}int\ (int\ (2 * t * p)))\ (-\ of\text{-}nat\ 1)$ **by** (rule $vge\text{-}inv\text{-}int[OF\ p\ z0\ nd]$)
show $?thesis$
by (rule $muPole\text{-}vge[OF\ p - p2]$) (use $a\ b$ **in** $\langle simp\text{-}all\ add$: $mult.assoc \rangle$)
qed

lemma $nat\text{-}log\text{-}le\text{-}one\text{-}of\text{-}lt\text{-}sq$: $prime\ p \implies K < p^2 \implies nat\text{-}log\ p\ K \leq 1$
proof –
assume p : $prime\ p$ **and** K : $K < p^2$
have $floorlog\ p\ K \leq 2$ **by** (rule $floorlog\text{-}leI[OF\ K]$) (use $prime\text{-}gt\text{-}1\text{-}nat[OF\ p]$ **in** $simp\text{-}all$)
thus $?thesis$ **by** ($simp\ add$: $nat\text{-}log\text{-}def$)
qed

lemma $muPole\text{-}vge\text{-}class$:

assumes *prime p OuterHyp K p 1 ≤ a a ≤ mHalf p j ∈ outerClassPoles K p a*
shows *vpGge p (muPole j) (- 5)*
proof –
have *p: prime p using assms(1) .*
have *p7: 7 ≤ p using outer-facts[OF assms(1,2)] by blast*
have *p2: p ≠ 2 using p7 by simp*
have *j ≤ K using assms(5) by (simp add: mem-class)*
moreover **have** *2 * K < p ^ 2 using assms(2) by (simp add: OuterHyp-def)*
ultimately **have** *jp: j < p ^ 2 by linarith*
have *h: vge p (H5 j) (- 5 * of-nat (nat-log p j)) by (rule vge-H5[OF p order.refl])*
have *nl: nat-log p j ≤ 1 by (rule nat-log-le-one-sq[OF p jp])*
have *h': vge p (H5 j) (- 5) by (rule vge-mono[OF h]) (use nl in simp)*
have *a: vge p (of-nat j ^ 4 * H5 j) (0 + - 5)*
by *(rule vge-mul[OF p vge-pow[OF p vge-of-nat[OF p]] h'])*
have *b: vge p (1 / (2 * of-nat j)) 0 by (rule vge-inv-two-mul[OF p class-not-dvd[OF*
assms] p2])
show *?thesis by (rule muPole-vge[OF p - p2]) (use a vge-mono[OF b] in simp-all)*
qed

$\mu^{\wedge 0-X}(R / \prod_{j \in S} (t + j^{\wedge 2}))$ is bounded by its residue terms, for integral R .

lemma *muXmod-vge-of-residues:*

assumes *prime p 7 ≤ p finite S vpGge p R 0 B ≤ 0*
and $\bigwedge j. j \in S \implies vpGge p ([:sqResidue S R j:] * muPole j) B$
shows *vpGge p (muXmod p S R) B*
proof –
have *p: prime p and p7: 7 ≤ p and S: finite S and R: vpGge p R 0 and B: B*
 ≤ 0
using *assms(1-5) .*
have *q: vpGge p (R div sqPoleProd S) 0*
by *(rule vpGge-div-monic[OF p R lead-coeff-sqPoleProd[OF S] vpGge-sqPoleProd[OF*
p]])
have *c: vpGge p [:muModPoly p (R div sqPoleProd S):] B*
by *(rule vpGge-mono[OF vpGge-C[OF muModPoly-vge[OF p p7 q]] B])*
show *?thesis unfolding muXmod-eq by (rule vpGge-add[OF p c vpGge-sum[OF*
p S assms(6)]])
qed

lemma *sq-sub-vge:*

assumes *prime p OuterHyp K p j ∈ outerClassPoles K p a*
shows *vge p (of-nat a ^ 2 - of-nat j ^ 2) 1*
proof –
have *p: prime p using assms(1) .*
have *j: j mod p = a mod p ∨ (j + a) mod p = 0 using assms(3) by (simp add:*
mem-class)
have *d: int p dvd (int a - int j) * (int a + int j)*
proof *(cases j mod p = a mod p)*
case *True*
hence *int j mod int p = int a mod int p by (metis of-nat-mod)*

```

    hence  $\text{int } p \text{ dvd } \text{int } a - \text{int } j$  by (simp add: mod-eq-dvd-iff dvd-diff-commute)
    thus ?thesis by simp
  next
    case False
    hence  $(j + a) \bmod p = 0$  using  $j$  by simp
    hence  $p \text{ dvd } j + a$  by (rule mod-0-imp-dvd)
    hence  $\text{int } p \text{ dvd } \text{int } a + \text{int } j$  by (simp add: add.commute flip: of-nat-add)
    thus ?thesis by simp
  qed
  have  $e: (\text{of-nat } a^2 - \text{of-nat } j^2 :: \text{rat}) = \text{of-int } ((\text{int } a - \text{int } j) * (\text{int } a + \text{int } j))$ 
  by (simp add: power2-eq-square algebra-simps)
  show ?thesis unfolding  $e$  by (rule vge-of-int-dvd[OF  $p \ d$ ])
qed

lemma eval-vge0:  $\text{prime } p \implies \text{vpGge } p \ P \ 0 \implies$ 
 $\text{vge } p \ x \ 0 \implies \text{vge } p \ (\text{poly } P \ x) \ 0$ 
using vge-poly-eval .

lemma vge-neg-sq:  $\text{prime } p \implies \text{vge } p \ (- (\text{of-nat } j^2)) \ 0$ 
proof -
  assume  $p: \text{prime } p$ 
  show ?thesis by (rule vge-neg[OF vge-pow[OF  $p \ \text{vge-of-nat}[OF \ p]$ ]])
qed

lemma vge-pow':  $\text{prime } p \implies \text{vge } p \ x \ b \implies \text{vge } p \ (x^n) \ (\text{of-nat } n * b)$ 
proof -
  assume  $p: \text{prime } p$  and  $h: \text{vge } p \ x \ b$ 
  show ?thesis
  proof (induction  $n$ )
    case 0
    show ?case by (simp add: vge-one)
  next
    case (Suc  $n$ )
    have  $\text{vge } p \ (x * x^n) \ (b + \text{of-nat } n * b)$  by (rule vge-mul[OF  $p \ h \ \text{Suc.IH}$ ])
    thus ?case by (simp add: algebra-simps)
  qed
qed

lemma vge-div-of-le:  $\text{prime } p \implies \text{vge } p \ x \ b \implies$ 
 $y \neq 0 \implies \text{of-int } (\text{vp } p \ y) \leq U \implies \text{vge } p \ (x / y) \ (b - U)$ 
proof -
  assume  $p: \text{prime } p$  and  $hx: \text{vge } p \ x \ b$  and  $y: y \neq 0$  and  $U: \text{of-int } (\text{vp } p \ y) \leq U$ 
  have  $pe: \text{prime-elem } (\text{int } p)$  using  $p$  by (simp add: prime-nat-int-transfer prime-imp-prime-elem)
  show ?thesis unfolding vge-def
  proof
    assume  $x / y \neq 0$ 
    hence  $x0: x \neq 0$  by auto
    have  $v: \text{vp } p \ (x / y) = \text{vp } p \ x - \text{vp } p \ y$  unfolding vp-def

```

by (rule qmultiplicity-divide[OF pe x0 y])
 have $b \leq \text{of-int } (vp \ p \ x)$ using hx x0 by (simp add: vge-def)
 thus $b - U \leq \text{of-int } (vp \ p \ (x / y))$ unfolding v of-int-diff using U by linarith
 qed
 qed

lemma *vp-int-le-one-of-lt-sq*: $\text{prime } p \implies z \neq 0 \implies$

$\text{nat } |z| < p^2 \implies vp \ p \ (\text{of-int } z) \leq 1$

proof –

assume p : *prime* p and z : $z \neq 0$ and l : $\text{nat } |z| < p^2$

have $vp \ p \ (\text{of-int } z) \leq \text{int } (\text{nat-log } p \ (p^2 - 1))$

by (rule vp-int-le-nat-log[OF p z]) (use l in linarith)

also have $\text{nat-log } p \ (p^2 - 1) \leq 1$

by (rule nat-log-le-one-of-lt-sq[OF p]) (use prime-gt-1-nat[OF p] in simp)

finally show ?thesis by simp

qed

lemma *D-eval-vge*:

assumes *prime* p *OuterHyp* $K \ p \ 1 \leq a \ j \in \text{outerClassPoles } K \ p \ a$

shows $vge \ p \ (\text{poly } (Dm \ (Nof \ K)) \ (- \ (\text{of-nat } j^2))) \ (\text{if } a \leq Nof \ K \ \text{then } 1 \ \text{else } 0)$

proof –

have p : *prime* p using assms(1) .

have e : $\text{poly } (Dm \ (Nof \ K)) \ x = (\prod_{k \in \{1..Nof \ K\}} \text{poly } (pX + [:of-nat \ k^2:] \ x))$ for x

by (simp add: Dm-def poly-prod)

have r : $vge \ p \ (\prod_{k \in I} \text{poly } (pX + [:of-nat \ k^2:] \ (- \ (\text{of-nat } j^2)))) \ 0$

if *finite* I for I

by (rule vge-prod0[OF p that], rule eval-vge0[OF p vpGge-pX-plus-sq[OF p] vge-neg-sq[OF p]])

show ?thesis

proof (cases $a \leq Nof \ K$)

case *True*

have aI : $a \in \{1..Nof \ K\}$ using *True* assms(3) by simp

have eq : $\text{poly } (Dm \ (Nof \ K)) \ (- \ (\text{of-nat } j^2)) =$

$\text{poly } (pX + [:of-nat \ a^2:] \ (- \ (\text{of-nat } j^2))) *$

$(\prod_{k \in \{1..Nof \ K\} - \{a\}} \text{poly } (pX + [:of-nat \ k^2:] \ (- \ (\text{of-nat } j^2))))$

unfolding e by (rule prod.remove[OF - aI]) simp

have f : $vge \ p \ (\text{poly } (pX + [:of-nat \ a^2:] \ (- \ (\text{of-nat } j^2)))) \ 1$

using sq-sub-vge[OF assms(1,2,4)] by (simp add: poly-monom algebra-simps)

have $vge \ p \ (\text{poly } (pX + [:of-nat \ a^2:] \ (- \ (\text{of-nat } j^2))) *$

$(\prod_{k \in \{1..Nof \ K\} - \{a\}} \text{poly } (pX + [:of-nat \ k^2:] \ (- \ (\text{of-nat } j^2))))$

$(1 +$

$0)$

by (rule vge-mul[OF p f r]) simp

thus ?thesis using *True* unfolding eq by simp

next

case *False*

have $vge \ p \ (\prod_{k \in \{1..Nof \ K\}} \text{poly } (pX + [:of-nat \ k^2:] \ (- \ (\text{of-nat } j^2))))$

```

0
  by (rule r) simp
  thus ?thesis using False unfolding e by simp
qed
qed

lemma node-factor-le:
  assumes prime p OuterHyp K p 1 ≤ a a ≤ mHalf p j ∈
    outerClassPoles K p a k ∈ outerClassPoles K p a j ≠ k
  shows vp p (of-nat k ^ 2 - of-nat j ^ 2) ≤ 1
proof -
  have p: prime p using assms(1) .
  have p7: 7 ≤ p using outer-facts[OF assms(1,2)] by blast
  have KK: 2 * K < p ^ 2 using assms(2) by (simp add: OuterHyp-def)
  have jK: j ≤ K and kK: k ≤ K using assms(5,6) by (simp-all add: mem-class)
  define u where u = int k - int j
  define v where v = int k + int j
  have u0: u ≠ 0 using assms(7) by (simp add: u-def)
  have v0: v ≠ 0 using assms(5) by (auto simp: v-def mem-class)
  have e: (of-nat k ^ 2 - of-nat j ^ 2 :: rat) = of-int u * of-int v
    by (simp add: u-def v-def power2-eq-square algebra-simps)
  have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer prime-imp-prime-elem)
  have vs: vp p (of-int u * of-int v) = vp p (of-int u) + vp p (of-int v)
    unfolding vp-def by (rule qmultiplicity-mult[OF pe]) (simp-all add: u0 v0)
  have au: |u| ≤ int K using jK kK by (simp add: u-def abs-le-iff)
  have av: |v| ≤ int (2 * K) using jK kK by (simp add: v-def)
  have nu: nat |u| < p ^ 2 using au KK by (simp only: nat-le-iff[symmetric])
  have nv: nat |v| < p ^ 2 using av KK by (simp only: nat-le-iff[symmetric])
  have bu: vp p (of-int u) ≤ 1 by (rule vp-int-le-one-of-lt-sq[OF p u0 nu])
  have bv: vp p (of-int v) ≤ 1 by (rule vp-int-le-one-of-lt-sq[OF p v0 nv])
  have nd: ¬ (int p dvd u ∧ int p dvd v)
proof
  assume int p dvd u ∧ int p dvd v
  hence int p dvd u + v by auto
  moreover have u + v = int (2 * k) by (simp add: u-def v-def)
  ultimately have p dvd 2 * k by (simp only: int-dvd-int-iff)
  hence p dvd 2 ∨ p dvd k using prime-dvd-mult-iff[OF p] by auto
  moreover have ¬ p dvd 2 using p7 by (auto dest: dvd-imp-le)
  moreover have ¬ p dvd k by (rule class-not-dvd[OF assms(1-4,6)])
  ultimately show False by blast
qed
  have z: vp p (of-int u) = 0 ∨ vp p (of-int v) = 0
    using nd by (auto simp: vp-def qmultiplicity-of-int not-dvd-imp-multiplicity-0)
  have nn: 0 ≤ vp p (of-int u) 0 ≤ vp p (of-int v)
    by (simp-all add: vp-def qmultiplicity-of-int)
  show ?thesis unfolding e vs using z bu bv nn by auto
qed

```

lemma node-prod-le:

assumes *prime p OuterHyp K p* $1 \leq a \leq mHalf\ p\ j \in outerClassPoles\ K\ p\ a$
shows $(\prod k \in outerClassPoles\ K\ p\ a - \{j\}. of_nat\ k^2 - of_nat\ j^2) \neq$
 $(0 :: rat) \wedge$
 $of_int\ (vp\ p\ (\prod k \in outerClassPoles\ K\ p\ a - \{j\}. of_nat\ k^2 -$
 $of_nat\ j^2)) \leq of_nat\ (card\ (outerClassPoles\ K\ p\ a)) - (1 :: rat)$
proof –
have *p: prime p using assms(1)* .
have *finC: finite (outerClassPoles K p a) by (simp add: outerClassPoles-def)*
have *fin: finite (outerClassPoles K p a - {j}) using finC by simp*
have *nz: (of-nat k^2 - of-nat j^2 :: rat) ≠ 0*
if $k \in outerClassPoles\ K\ p\ a - \{j\}$ **for** *k*
using *that by (simp add: power2-eq-iff-nonneg)*
have *pz: $(\prod k \in outerClassPoles\ K\ p\ a - \{j\}. of_nat\ k^2 - of_nat\ j^2) \neq$*
 $(0 :: rat)$
using *fin nz by (simp add: prod-zero-iff)*
have *vp p $(\prod k \in outerClassPoles\ K\ p\ a - \{j\}. of_nat\ k^2 - of_nat\ j^2 ::$*
 $rat)$
 $= (\sum k \in outerClassPoles\ K\ p\ a - \{j\}. vp\ p\ (of_nat\ k^2 - of_nat\ j^2 ::$
 $rat))$
by *(rule vp-prod-rat[OF p fin nz])*
also have $\dots \leq (\sum k \in outerClassPoles\ K\ p\ a - \{j\}. 1)$
by *(rule sum-mono) (use node-factor-le[OF assms(1-5)] in auto)*
also have $\dots = int\ (card\ (outerClassPoles\ K\ p\ a) - 1)$ **using** *assms(5)*
by *(simp add: card-Diff-singleton)*
finally have *v: vp p $(\prod k \in$*
 $outerClassPoles\ K\ p\ a - \{j\}. of_nat\ k^2 - of_nat\ j^2 :: rat) \leq$
 $int\ (card\ (outerClassPoles\ K\ p\ a) - 1)$.
have *c1: $1 \leq card\ (outerClassPoles\ K\ p\ a)$ using assms(5) finC*
by *(auto simp: card-gt-0-iff Suc-le-eq)*
have *(of-int (vp p $(\prod k \in outerClassPoles\ K\ p\ a - \{j\}. of_nat\ k^2 -$*
 $of_nat\ j^2 :: rat)) :: rat)$
 $\leq of_int\ (int\ (card\ (outerClassPoles\ K\ p\ a) - 1))$ **using** *v by (simp only:*
 $of_int-le-iff)$
hence *(of-int (vp p $(\prod k \in outerClassPoles\ K\ p\ a - \{j\}. of_nat\ k^2 -$*
 $of_nat\ j^2 :: rat)) :: rat)$
 $\leq of_nat\ (card\ (outerClassPoles\ K\ p\ a)) - 1$ **using** *c1 by (simp add:*
 $of_nat-diff)$
thus *?thesis using pz by simp*
qed

The class-*a* factor δ .

definition *dlt :: nat $\Rightarrow$ nat $\Rightarrow$ rat where* $dlt\ K\ a = (if\ a \leq$
 $Nof\ K\ then\ 1\ else\ 0)$

(4.12), ordinary rows: a same-class entry between rows $i, i' < l - 2$.

lemma *entry-ord:*

assumes *p: prime p and H: OuterHyp K p and a1: $1 \leq a$ and am: $a \leq mHalf$*
 p
shows $vpGge\ p\ (\mu Xmod\ p\ (outerClassPoles\ K\ p\ a))$

```

      (Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outerClassPoles K p a) *
      (pX +
      [:of-nat a ^ 2:] ^ i * (pX + [:of-nat a ^ 2:] ^ i'))
      (min 0 (of-nat i + of-nat i' + 6 * dlt K a - of-nat (ell p K a) - 4))
    proof -
      have p7: 7 ≤ p using H by (simp add: OuterHyp-def)
      have finC: finite (outerClassPoles K p a) by (simp add: outerClassPoles-def)
      define R where R = Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outer-
      ClassPoles K p a) * (pX +
      [:of-nat a ^ 2:] ^ i * (pX + [:of-nat a ^ 2:] ^ i'))
      have R0: vpGge p R 0
      unfolding R-def by (intro vpGge-mul0[OF p] vpGge-pow0[OF p] vpGge-Dm[OF
      p] vpGge-sqPoleProd[OF
      p] vpGge-pX-plus-sq[OF p])
      have cardC: of-nat (card (outerClassPoles K p a)) = of-nat (ell p K a) - dlt K
      a
      using card-outerClass[OF p H a1 am] by (auto simp: dlt-def)
      show ?thesis unfolding R-def[symmetric]
      proof (rule muXmod-vge-of-residues[OF p p7 finC R0 min.cobounded1])
        fix j assume j: j ∈ outerClassPoles K p a
        have num: vge p (poly R (- (of-nat j ^ 2))) (5 * dlt K a + of-nat i + of-nat
        i')
        proof -
          have x: vge p (- (of-nat j ^ 2)) 0 by (rule vge-neg-sq[OF p])
          have d: vge p (poly (Dm (Nof K)) (- (of-nat j ^ 2))) (dlt K a)
          using D-eval-vge[OF p H a1 j] by (simp add: dlt-def)
          have s: vge p (poly (sqPoleProd (tailPoles K - outerClassPoles K p a)) (-
          (of-nat j ^ 2))) 0
          by (rule eval-vge0[OF p vpGge-sqPoleProd[OF p] x])
          have f: vge p (poly (pX + [:of-nat a ^ 2:] (- (of-nat j ^ 2))) 1
          using sq-sub-vge[OF p H j] by (simp add: poly-monom algebra-simps)
          have vge p (poly (Dm (Nof K)) (- (of-nat j ^ 2)) ^ 5 * poly (sqPoleProd
          (tailPoles K -
          outerClassPoles K p a)) (- (of-nat j ^ 2)) *
          poly (pX + [:of-nat a ^ 2:] (- (of-nat j ^ 2)) ^ i * poly (pX +
          [:of-nat a ^ 2:] (- (of-nat j ^ 2)) ^ i')
          (of-nat 5 * dlt K a + 0 + of-nat i * 1 + of-nat i' * 1)
          by (rule vge-mul[OF p vge-mul[OF p vge-mul[OF p vge-pow'[OF p d] s]
          vge-pow'[OF p f]]
          vge-pow'[OF p f])
          thus ?thesis unfolding R-def poly-mult poly-power by simp
        qed
      have res: vge p (sqResidue (outerClassPoles K p a) R j) (5 * dlt K a + of-nat
      i + of-nat i' -
      (of-nat (card (outerClassPoles K p a)) - 1))
      unfolding sqResidue-def using node-prod-le[OF p H a1 am j]
      by (intro vge-div-of-le[OF p num]) auto
      show vpGge p ([:sqResidue (outerClassPoles K p a) R j:] * muPole j) (min 0
      (of-nat i +

```



```

      of-nat i' + 6 * dlt K a - of-nat (ell p K a) - 4))
    by (rule vpGge-mono[OF vpGge-mul[OF p vpGge-C[OF res] muPole-vge-class[OF
p H a1 am j]]])
      (use cardC min.cobounded2[of 0 of-nat i + of-nat i' + 6 * dlt K a - of-nat
(ell p K a) -
4 :: rat] in linarith)
  qed
qed

```

lemma *small-poles*:

```

  assumes prime p OuterHyp K p 1 ≤ a a ≤ mHalf p
  shows {j ∈ outerClassPoles K p a. j < p} = (if a ≤ Nof K then {p - a} else
{a, p - a})
proof -
  obtain m: 2 * mHalf p + 1 = p using outer-facts[OF assms(1,2)] by blast
  have ap: a < p and pap: p - a < p using m assms(3,4) by auto
  have j ∈ {j ∈ outerClassPoles K p a. j < p} ⟷ j = p - a ∨
    (¬ a ≤ Nof K ∧ j = a) for j
  using class-small[OF assms, of j] pa-mem[OF assms] a-mem-iff[OF assms] ap
pap by auto
  thus ?thesis by auto
qed

```

22.5 The two small poles $a, p - a$

$\sum -\{v=a\} \wedge \{p-a\} v \wedge -5 \equiv 0 \pmod{p}$, by pairing $v \leftrightarrow p - v$.

lemma *pair-sum-vge*:

```

  assumes prime p p ≠ 2 1 ≤ a 2 * a < p
  shows vge p (∑ v = a..p - a. 1 / of-nat v ^ 5) 1
proof -
  have p: prime p using assms(1) .
  define S where S = (∑ v = a..p - a. 1 / of-nat v ^ 5 :: rat)
  have bij: bij-betw (λv. p - v) {a..p - a} {a..p - a}
    by (rule bij-betw-byWitness[where f' = λv. p - v]) (use assms(4) in auto)
  have S = (∑ v = a..p - a. 1 / of-nat (p - v) ^ 5 :: rat)
    unfolding S-def using sum.reindex-bij-betw[OF bij, of λv. 1 / of-nat v ^ 5 ::
rat]
    by order
  hence two: 2 * S = (∑ v = a..p - a. 1 / of-nat v ^ 5 + 1 / of-nat (p - v) ^
5 :: rat)
    unfolding S-def by fastforce
  have n2: ¬ p dvd 2 using p assms(2) prime-ge-2-nat[OF p] dvd-imp-le[of p 2]
by auto
  have trm: vge p (1 / of-nat v ^ 5 + 1 / of-nat (p - v) ^ 5 :: rat) 1 if v: v ∈
{a..p - a}
    for v
  proof -
    have v1: 1 ≤ v v < p using v assms(3,4) by auto
    have ¬ p dvd v using v1 by force

```

```

moreover have  $\neg p \text{ dvd } (p - v)$  using v1 dvd-imp-le[of p p - v] by linarith
ultimately have nvw:  $\neg p \text{ dvd } (v * (p - v)) ^ 5$ 
using p prime-dvd-power prime-dvd-mult-iff by blast
define x where x = (of-nat v :: rat)
define y where y = (of-nat (p - v) :: rat)
have xy:  $x + y = \text{of-nat } p$  unfolding x-def y-def using v1 by simp
have x0:  $x \neq 0$  and y0:  $y \neq 0$  unfolding x-def y-def using v1 by auto
have fac:  $x ^ 5 + y ^ 5 = (x + y) * (x ^ 4 - x ^ 3 * y + x ^ 2 * y ^ 2 - x$ 
 $* y ^ 3 + y ^ 4)$ 
by (auto simp: algebra-simps eval-nat-numeral)
have q:  $x ^ 4 - x ^ 3 * y + x ^ 2 * y ^ 2 - x * y ^ 3 + y ^ 4 =$ 
 $\text{of-int } (\text{int } v ^ 4 - \text{int } v ^ 3 * \text{int } (p - v) + \text{int } v ^ 2 * \text{int } (p - v) ^ 2 -$ 
 $\text{int } v * \text{int } (p -$ 
 $v) ^ 3 + \text{int } (p - v) ^ 4)$ 
unfolding x-def y-def by force
have e:  $1 / x ^ 5 + 1 / y ^ 5 = (x ^ 5 + y ^ 5) * \text{inverse } (\text{of-nat } ((v * (p -$ 
 $v)) ^ 5))$ 
unfolding x-def y-def using x0 y0 unfolding x-def y-def
by (auto simp: field-simps power-mult-distrib)
have numv:  $\text{vge } p (x ^ 5 + y ^ 5) 1$ 
proof -
have vge p ( $\text{of-nat } p * \text{of-int } (\text{int } v ^ 4 - \text{int } v ^ 3 * \text{int } (p - v) + \text{int } v ^ 2$ 
 $* \text{int } (p -$ 
 $v) ^ 2 - \text{int } v * \text{int } (p - v) ^ 3 + \text{int } (p - v) ^ 4)) (1 + 0)$ 
by (rule vge-mul[OF p vge-p-self[OF p] vge-of-int[OF p]])
thus ?thesis unfolding fac xy[symmetric] q by auto
qed
have vge p ( $((x ^ 5 + y ^ 5) * \text{inverse } (\text{of-nat } ((v * (p - v)) ^ 5))) (1 + 0)$ 
by (rule vge-mul[OF p numv vge-inverse-unit[OF p nvw]])
thus ?thesis using e unfolding x-def y-def by algebra
qed
have vge p ( $2 * S$ ) 1 unfolding two by (rule vge-sum[OF p]) (use trm in auto)
hence vge p ( $\text{inverse } (\text{of-nat } 2) * (2 * S)) (0 + 1)$ 
by (rule vge-mul[OF p vge-inverse-unit[OF p n2]])
thus ?thesis unfolding S-def by fastforce
qed

```

The constant term of $\mu\text{-}X(1/(t+j^2))$.

definition *c0* :: *nat* $\Rightarrow$ *rat* **where** *c0 j* =
 $-(\text{of-nat } j ^ 4 * H5 j) - 1 / 4 + 1 / (2 * \text{of-nat } j)$

lemma *muPole-eq*: $\mu\text{Pole } j = [:\text{c0 } j, \text{of-nat } j ^ 4:]$
unfolding *muPole-def c0-def pX-minus-const* **by** *simp*

lemma *muPole-coeff*: $\text{Polynomial.coeff } (\mu\text{Pole } j) n = (\text{if } n = 1 \text{ then } \text{of-nat } j ^ 4 \text{ else}$
 $\text{if } n = 0 \text{ then } \text{c0 } j \text{ else } 0)$
by (*auto simp: muPole-eq coeff-pCons split: nat.split*)

```

lemma H5-split:  $1 \leq a \implies a \leq b \implies$ 
 $H5\ b = H5\ (a - 1) + (\sum\ v = a..b.\ 1\ /\ of\text{-}nat\ v\ \wedge\ 5)$ 
proof -
  assume a1:  $1 \leq a$  and ab:  $a \leq b$ 
  have u:  $\{1..b\} = \{1..a-1\} \cup \{a..b\}$  using a1 ab by auto
  show ?thesis unfolding H5-def u by (rule sum.union-disjoint) auto
qed

```

The two-node congruence $\mu\text{-}X(1/(t+a^2)) \equiv \mu\text{-}X(1/(t+(p-a)^2)) \pmod{p}$ coefficientwise.

```

lemma pole-congr:
  assumes prime p  $p \neq 2$   $1 \leq a$   $2 * a < p$ 
  shows  $vge\ p\ (Polynomial.coeff\ (\mu Pole\ a)\ n - Polynomial.coeff\ (\mu Pole\ (p - a))\ n)\ 1$ 
proof -
  have p: prime p using assms(1) .
  define b where  $b = p - a$ 
  have ab:  $a \leq b$   $a + b = p$   $1 \leq b$   $b < p$   $a < p$  using assms(3,4) unfolding b-def
    by auto
  have pb:  $(of\text{-}nat\ p :: rat) = of\text{-}nat\ a + of\text{-}nat\ b$  using ab(2) by force
  have d4:  $(of\text{-}nat\ b^4 - of\text{-}nat\ a^4 :: rat) =$ 
 $of\text{-}nat\ p * of\text{-}int\ ((int\ b - int\ a) * (int\ b^2 + int\ a^2))$ 
    using pb by (auto simp: algebra-simps eval-nat-numeral)
  have  $vge\ p\ (Polynomial.coeff\ (\mu Pole\ a)\ n - Polynomial.coeff\ (\mu Pole\ b)\ n)\ 1$ 
proof (cases n = 1)
  case True
  have e:  $Polynomial.coeff\ (\mu Pole\ a)\ n - Polynomial.coeff\ (\mu Pole\ b)\ n =$ 
 $-(of\text{-}nat\ b^4 - of\text{-}nat\ a^4)$ 
    using True unfolding muPole-coeff by fastforce
  have  $vge\ p\ (of\text{-}nat\ p * of\text{-}int\ ((int\ b - int\ a) * (int\ b^2 + int\ a^2)))\ (1 + 0)$ 
    by (rule vge-mul[OF p vge-p-self[OF p] vge-of-int[OF p]])
  hence  $vge\ p\ (of\text{-}nat\ b^4 - of\text{-}nat\ a^4)\ 1$  using d4 by fastforce
  thus ?thesis unfolding e by (rule vge-neg)
next
  case n1: False
  show ?thesis
proof (cases n = 0)
  case True
  have e:  $Polynomial.coeff\ (\mu Pole\ a)\ n - Polynomial.coeff\ (\mu Pole\ b)\ n =$ 
 $c0\ a - c0\ b$ 
    using True unfolding muPole-coeff by fastforce
  define S where  $S = (\sum\ v = a..b.\ 1\ /\ of\text{-}nat\ v\ \wedge\ 5 :: rat)$ 
  define H where  $H = H5\ (a - 1)$ 
  have hb:  $H5\ b = H + S$  unfolding H-def S-def using H5-split[of a b] ab
    assms(3) by fastforce
  have ha:  $H5\ a = H + 1\ /\ of\text{-}nat\ a\ \wedge\ 5$  unfolding H-def using H5-split[of a a]
    assms(3) by fastforce

```

```

    have a0: (of-nat a :: rat) ≠ 0 and b0: (of-nat b :: rat) ≠ 0
      using ab assms(3) by auto
    have ce: c0 a - c0 b = (of-nat b ^ 4 - of-nat a ^ 4) * H + of-nat b ^ 4 *
S - of-nat p *
      inverse (of-nat (2 * a * b))
    proof -
      have i: of-nat p * inverse (of-nat (2 * a * b)) =
        1 / (2 * of-nat a) + 1 / (2 * (of-nat b :: rat))
        using pb a0 b0 by (fastforce simp: field-simps)
      have j: of-nat a ^ 4 * (1 / of-nat a ^ 5) = 1 / (of-nat a :: rat) using a0
        by (force simp: eval-nat-numeral)
      show ?thesis unfolding c0-def ha hb i using j by (simp add: algebra-simps)
    qed
    have hv: vge p H 0 unfolding H-def using vge-H5-small[OF p] ab by auto
    have v1: vge p ((of-nat b ^ 4 - of-nat a ^ 4) * H) 1
    proof -
      have vge p (of-nat p * of-int ((int b - int a) * (int b ^ 2 + int a ^ 2))) (1
+ 0)
        by (rule vge-mul[OF p vge-p-self[OF p] vge-of-int[OF p]])
      hence vge p (of-nat b ^ 4 - of-nat a ^ 4) 1 using d4 by force
      from vge-mul[OF p this hv] show ?thesis by simp
    qed
    have sv: vge p S 1 unfolding S-def b-def by (rule pair-sum-vge[OF assms])
    have v2: vge p (of-nat b ^ 4 * S) 1 using vge-mul[OF p vge-pow[OF p
vge-of-nat[OF p]] sv]
      by force
    have ¬ p dvd 2 using p assms(2) prime-ge-2-nat[OF p] dvd-imp-le[of p 2]
by force
    moreover have ¬ p dvd a using ab assms(3) dvd-imp-le[of p a] by linarith
    moreover have ¬ p dvd b using ab dvd-imp-le[of p b] by linarith
    ultimately have nd: ¬ p dvd 2 * a * b using prime-dvd-mult-iff[OF p] by
presburger
    have v3: vge p (of-nat p * inverse (of-nat (2 * a * b))) 1
      using vge-mul[OF p vge-p-self[OF p] vge-inverse-unit[OF p nd]] by fastforce
    show ?thesis unfolding e ce by (rule vge-diff[OF p vge-add[OF p v1 v2] v3])
  next
    case False
    have e: Polynomial.coeff (muPole a) n - Polynomial.coeff (muPole b) n = 0
      using n1 False unfolding muPole-coeff by algebra
    show ?thesis unfolding e by (rule vge-zero)
  qed
qed
thus ?thesis unfolding b-def .
qed

lemma eval-sub-vge:
  assumes prime p vpGge p R 0 vge p x 0 vge p y 0 vge p (x - y) b
  shows vge p (poly R x - poly R y) b
proof -

```

```

have p: prime p and R: vpGge p R 0 and x: vge p x 0 and y: vge p y 0 and
xy: vge p (x -
  y) b using assms .
have e: poly R x - poly R y = ( $\sum i \leq \text{degree } R. \text{Polynomial.coeff } R i * (x \wedge i - y \wedge i)$ )
by (simp add: poly-altdef sum-subtractf right-diff-distrib)
show ?thesis unfolding e
proof (rule vge-sum[OF p])
  fix i
  have s: vge p ( $\sum k < i. y \wedge (i - \text{Suc } k) * x \wedge k$ ) 0
  proof (rule vge-sum[OF p])
    fix k
    show vge p ( $y \wedge (i - \text{Suc } k) * x \wedge k$ ) 0 using vge-mul[OF p vge-pow[OF p
y] vge-pow[OF p x]]
    by simp
  qed simp
  have vge p ( $\text{Polynomial.coeff } R i * ((x - y) * (\sum k < i. y \wedge (i - \text{Suc } k) * x \wedge k))$ ) (0 + (b + 0))
  by (rule vge-mul[OF p R[unfolded vpGge-iff, rule-format] vge-mul[OF p xy s]])
  thus vge p ( $\text{Polynomial.coeff } R i * (x \wedge i - y \wedge i)$ ) b unfolding power-diff-sumr2
by simp
qed simp
qed

```

The divided difference at the two nodes $a, p - a$ is integral.

lemma *two-pole-vge*:

```

assumes p: prime p and p7:  $7 \leq p$  and a1:  $1 \leq a$  and a2:  $2 * a < p$ 
and R: vpGge p R 0
shows vpGge p ( $\text{muXmod } p \{a, p - a\} R$ ) 0
proof -
  define b where  $b = p - a$ 
  have ab:  $a \neq b$  using a2 by (simp add: b-def)
  have p2:  $p \neq 2$  using p7 by simp
  define d where  $d = (\text{of-nat } b \wedge 2 - \text{of-nat } a \wedge 2 :: \text{rat})$ 
  have d0:  $d \neq 0$  using ab by (simp add: d-def power2-eq-iff)
  have dv:  $\text{of-int } (vp p d) \leq (1 :: \text{rat})$ 
  proof -
    have ba:  $a < b$   $b + a = p$  using a2 by (simp-all add: b-def)
    have e:  $d = \text{of-nat } (b - a) * \text{of-nat } p$ 
    proof -
      have d =  $(\text{of-nat } b - \text{of-nat } a) * (\text{of-nat } b + \text{of-nat } a)$ 
      by (simp add: d-def power2-eq-square algebra-simps)
      also have  $\text{of-nat } b + \text{of-nat } a = (\text{of-nat } p :: \text{rat})$  by (metis ba(2) of-nat-add)
      also have  $\text{of-nat } b - \text{of-nat } a = (\text{of-nat } (b - a) :: \text{rat})$  using ba(1)
      by (simp add: of-nat-diff)
      finally show ?thesis .
    qed
  have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer)

```

```

prime-imp-prime-elim)
  have g0:  $0 < b - a$  and gp:  $b - a < p$  using ba a1 by (auto simp: b-def)
  have nd:  $\neg p \text{ dvd } (b - a)$ 
  proof
    assume p dvd b - a
    hence  $p \leq b - a$  by (rule dvd-imp-le) (rule g0)
    thus False using gp by simp
  qed
  have v1:  $vp \ p \ (of\text{-}nat \ (b - a)) = 0$  unfolding vp-def qmultiplicity-of-nat
multiplicity-int
  using not-dvd-imp-multiplicity-0[OF nd] by simp
  have v2:  $vp \ p \ (of\text{-}nat \ p) = 1$  unfolding vp-def qmultiplicity-of-nat multiplic-
ity-int
  using prime-multiplicity-prime[OF p p] by simp
  have vp p d =  $vp \ p \ (of\text{-}nat \ (b - a)) + vp \ p \ (of\text{-}nat \ p)$ 
  unfolding e vp-def by (rule qmultiplicity-mult[OF pe]) (use g0 prime-gt-0-nat[OF
p] in
    simp-all)
  thus ?thesis using v1 v2 by simp
  qed
  have res:  $\text{muXmod } p \ \{a, b\} \ R = [\text{muModPoly } p \ (R \text{ div } sqPoleProd \ \{a, b\}):] +$ 
 $[\text{poly } R \ (- \ (of\text{-}nat \ a \ ^2)) / d:] * \text{muPole } a + [\text{poly } R \ (- \ (of\text{-}nat \ b \ ^2)) /$ 
 $(- \ d):] *$ 
 $\text{muPole } b$ 
  using ab by (simp add: muXmod-eq sqResidue-def d-def insert-Diff-if add.assoc)
  have c:  $vge \ p \ (\text{muModPoly } p \ (R \text{ div } sqPoleProd \ \{a, b\})) \ 0$ 
  by (rule muModPoly-vge[OF p p7 vpGge-div-monic[OF p R lead-coeff-sqPoleProd
vpGge-sqPoleProd[OF
p]]]) simp
  have cf:  $\bigwedge n. \ vge \ p \ (\text{poly } R \ (- \ (of\text{-}nat \ a \ ^2)) / d * \text{Polynomial.coeff } (\text{muPole }
a) \ n +$ 
 $\text{poly } R \ (- \ (of\text{-}nat \ b \ ^2)) / (- \ d) * \text{Polynomial.coeff } (\text{muPole } b) \ n) \ 0$ 
  proof -
    fix n
    have x:  $vge \ p \ (\text{poly } R \ (- \ (of\text{-}nat \ a \ ^2))) \ 0$  by (rule eval-vge0[OF p R
vge-neg-sq[OF p]])
    have xy:  $vge \ p \ (\text{poly } R \ (- \ (of\text{-}nat \ a \ ^2)) - \text{poly } R \ (- \ (of\text{-}nat \ b \ ^2))) \ 1$ 
  proof -
    have ba:  $a < b \ b + a = p$  using a2 by (simp-all add: b-def)
    have e:  $d = of\text{-}nat \ (b - a) * of\text{-}nat \ p$ 
  proof -
    have d =  $(of\text{-}nat \ b - of\text{-}nat \ a) * (of\text{-}nat \ b + of\text{-}nat \ a)$ 
    by (simp add: d-def power2-eq-square algebra-simps)
    also have  $of\text{-}nat \ b + of\text{-}nat \ a = (of\text{-}nat \ p :: rat)$  by (metis ba(2) of-nat-add)
    also have  $of\text{-}nat \ b - of\text{-}nat \ a = (of\text{-}nat \ (b - a) :: rat)$  using ba(1)
    by (simp add: of-nat-diff)
    finally show ?thesis .
  qed
  have vge p (of-nat (b - a) * of-nat p) (0 + 1)

```

```

    by (rule vge-mul[OF p vge-of-nat[OF p] vge-p-self[OF p]])
  hence dd: vge p d 1 by (simp add: e)
  have - (of-nat a ^ 2) - - (of-nat b ^ 2) = d by (simp add: d-def)
  hence vge p (- (of-nat a ^ 2) - - (of-nat b ^ 2)) 1 using dd by simp
  thus ?thesis by (rule eval-sub-vge[OF p R vge-neg-sq[OF p] vge-neg-sq[OF
p]])
qed
have vge p (Polynomial.coeff (muPole a) n) 0 and yb: vge p (Polynomial.coeff
(muPole b) n) 0
  using muPole-vge-small[OF p a1 - p2] muPole-vge-small[OF p - - p2, of b]
a1 a2
  by (auto simp: vpGge-iff b-def)
have yy: vge p (Polynomial.coeff (muPole a) n - Polynomial.coeff (muPole b)
n) 1
  unfolding b-def by (rule pole-congr[OF p p2 a1 a2])
have num: vge p (poly R (- (of-nat a ^ 2)) * Polynomial.coeff (muPole a) n
-
  poly R (- (of-nat b ^ 2)) * Polynomial.coeff (muPole b) n) 1
proof -
  have eq: poly R (- (of-nat a ^ 2)) * Polynomial.coeff (muPole a) n -
    poly R (- (of-nat b ^ 2)) * Polynomial.coeff (muPole b) n =
    poly R (- (of-nat a ^ 2)) * (Polynomial.coeff (muPole a) n -
      Polynomial.coeff (muPole b) n) +
    (poly R (- (of-nat a ^ 2)) - poly R (- (of-nat b ^ 2))) * Polynomial.coeff
(muPole b) n
  by (simp add: algebra-simps)
  have A: vge p (poly R (- (of-nat a ^ 2)) * (Polynomial.coeff (muPole a) n -
    Polynomial.coeff (muPole b) n)) 1
  using vge-mul[OF p x yy] by simp
  have B: vge p ((poly R (- (of-nat a ^ 2)) - poly R (- (of-nat b ^ 2))) *
    Polynomial.coeff (muPole b) n) 1
  using vge-mul[OF p xy yb] by simp
  show ?thesis unfolding eq by (rule vge-add[OF p A B])
qed
have e: poly R (- (of-nat a ^ 2)) / d * Polynomial.coeff (muPole a) n +
  poly R (- (of-nat b ^ 2)) / (- d) * Polynomial.coeff (muPole b) n =
  (poly R (- (of-nat a ^ 2)) * Polynomial.coeff (muPole a) n -
    poly R (- (of-nat b ^ 2)) * Polynomial.coeff (muPole b) n) / d
  using d0 by (simp add: field-simps)
show vge p (poly R (- (of-nat a ^ 2)) / d * Polynomial.coeff (muPole a) n +
  poly R (- (of-nat b ^ 2)) / (- d) * Polynomial.coeff (muPole b) n) 0
  unfolding e using vge-div-of-le[OF p num d0 dv] by simp
qed
show ?thesis unfolding b-def[symmetric] res
  unfolding vpGge-iff
proof
  fix n
  have vge p (Polynomial.coeff [:muModPoly p (R div sqPoleProd {a, b}):] n +
    (poly R (- (of-nat a ^ 2)) / d * Polynomial.coeff (muPole a) n + poly R

```

```

(− (of-nat b ^ 2)) /
  (− d) * Polynomial.coeff (muPole b) n)) 0
  by (rule vge-add[OF p - cf]) (simp add: coe-pCons c vge-zero split: nat.split)
  thus vge p (Polynomial.coeff ([:muModPoly p (R div sqPoleProd {a,
    b}):] + [:poly R (− (of-nat a ^ 2)) / d:] * muPole a +
    [:poly R (− (of-nat b ^ 2)) / (− d):] * muPole b) n) 0 by (simp add:
algebra-simps)
qed
qed

```

(4.12), rows $i \geq l - 2$: once $E-a$ is a factor, the entry is integral.

lemma *entry-large*:

```

  assumes p: prime p and H: OuterHyp K p and a1: 1 ≤ a and am: a ≤
    mHalf p and R: vpGge p R 0
  shows vpGge p (muXmod p (outerClassPoles K p a) (R * sqPoleProd {j ∈
    outerClassPoles K p a. p < j})) 0
  proof −
    have p7: 7 ≤ p using H by (simp add: OuterHyp-def)
    have split: {j ∈ outerClassPoles K p a. p < j} ∪ {j ∈ outerClassPoles K p a. j
    <
      p} = outerClassPoles K p a
    proof −
      have j ≠ p if j ∈ outerClassPoles K p a for j
        using class-not-dvd[OF p H a1 am that] by auto
      hence p < j ∨ j < p if j ∈ outerClassPoles K p a for j using that
        by (meson linorder-neqE-nat)
      thus ?thesis by auto
    qed
    have cancel: muXmod p (outerClassPoles K p a) (R * sqPoleProd {j ∈ outer-
    ClassPoles K p a. p <
      j}) = muXmod p {j ∈ outerClassPoles K p a. j < p} R
    proof −
      have fin: finite {j ∈ outerClassPoles K p a. P j} for P
        by (rule finite-subset[of - {Nof K + 1..K}]) (auto simp: outerClassPoles-def)
      have muXmod p ({j ∈ outerClassPoles K p a. p < j} ∪ {j ∈
        outerClassPoles K p a. j < p})
        (R * sqPoleProd {j ∈ outerClassPoles K p a. p < j}) = muXmod p {j ∈
        outerClassPoles K p a. j < p} R
        by (rule muXmod-mul-cancel-set[OF fin fin]) auto
      thus ?thesis unfolding split .
    qed
    show ?thesis unfolding cancel small-poles[OF p H a1 am]
    proof (cases a ≤ Nof K)
      case True
        obtain m: 2 * mHalf p + 1 = p using outer-facts[OF p H] by blast
        have b1: 1 ≤ p − a and bp: p − a < p and p2: p ≠ 2 using m am a1 p7 by
        auto
        show vpGge p (muXmod p (if a ≤ Nof K then {p − a} else {a, p − a}) R) 0
          unfolding if-P[OF True]

```



```

proof (rule muXmod-vge-of-residues[OF p p7 - R order.refl])
  show finite {p - a} by simp
  fix j assume j ∈ {p - a}
  hence j: j = p - a by simp
  have vge p (sqResidue {p - a} R (p - a)) 0
    unfolding sqResidue-def by (simp add: eval-vge0[OF p R vge-neg-sq[OF
p]])
  from vpGge-mul[OF p vpGge-C[OF this] muPole-vge-small[OF p b1 bp p2]]
  show vpGge p ([:sqResidue {p - a} R j:] * muPole j) 0 unfolding j by simp
qed
next
case False
obtain m: 2 * mHalf p + 1 = p using outer-facts[OF p H] by blast
hence a2: 2 * a < p using am by linarith
show vpGge p (muXmod p (if a ≤ Nof K then {p - a} else {a, p - a}) R) 0
  unfolding if-not-P[OF False]
  by (rule two-pole-vge[OF p p7 a1 a2 R])
qed
qed

```

22.6 The zero class

```

lemma vp-three-sq: prime p ⇒ 7 ≤ p ⇒
  vp p (3 * of-nat p ^ 2) ≤ 2
proof -
  assume p: prime p and p7: 7 ≤ p
  have pe: prime-elem (int p) using p by (simp add: prime-nat-int-transfer prime-imp-prime-elem)
  have n3: ¬ p dvd 3 using p7 by (auto dest: dvd-imp-le)
  have m3: multiplicity p (3::nat) = 0 by (rule not-dvd-imp-multiplicity-0[OF
n3])
  have vp p (of-nat 3) = 0 unfolding vp-def qmultiplicity-of-nat multiplicity-int
m3 by simp
  hence v3: vp p (3 :: rat) = 0 by simp
  have vp p (of-nat p) = 1 unfolding vp-def qmultiplicity-of-nat multiplicity-int
using prime-multiplicity-prime[OF p p] by simp
  hence vp1: vp p (of-nat p :: rat) = 1 .
  have p0: (of-nat p :: rat) ≠ 0 using p by (simp add: prime-gt-0-nat)
  have vp p (3 * of-nat p ^ 2 :: rat) = vp p (3 :: rat) + vp p (of-nat p ^ 2 :: rat)
  unfolding vp-def by (rule qmultiplicity-mult[OF pe]) (simp-all add: p0 prime-gt-0-nat[OF
p])
  also have vp p (of-nat p ^ 2 :: rat) = 2 by (simp add: vp-power-rat[OF p p0]
vp1)
  finally show ?thesis using v3 by simp
qed

```

The zero class with a single pole p .

lemma entry-zero-one:

```

assumes prime p OuterHyp K p K < 2 * p vpGge p R 0
shows vpGge p (muXmod p (outerClassPoles K p 0) R) (- 1)

```

proof –
have p : prime p **and** $p7$: $7 \leq p$ **using** $assms(1,2)$ **by** ($simp$ -all add: OuterHyp-def)
have C : outerClassPoles K p $0 = \{p\}$ **using** zero-class[OF $assms(1,2)$] $assms(3)$
by $simp$
show ?thesis **unfolding** C
proof (rule muXmod-vge-of-residues[OF p $p7$ - $assms(4)$])
show finite $\{p\}$ **by** $simp$
show $(-1 :: \text{rat}) \leq 0$ **by** $simp$
fix j **assume** $j \in \{p\}$
hence $j: j = p$ **by** $simp$
have vge p ($sqResidue$ $\{p\}$ R p) 0
unfolding $sqResidue$ -def **by** ($simp$ add: eval-vge0[OF p $assms(4)$] vge -neg-sq[OF p])
from $vpGge$ -mul[OF p $vpGge$ -C[OF this] $muPole$ -vge-mult[OF p $p7$, of 1]]
show $vpGge$ p ($[sqResidue$ $\{p\}$ R $j]$ * $muPole$ j) (-1) **unfolding** j **by** $simp$
qed
qed

The zero class with two poles p , $2p$.

lemma entry-zero-two:

assumes prime p OuterHyp K p $2 * p \leq K$ $vpGge$ p A 0
shows $vpGge$ p ($muXmod$ p ($outerClassPoles$ K p 0) $(A * (pX + [:of-nat$ p $\wedge$ $2:] \wedge i * (pX +$
 $[:of-nat$ p $\wedge 2:] \wedge i')$
 $(\min 0 (2 * (of-nat$ $i + of-nat$ $i') - 3)))$

proof –

have p : prime p **using** $assms(1)$.
have $p7$: $7 \leq p$ **using** $assms(2)$ **by** ($simp$ add: OuterHyp-def)
have C : outerClassPoles K p $0 = \{p, 2 * p\}$ **using** zero-class[OF $assms(1,2)$] $assms(3)$ **by** $simp$
have Yi : $vpGge$ p $((pX + [:of-nat$ p $\wedge 2:] \wedge n) 0$ **for** n
by (rule $vpGge$ -pow0[OF p $vpGge$ -pX-plus-sq[OF p]])
have $R0$: $vpGge$ p $(A * (pX + [:of-nat$ p $\wedge 2:] \wedge i * (pX + [:of-nat$ p $\wedge 2:] \wedge$
 $i') 0$
by (rule $vpGge$ -mul0[OF p $vpGge$ -mul0[OF p $assms(4)$] Yi] Yi)
have ev : $poly$ $(pX + [:of-nat$ p $\wedge 2:] \wedge (of-nat$ $(t * p) \wedge 2)) =$
 $(of-nat$ $p \wedge 2 * of-int$ $(1 - int$ $t \wedge 2) :: \text{rat})$ **for** t
by ($simp$ add: poly-monom algebra-simps power-mult-distrib)
have $v2$: vge p $(of-nat$ $p \wedge 2 * of-int$ $z) 2$ **for** $z :: \text{int}$
using vge -mul[OF p vge -pow-mul[OF p vge -p-self[OF p], of 2] vge -of-int[OF p ,
of z] **by** $simp$
have key: $vpGge$ p ($[sqResidue$ $\{p, 2 * p\}$ $(A * (pX + [:of-nat$ p $\wedge 2:] \wedge i * (pX$
 $+$
 $[:of-nat$ $p \wedge 2:] \wedge i') (t * p):] * muPole$ $(t * p))$
 $(\min 0 (2 * (of-nat$ $i + of-nat$ $i') - 3)))$ **if** $t: t = 1 \vee t = 2$ **for** t
proof –
have x : vge p $(- (of-nat$ $(t * p) \wedge 2)) 0$ **by** (rule vge -neg-sq[OF p])
have Yv : vge p $(poly$ $(pX + [:of-nat$ p $\wedge 2:] \wedge (of-nat$ $(t * p) \wedge 2))) 2$

```

unfolding ev
  by (rule v2)
  have num: vge p (poly (A * (pX + [:of-nat p ^ 2:]) ^ i * (pX +
    [:of-nat p ^ 2:]) ^ i') (- (of-nat (t * p) ^ 2)))
    (0 + of-nat i * 2 + of-nat i' * 2)
  unfolding poly-mult poly-power
  by (rule vge-mul[OF p vge-mul[OF p eval-vge0[OF p assms(4) x] vge-pow-mul[OF
p Yv]]
    vge-pow-mul[OF p Yv]])
  define D where D = (∏ k ∈
    {p, 2 * p} - {t * p}. of-nat k ^ 2 - of-nat (t * p) ^ 2 :: rat)
  have S: {p, 2 * p} - {t * p} = (if t = 1 then {2 * p} else {p}) using t p7
by auto
  have D: D = 3 * of-nat p ^ 2 ∨ D = - (3 * of-nat p ^ 2)
  unfolding D-def S using t by (auto simp: algebra-simps power2-eq-square)
  have D0: D ≠ 0 using D p7 by auto
  have Dv: of-int (vp p D) ≤ (2 :: rat)
  using D vp-three-sq[OF p p7] by (auto simp: vp-def qmultiplicity-minus)
  have r: vge p (sqResidue {p, 2 * p} (A * (pX + [:of-nat p ^ 2:]) ^ i * (pX +
    [:of-nat p ^ 2:]) ^ i') (t * p))
    (0 + of-nat i * 2 + of-nat i' * 2 - 2)
  unfolding sqResidue-def D-def[symmetric] by (rule vge-div-of-le[OF p num
D0 Dv])
  have vpGge p ([:sqResidue {p, 2 * p} (A * (pX + [:of-nat p ^ 2:]) ^ i * (pX +
    [:of-nat p ^ 2:]) ^ i') (t * p):] * muPole (t * p))
    ((0 + of-nat i * 2 + of-nat i' * 2 - 2) + - 1)
  by (rule vpGge-mul[OF p vpGge-C[OF r] muPole-vge-mult[OF p p7]]) (use t
in auto)
  moreover have min 0 (2 * (of-nat i + of-nat i') - 3) ≤
    (0 + of-nat i * 2 + of-nat i' * 2 - 2 + - 1 :: rat)
  by (simp add: min-def)
  ultimately show ?thesis by (rule vpGge-mono)
qed
show ?thesis unfolding C
proof (rule muXmod-vge-of-residues[OF p p7 - R0 min.cobounded1])
  show finite {p, 2 * p} by simp
  fix j assume j ∈ {p, 2 * p}
  hence j = 1 * p ∨ j = 2 * p by auto
  thus vpGge p ([:sqResidue {p, 2 * p} (A * (pX + [:of-nat p ^ 2:]) ^ i * (pX +
    [:of-nat p ^ 2:]) ^ i') j:] * muPole j)
    (min 0 (2 * (of-nat i + of-nat i') - 3)) using key[of 1] key[of 2] by auto
qed
qed

```

22.7 The entry bound (4.12)

lemma *min-add-le-min*: *min 0 A + min 0 B ≤ min 0 (A + B)* **for** *A B :: rat*
by (*simp add: min-def*)

The cancellation behind a same-class entry.

lemma *same-class-cancel*:
assumes *finite* Ca $Ca \subseteq \text{tailPoles } K$
shows $\text{muXmod } p \{1..K\} (Dm (Nof K) \wedge 6 * (\text{sqPoleProd } (\text{tailPoles } K - Ca) * A) * (\text{sqPoleProd } (\text{tailPoles } K - Ca) * B)) =$
 $\text{muXmod } p Ca (Dm (Nof K) \wedge 5 * \text{sqPoleProd } (\text{tailPoles } K - Ca) * A * B)$
proof –
define T **where** $T = \{1..Nof K\} \cup (\text{tailPoles } K - Ca)$
have $Nof K \leq K$ **by** (*simp add: Nof-def*)
with *assms*(2) **have** $u: \{1..K\} = T \cup Ca$ **by** (*auto simp: T-def tailPoles-def*)
have *fT*: *finite* T **by** (*simp add: T-def tailPoles-def*)
have *dTC*: $T \cap Ca = \{\}$ **using** *assms*(2) **by** (*auto simp: T-def tailPoles-def*)
have *sT*: $\text{sqPoleProd } T = Dm (Nof K) * \text{sqPoleProd } (\text{tailPoles } K - Ca)$
unfolding *T-def* **by** (*subst sqPoleProd-union*) (*auto simp: tailPoles-def Dm-def sqPoleProd-def*)
have *d6*: $Dm (Nof K) \wedge 6 = Dm (Nof K) \wedge 5 * Dm (Nof K)$ **using** *power-Suc2*[*of* $Dm (Nof K) 5$]
by *simp*
have $n: Dm (Nof K) \wedge 6 * (\text{sqPoleProd } (\text{tailPoles } K - Ca) * A) * (\text{sqPoleProd } (\text{tailPoles } K - Ca) * B) =$
 $(Dm (Nof K) \wedge 5 * \text{sqPoleProd } (\text{tailPoles } K - Ca) * A * B) * \text{sqPoleProd } T$
unfolding *sT d6* **by** (*simp add: mult-ac*)
show *?thesis* **unfolding** $u n$ **by** (*rule muXmod-mul-cancel-set*[*OF fT assms*(1) *dTC*])
qed

The cancellation behind a cross-class entry.

lemma *cross-class-cancel*:
assumes *finite* Ca *finite* Cc $Ca \cap Cc = \{\}$ $Ca \subseteq \text{tailPoles } K$ $Cc \subseteq \text{tailPoles } K$
shows $\text{muXmod } p \{1..K\} (Dm (Nof K) \wedge 6 * (\text{sqPoleProd } (\text{tailPoles } K - Ca) * A) * (\text{sqPoleProd } (\text{tailPoles } K - Cc) * B)) =$
 $[\text{muModPoly } p (Dm (Nof K) \wedge 5 * \text{sqPoleProd } (\text{tailPoles } K - (Ca \cup Cc)) * A * B):]$
proof –
define g **where** $g = (\lambda j. pX + [:of\text{-nat } j \wedge 2:] :: \text{rat poly})$
have *ft*: *finite* $(\text{tailPoles } K - X)$ **for** X **by** (*simp add: tailPoles-def*)
have $U: (\text{tailPoles } K - Ca) \cup (\text{tailPoles } K - Cc) = \text{tailPoles } K$ **using** *assms*(3)
by *auto*
have $I: (\text{tailPoles } K - Ca) \cap (\text{tailPoles } K - Cc) = \text{tailPoles } K - (Ca \cup Cc)$
by *auto*
have $ui: \text{sqPoleProd } (\text{tailPoles } K - Ca) * \text{sqPoleProd } (\text{tailPoles } K - Cc) =$
 $\text{sqPoleProd } (\text{tailPoles } K) * \text{sqPoleProd } (\text{tailPoles } K - (Ca \cup Cc))$
proof –
have $sg: \text{sqPoleProd } S = \text{prod } g S$ **for** S **by** (*simp add: sqPoleProd-def g-def*)
have $\text{prod } g ((\text{tailPoles } K - Ca) \cup (\text{tailPoles } K - Cc)) * \text{prod } g ((\text{tailPoles } K - (Ca \cup Cc))) =$

```

—
  Ca)  $\cap$  (tailPoles K - Cc)) =
  prod g (tailPoles K - Ca) * prod g (tailPoles K - Cc) by (rule prod.union-inter[OF
ft ft])
  thus ?thesis unfolding sg U I by (rule sym)
qed
have d6: Dm (Nof K)  $\wedge$  6 = Dm (Nof K)  $\wedge$  5 * Dm (Nof K) using power-Suc2[of
Dm (Nof K) 5]
by simp
define X where X = Dm (Nof K)  $\wedge$  5 * sqPoleProd (tailPoles K - (Ca  $\cup$  Cc))
* A * B
have Dm (Nof K)  $\wedge$  6 * (sqPoleProd (tailPoles K - Ca) * A) * (sqPoleProd
(tailPoles K - Cc) * B) =
  Dm (Nof K)  $\wedge$  5 * Dm (Nof K) * (sqPoleProd (tailPoles K - Ca) * sqPoleProd
(tailPoles K -
  Cc)) * A * B
unfolding d6 by (simp add: mult-ac)
also have ... = Dm (Nof K)  $\wedge$  5 * Dm (Nof K) * (sqPoleProd (tailPoles K) *
sqPoleProd (tailPoles K - (Ca  $\cup$  Cc))) * A * B
by (simp only: ui)
also have ... = X * (Dm (Nof K) * sqPoleProd (tailPoles K)) by (simp add:
X-def mult-ac)
also have ... = X * sqPoleProd {1..K} by (simp only: sqPoleProd-Icc-split)
finally have n: Dm (Nof K)  $\wedge$  6 * (sqPoleProd (tailPoles K - Ca) * A) *
(sqPoleProd (tailPoles K -
  Cc) * B) = X * sqPoleProd {1..K} .
have muXmod p ({1..K}  $\cup$  {}) (X * sqPoleProd {1..K}) = muXmod p {} X
by (rule muXmod-mul-cancel-set) simp-all
thus ?thesis unfolding n X-def[symmetric] by (simp add: muXmod-empty)
qed

```

(4.12) and the zero-class weights: entries of A in the outer basis satisfy the weights. The longest Lean file of the proof (OuterEntries.lean, 1,045 lines): cross-class, ordinary and E - a rows, the two-node congruence $H\text{-}\{p-a\} \equiv H\text{-}\{a-1\} \pmod{p}$, the zero class.

theorem outerGram-entry-bound:

```

assumes p: prime p and K: 40 dvd K and H: OuterHyp K p and x: x <
  length (outer-idx K p) and y: y < length (outer-idx K p)
shows vpGge p (outerGram K p $$ (x, y)) (wOut K p (outer-idx K p ! x) +
  wOut K p (outer-idx K p ! y))
proof —
define ix where ix = outer-idx K p ! x
define iy where iy = outer-idx K p ! y
define a where a = fst ix
define c where c = fst iy
have p7: 7  $\leq$  p using H by (simp add: OuterHyp-def)
have ix: ix  $\in$  set (outer-idx K p) and iys: iy  $\in$  set (outer-idx K p)
by (simp-all add: ix-def iy-def x y)
have am: a  $\leq$  mHalf p and xi: snd ix < outerCount K p a and cm: c  $\leq$ 

```

```

    mHalf p and yi: snd iy < outerCount K p c
    using mem-outer-idx ixs iys by (auto simp: a-def c-def)
    have w0: wOut K p ix + wOut K p iy ≤ 0 using wOut-nonpos[of K p ix]
    wOut-nonpos[of K p iy]
    by linarith
    have finC:  $\bigwedge a.$  finite (outerClassPoles K p a) and subC:  $\bigwedge a.$  outerClassPoles
    K p a
     $\subseteq$  tailPoles K
    by (auto simp: outerClassPoles-def tailPoles-def)
    have entry: outerGram K p $$ (x, y) =
    muXmod p {1..K} (Dm (Nof K) ^ 6 * (sqPoleProd (tailPoles K - outer-
    ClassPoles K p a) *
    oq K p ix) * (sqPoleProd (tailPoles K - outerClassPoles K p c) * oq K p iy))
    using outerGram-apply[OF x y] outerBasis-eq[of K p ix] outerBasis-eq[of K p
    iy]
    by (simp add: ix-def iy-def a-def c-def)
    have int0: vpGge p (Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outer-
    ClassPoles K p a)) 0
    by (intro vpGge-mul0[OF p] vpGge-pow0[OF p] vpGge-Dm[OF p] vpGge-sqPoleProd[OF
    p])
    show ?thesis
    proof (cases a = c)
    case False
    — cross-class: the entry is a constant polynomial with an integral value
    have outerGram K p $$ (x, y) = [:muModPoly p (Dm (Nof K) ^ 5 * sqPoleProd
    (tailPoles K -
    (outerClassPoles K p a  $\cup$  outerClassPoles K p c)) * oq K p ix * oq K p iy):]
    unfolding entry by (rule cross-class-cancel[OF finC finC class-disjoint[OF p
    H am cm False]
    subC subC])
    thus ?thesis
    unfolding ix-def[symmetric] iy-def[symmetric]
    using vpGge-mono[OF vpGge-C[OF muModPoly-vge[OF p p7
    vpGge-mul0[OF p vpGge-mul0[OF p vpGge-mul0[OF p vpGge-pow0[OF p
    vpGge-Dm[OF p]]
    vpGge-sqPoleProd[OF p]] oq-vge[OF p]] oq-vge[OF p]]]] w0]
    by simp
    next
    case True
    — same class
    have same: outerGram K p $$ (x, y) =
    muXmod p (outerClassPoles K p a) (Dm (Nof K) ^ 5 * sqPoleProd (tailPoles
    K -
    outerClassPoles K p a) * oq K p ix * oq K p iy)
    unfolding entry True[symmetric] by (rule same-class-cancel[OF finC subC])
    show ?thesis
    proof (cases a = 0)
    case a0: True
    — the zero class

```

```

have oqx: oq K p ix = (pX + [:of-nat p ^ 2:]) ^ snd ix and ogy: oq K p iy =
  (pX + [:of-nat p ^ 2:]) ^ snd iy
using a0 True by (simp-all add: oq-def Let-def a-def c-def)
have cnt: outerCount K p 0 = (if 2 * p ≤ K then 2 else 1)
using p7 by (simp add: outerCount-def zero-class[OF p H])
show ?thesis
proof (cases 2 * p ≤ K)
case True2: True
have bnd: vpGge p (outerGram K p $$ (x,
  y)) (min 0 (2 * (of-nat (snd ix) + of-nat (snd iy)) - 3))
  unfolding same oqx ogy a0 by (rule entry-zero-two[OF p H True2
int0[unfolded a0]])
have wx: wOut K p ix = (if snd ix = 0 then - 2 else 0) and wy: wOut K
p iy = (if snd iy =
  0 then - 2 else 0)
using a0 True cnt True2 by (simp-all add: wOut-def Let-def a-def c-def)
have snd ix < 2 snd iy < 2 using xi yi cnt True2 a0 True by simp-all
thus ?thesis unfolding ix-def[symmetric] iy-def[symmetric]
  by (intro vpGge-mono[OF bnd]) (auto simp: wx wy)
next
case False2: False
have bnd: vpGge p (outerGram K p $$ (x, y)) (- 1)
  unfolding same a0 oqx ogy
  by (rule entry-zero-one[OF p H - vpGge-mul0[OF p vpGge-mul0[OF p
int0[unfolded a0]]
  vpGge-pow0[OF p vpGge-pX-plus-sq[OF p]]]
  vpGge-pow0[OF p vpGge-pX-plus-sq[OF p]]]) (use False2 in simp)
have wx: wOut K p ix = - 1 / 2 and wy: wOut K p iy = - 1 / 2
  using a0 True cnt False2 by (simp-all add: wOut-def Let-def a-def c-def)
show ?thesis unfolding ix-def[symmetric] iy-def[symmetric] wx wy
  by (rule vpGge-mono[OF bnd]) simp
qed
next
case a1': False
— an ordinary class
have a1: 1 ≤ a using a1' by simp
define l where l = ell p K a
show ?thesis
proof (cases snd ix < l - 2 ∧ snd iy < l - 2)
case both: True
have oqx: oq K p ix = (pX + [:of-nat a ^ 2:]) ^ snd ix and ogy: oq K p iy
=
  (pX + [:of-nat a ^ 2:]) ^ snd iy
  using both a1' True by (simp-all add: oq-def Let-def a-def c-def l-def)
have bnd: vpGge p (outerGram K p $$ (x,
  y)) (min 0 (of-nat (snd ix) + of-nat (snd iy) + 6 * dlt K a - of-nat (ell
p K a) - 4))
  unfolding same oqx ogy by (rule entry-ord[OF p H a1 am])
have wx: wOut K p ix = min 0 (of-nat (snd ix) + 3 * dlt K a - (of-nat

```

```

(ell p K a) + 4) /
2)
and wy: wOut K p iy = min 0 (of-nat (snd iy) + 3 * dlt K a - (of-nat
(ell p K a) + 4) /
2)
using both a1' True by (simp-all add: wOut-def Let-def a-def c-def l-def
dlt-def)
show ?thesis unfolding ix-def[symmetric] iy-def[symmetric] wx wy
by (rule vpGge-mono[OF bnd]) (rule order-trans[OF min-add-le-min],
simp add: algebra-simps)
next
case False
— one of the rows has the factor  $E$ -a: the entry is integral
have vpGge p (outerGram K p $$ (x, y)) 0
unfolding same
proof —
define E where E = sqPoleProd {j ∈ outerClassPoles K p a. p < j}
have Q: vpGge p ((pX + [:of-nat a ^ 2:]) ^ n) 0 for n
by (rule vpGge-pow0[OF p vpGge-pX-plus-sq[OF p]])
show vpGge p (muXmod p (outerClassPoles K p a) (Dm (Nof K) ^ 5 *
sqPoleProd (tailPoles K - outerClassPoles K p a) * oq K p ix * oq K p
iy)) 0
proof (cases snd ix < l - 2)
case big: False
have o: oq K p ix = E * (pX + [:of-nat a ^ 2:]) ^ (snd ix - (l - 2))
using big a1' by (simp add: oq-def Let-def E-def l-def a-def)
have r: Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outerClassPoles
K p a) * oq K p ix *
oq K p iy =
(Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outerClassPoles K p
a) * (pX +
[:of-nat a ^ 2:]) ^ (snd ix - (l - 2)) * oq K p ix) * E
unfolding o by (simp add: mult-ac)
show ?thesis unfolding r E-def
by (rule entry-large[OF p H a1 am vpGge-mul0[OF p vpGge-mul0[OF
p int0 Q] oq-vge[OF
p]]])
next
case small: True
hence big: ¬ snd iy < l - 2 using False by simp
have o: oq K p iy = E * (pX + [:of-nat a ^ 2:]) ^ (snd iy - (l - 2))
using big a1' True by (simp add: oq-def Let-def E-def l-def c-def)
have r: Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outerClassPoles
K p a) * oq K p ix *
oq K p iy =
(Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outerClassPoles K p
a) * oq K p ix *
(pX + [:of-nat a ^ 2:]) ^ (snd iy - (l - 2))) * E
unfolding o by (simp add: mult-ac)

```



```

      show ?thesis unfolding r E-def
      by (rule entry-large[OF p H a1 am vpGge-mul0[OF p vpGge-mul0[OF
p int0 oq-vge[OF p]]
      Q]])
    qed
  qed
  thus ?thesis unfolding ix-def[symmetric] iy-def[symmetric] by (rule
vpGge-mono) (rule w0)
  qed
  qed
  qed
  qed

```

22.8 The closed form (4.14) of $\gamma\text{-}p^{\wedge}\text{out}$

definition $\text{ind} :: \text{bool} \Rightarrow \text{nat}$ **where** $\text{ind } P = (\text{if } P \text{ then } 1 \text{ else } 0)$

lemma ind-le-one : $\text{ind } P \leq 1$ **by** (simp add: ind-def)
lemma ind-True : $P \implies \text{ind } P = 1$ **by** (simp add: ind-def)
lemma ind-False : $\neg P \implies \text{ind } P = 0$ **by** (simp add: ind-def)
lemma ind-mult : $\text{ind } P * \text{ind } Q = \text{ind } (P \wedge Q)$ **by** (simp add: ind-def)

Membership in a class $\pm a \pmod{p}$ for $j < 3p$.

lemma class-char :

```

  fixes j a p :: nat
  assumes 1 ≤ a < p < 3 * p
  shows (j mod p = a mod p ∨ (j + a) mod p = 0) ⟷
    (j = a ∨ j = p + a ∨ j = 2 * p + a ∨ j + a = p ∨ j + a = 2 * p ∨
    j + a = 3 * p)

```

proof –

```

  have p0: 0 < p using assms by simp
  define r where r = j mod p
  define q where q = j div p
  have e: j = q * p + r by (simp add: q-def r-def)
  have rp: r < p using p0 by (simp add: r-def)
  have q3: q < 3 using assms(3) p0 by (simp add: q-def div-less-iff-less-mult)
  have L: (j mod p = a mod p ∨ (j + a) mod p = 0) ⟷ (r = a ∨
    r = p - a)
    using mod-add-eq-0-iff[of a p j] assms by (simp add: r-def)
  have q = 0 ∨ q = 1 ∨ q = 2 using q3 by auto
  thus ?thesis unfolding L using e rp assms
  proof (elim disjE)
    assume q: q = 0
    have j = r using e q by auto
    moreover have r = p - a ⟷ r + a = p using assms rp by linarith
    moreover have r ≠ p + a ∧ r ≠ 2 * p + a ∧ r + a ≠ 2 * p ∧ r + a ≠
      3 * p using assms rp by auto
    ultimately show (r = a ∨ r = p - a) = (j = a ∨ j = p + a ∨ j = 2 * p +
a ∨

```

```

     $j + a = p \vee j + a = 2 * p \vee j + a = 3 * p$ ) by fast
next
  assume  $q: q = 1$ 
  have  $j = p + r$  using  $e\ q$  by simp
  moreover have  $r = p - a \longleftrightarrow r + a = p$  using  $assms\ rp$  by linarith
  moreover have  $p + r \neq a \wedge p + r = p + a \longleftrightarrow$ 
     $r = a \wedge p + r \neq 2 * p + a \wedge p + r + a \neq p$ 
     $p + r + a = 2 * p \longleftrightarrow r + a = p \wedge p + r + a \neq 3 * p$  using  $assms\ rp$ 
    by auto
  ultimately show  $(r = a \vee r = p - a) = (j = a \vee j = p + a \vee j = 2 * p +$ 
 $a \vee$ 
     $j + a = p \vee j + a = 2 * p \vee j + a = 3 * p)$  by fast
next
  assume  $q: q = 2$ 
  have  $j = 2 * p + r$  using  $e\ q$  by simp
  moreover have  $r = p - a \longleftrightarrow r + a = p$  using  $assms\ rp$  by linarith
  moreover have  $2 * p + r \neq a \wedge 2 * p + r \neq p + a \wedge 2 * p + r =$ 
     $2 * p + a \longleftrightarrow r = a \wedge 2 * p + r + a \neq p$ 
     $2 * p + r + a \neq 2 * p \wedge 2 * p + r + a = 3 * p \longleftrightarrow r + a = p$ 
    using  $assms\ rp$  by auto
  ultimately show  $(r = a \vee r = p - a) = (j = a \vee j = p + a \vee j = 2 * p +$ 
 $a \vee$ 
     $j + a = p \vee j + a = 2 * p \vee j + a = 3 * p)$  by fast
qed
qed

```

lemma *sum6*:

```

  fixes  $f :: nat \Rightarrow nat$ 
  assumes  $distinct\ [x1, x2, x3, x4, x5, x6]$ 
  shows  $(\sum x \in \{x1, x2, x3, x4, x5, x6\}. f\ x) = f\ x1 + f\ x2 + f\ x3 + f\ x4 + f\ x5 + f\ x6$ 
  using  $assms$  by simp

```

The class $\pm a$ in $\{L..K\}$, as six indicators.

lemma *card-class*:

```

  fixes  $L\ K\ a\ p :: nat$ 
  assumes  $1 \leq a \wedge 2 * a < p \wedge K < 3 * p$ 
  shows  $card\ \{j \in \{L..K\}. j\ mod\ p = a\ mod\ p \vee (j + a)\ mod\ p = 0\} =$ 
     $ind\ (L \leq a \wedge a \leq K) + ind\ (L \leq p - a \wedge$ 
     $p - a \leq K) + ind\ (L \leq p + a \wedge p + a \leq K) +$ 
     $ind\ (L \leq 2 * p - a \wedge 2 * p - a \leq K) + ind\ (L \leq 2 * p + a \wedge$ 
     $2 * p + a \leq K) + ind\ (L \leq 3 * p - a \wedge 3 * p - a \leq K)$ 

```

proof –

```

  define  $A$  where  $A = \{a, p - a, p + a, 2 * p - a, 2 * p + a, 3 * p - a\}$ 
  have  $ap: a < p$  using  $assms(2)$  by linarith
  have  $conv: (j + a = p) = (j = p - a) \wedge (j + a = 2 * p) = (j = 2 * p - a) \wedge (j +$ 
 $a = 3 * p) = (j =$ 
     $3 * p - a)$  for  $j$ 
  using  $ap$  by auto

```

```

have S: {j ∈ {L..K}. j mod p = a mod p ∨
(j + a) mod p = 0} = {j ∈ A. L ≤ j ∧ j ≤ K}
proof (intro equalityI subsetI)
  fix j assume j: j ∈ {j ∈ {L..K}. j mod p = a mod p ∨ (j + a) mod p = 0}
  have j3: j < 3 * p using j assms(3) by auto
  have j = a ∨ j = p + a ∨ j = 2 * p + a ∨ j + a = p ∨ j + a = 2 * p ∨
    j + a = 3 * p
    using class-char[OF assms(1) ap j3] j by force
  thus j ∈ {j ∈ A. L ≤ j ∧ j ≤ K} using j conv unfolding A-def by force
next
fix j assume j: j ∈ {j ∈ A. L ≤ j ∧ j ≤ K}
have j3: j < 3 * p using j assms(1,2,3) unfolding A-def by force
have j = a ∨ j = p + a ∨ j = 2 * p + a ∨ j + a = p ∨ j + a = 2 * p ∨
  j + a = 3 * p
  using j assms(1,2) conv unfolding A-def by blast
thus j ∈ {j ∈ {L..K}. j mod p = a mod p ∨ (j + a) mod p = 0}
  using class-char[OF assms(1) ap j3] j by force
qed
have C: card {j ∈ A. L ≤ j ∧ j ≤ K} = (∑ x ∈ A. ind (L ≤ x ∧
x ≤ K))
proof -
  have fA: finite A unfolding A-def by simp
  show ?thesis unfolding ind-def
    using sum.inter-filter[OF fA, of λ-. 1::nat λx. L ≤ x ∧
x ≤ K] card-eq-sum[of {j ∈ A. L ≤ j ∧ j ≤ K}] by linarith
qed
have dist: a ≠ p - a a ≠ p + a a ≠ 2 * p - a a ≠
  2 * p + a a ≠ 3 * p - a
  p - a ≠ p + a p - a ≠ 2 * p - a p - a ≠ 2 * p + a p - a ≠
  3 * p - a
  p + a ≠ 2 * p - a p + a ≠ 2 * p + a p + a ≠ 3 * p - a
  2 * p - a ≠ 2 * p + a 2 * p - a ≠ 3 * p - a 2 * p + a ≠ 3 * p - a
  using assms(1,2) by auto
have D: (∑ x ∈ A. ind (L ≤ x ∧ x ≤ K)) =
  ind (L ≤ a ∧ a ≤ K) + ind (L ≤ p - a ∧
  p - a ≤ K) + ind (L ≤ p + a ∧ p + a ≤ K) +
  ind (L ≤ 2 * p - a ∧ 2 * p - a ≤ K) + ind (L ≤ 2 * p + a ∧
  2 * p + a ≤ K) + ind (L ≤ 3 * p - a ∧ 3 * p - a ≤ K)
  unfolding A-def by (rule sum6) (use dist in simp)
show ?thesis unfolding S C D ..
qed

lemma ell-eq:
  assumes prime p OuterHyp K p 1 ≤ a a ≤ mHalf p
  shows ell p K a = (if K < 2 * p then 2 else 4) + ind (a ≤ K mod p) + ind (p
≤
  K mod p + a)
proof -
  have of: 2 * mHalf p + 1 = p ∧ p ≤ K ∧ K < 3 * p ∧

```

```

    (K < 2 * p → K mod p = K - p) ∧ (2 * p ≤ K →
    K mod p = K - 2 * p)
    using outer-facts[OF assms(1,2)] by (force simp: ind-def)
  have m: 2 * mHalf p + 1 = p and pK: p ≤ K and K3: K < 3 * p using of
    by (auto simp: ind-def)
  have a2: 2 * a < p using m assms(4) by (simp add: ind-def)
  have cc: ell p K a = ind (1 ≤ a ∧ a ≤ K) + ind (1 ≤ p - a ∧
    p - a ≤ K) + ind (1 ≤ p + a ∧ p + a ≤ K) +
    ind (1 ≤ 2 * p - a ∧ 2 * p - a ≤ K) + ind (1 ≤ 2 * p + a ∧
    2 * p + a ≤ K) + ind (1 ≤ 3 * p - a ∧ 3 * p - a ≤ K)
    unfolding ell-def by (rule card-class[OF assms(3) a2 K3])
  have i1: ind (1 ≤ a ∧ a ≤ K) = 1 using assms(3) a2 pK by (force simp:
ind-def)
  have i2: ind (1 ≤ p - a ∧ p - a ≤ K) = 1 using assms(3) a2 pK
    by (force simp: ind-def)
  show ?thesis
  proof (cases K < 2 * p)
    case True
      have v: K mod p = K - p using of True by (fastforce simp: ind-def)
      have (1 ≤ p + a ∧ p + a ≤ K) ↔ a ≤ K mod p
        using v pK a2 assms(3) by linarith
      hence i3: ind (1 ≤ p + a ∧ p + a ≤ K) = ind (a ≤ K mod p) by presburger
      have i4: ind (1 ≤ 2 * p - a ∧ 2 * p - a ≤ K) = ind (p ≤ K mod p + a)
        using v pK a2 by (fastforce simp: ind-def)
      have i5: ind (1 ≤ 2 * p + a ∧ 2 * p + a ≤ K) = 0 using True
        by (auto simp: ind-def)
      have i6: ind (1 ≤ 3 * p - a ∧ 3 * p - a ≤ K) = 0 using True a2
        by (force simp: ind-def)
      show ?thesis unfolding cc i1 i2 i3 i4 i5 i6 using True by presburger
    next
      case False
      have v: K mod p = K - 2 * p using of False by (auto simp: ind-def)
      have i3: ind (1 ≤ p + a ∧ p + a ≤ K) = 1 using False a2 by (force simp:
ind-def)
      have i4: ind (1 ≤ 2 * p - a ∧ 2 * p - a ≤ K) = 1 using False a2
        by (auto simp: ind-def)
      have (1 ≤ 2 * p + a ∧ 2 * p + a ≤ K) ↔ a ≤ K mod p
        using v False a2 assms(3) by linarith
      hence i5: ind (1 ≤ 2 * p + a ∧ 2 * p + a ≤ K) = ind (a ≤ K mod p)
        by presburger
      have i6: ind (1 ≤ 3 * p - a ∧ 3 * p - a ≤ K) = ind (p ≤ K mod p + a)
        using v False a2 by (force simp: ind-def)
      show ?thesis unfolding cc i1 i2 i3 i4 i5 i6 using False by presburger
  qed
qed

lemma outerCount-add:
  assumes prime p OuterHyp K p 1 ≤ a a ≤ mHalf p
  shows outerCount K p a + ind (a ≤ Nof K) = ell p K a

```

using *card-outerClass*[*OF assms*] **by** (*simp add: outerCount-def ind-def*)

lemma *outerCount-zero*:

assumes *prime p OuterHyp K p*

shows *outerCount K p 0 = (if K < 2 * p then 1 else 2)*

using *outer-facts*[*OF assms*] **by** (*simp add: outerCount-def zero-class*[*OF assms*])

The weight (4.12) of row i of a class with l poles and $\delta = d$.

definition *Wc* :: *nat* $\Rightarrow$ *nat* $\Rightarrow$ *nat* $\Rightarrow$ *rat* **where**

*Wc l d i = (if i < l - 2 then min 0 (of-nat i + 3 * of-nat d - (of-nat l + 4) / 2) else 0)*

The two finite class computations: classes with $2 + A + B$ and $4 + A + B$ poles.

lemma *card-lt-Suc*: *card {i. i < Suc n $\wedge$ P i} = card {i. i < n $\wedge$ P i} + (if P n then 1 else 0)*

proof –

have *e*: *{i. i < Suc n $\wedge$ P i} = (if P n then insert n {i. i < n $\wedge$ P i} else {i. i < n $\wedge$ P i})*

by (*auto simp: less-Suc-eq*)

show *?thesis* **unfolding** *e* **by** *simp*

qed

lemma *cls2*:

assumes *A $\leq$ 1 B $\leq$ 1 d $\leq$ 1*

shows *2 * ($\sum i < 2 + A + B - d. Wc (2 + A + B) d i$) =*

*– 7 * (of-nat A + of-nat B) + 6 * of-nat d * (of-nat A + of-nat B) $\wedge$*

*card {i. i < 2 + A + B - d $\wedge$ Wc (2 + A + B) d i = 0} = 2 - d + d **

*A * B*

proof –

have *A = 0 $\vee$ A = 1 B = 0 $\vee$ B = 1 d = 0 $\vee$ d = 1* **using** *assms* **by** *auto*

thus *?thesis* **by** (*elim disjE*) (*simp-all add: Wc-def eval-nat-numeral card-lt-Suc lessThan-Suc min-def*)

qed

lemma *cls4*:

assumes *A $\leq$ 1 B $\leq$ 1 d $\leq$ 1*

shows *2 * ($\sum i < 4 + A + B - d. Wc (4 + A + B) d i$) =*

*– 14 - 7 * (of-nat A + of-nat B) + of-nat d * (12 + 5 * (of-nat A + of-nat B)) $\wedge$*

*card {i. i < 4 + A + B - d $\wedge$ Wc (4 + A + B) d i = 0} = 2 + d * A **

B

proof –

have *A = 0 $\vee$ A = 1 B = 0 $\vee$ B = 1 d = 0 $\vee$ d = 1* **using** *assms* **by** *auto*

thus *?thesis* **by** (*elim disjE*) (*simp-all add: Wc-def eval-nat-numeral card-lt-Suc lessThan-Suc min-def*)

qed

The weight as a function of naturals.

definition $Wn :: nat \Rightarrow nat \Rightarrow nat \Rightarrow nat \Rightarrow rat$ **where**

$Wn\ K\ p\ a\ i = (if\ a = 0\ then\ (if\ outerCount\ K\ p\ 0 = 1\ then\ -\ 1\ /\ 2\ else\ if\ i = 0\ then\ -\ 2\ else\ 0))$
 $\quad\quad\quad else\ Wc\ (ell\ p\ K\ a)\ (ind\ (a \leq Nof\ K))\ i)$

lemma $wOut\text{-}eq$: $wOut\ K\ p\ x = Wn\ K\ p\ (fst\ x)\ (snd\ x)$
by ($simp\ add$: $wOut\text{-}def\ Wn\text{-}def\ Wc\text{-}def\ ind\text{-}def\ Let\text{-}def$)

lemma $sum\text{-}wOut$: $(\sum\ x \leftarrow outer\text{-}idx\ K\ p.\ wOut\ K\ p\ x) = (\sum\ a \leq mHalf\ p.\ \sum\ i < outerCount\ K\ p\ a.\ Wn\ K\ p\ a\ i)$
unfolding $outer\text{-}idx\text{-}def$
by ($simp\ add$: $map\text{-}concat\ sum\text{-}list\text{-}concat\text{-}aux\ o\text{-}def\ sum\text{-}list\text{-}distinct\text{-}conv\text{-}sum\text{-}set\ wOut\text{-}eq$
 $atLeast0LessThan$
 $lessThan\text{-}Suc\text{-}atMost\ del$: $upt\text{-}Suc$)

lemma $card\text{-}zero\text{-}wOut$:
 $length\ (filter\ (\lambda x.\ wOut\ K\ p\ x = 0)\ (outer\text{-}idx\ K\ p)) = (\sum\ a \leq mHalf\ p.\ card\ \{i.\ i < outerCount\ K\ p\ a \wedge Wn\ K\ p\ a\ i = 0\})$
unfolding $outer\text{-}idx\text{-}def$
by ($simp\ add$: $filter\text{-}concat\ length\text{-}concat\ o\text{-}def\ filter\text{-}map\ sum\text{-}list\text{-}distinct\text{-}conv\text{-}sum\text{-}set\ distinct\text{-}length\text{-}filter$
 $wOut\text{-}eq\ atLeast0LessThan\ lessThan\text{-}Suc\text{-}atMost\ Int\text{-}def\ conj\text{-}commute\ del$:
 $upt\text{-}Suc$)

lemma $class\text{-}low$:

assumes $prime\ p\ OuterHyp\ K\ p\ K < 2 * p\ 1 \leq a\ a \leq mHalf\ p$
shows $2 * (\sum\ i < outerCount\ K\ p\ a.\ Wn\ K\ p\ a\ i) =$
 $- 7 * (of\text{-}nat\ (ind\ (a \leq K\ mod\ p)) + of\text{-}nat\ (ind\ (p \leq K\ mod\ p + a))) + 6 * of\text{-}nat\ (ind\ (a \leq Nof\ K)) * (of\text{-}nat\ (ind\ (a \leq K\ mod\ p)) + of\text{-}nat\ (ind\ (p \leq K\ mod\ p + a))) \wedge$
 $card\ \{i.\ i < outerCount\ K\ p\ a \wedge Wn\ K\ p\ a\ i = 0\} = 2 - ind\ (a \leq Nof\ K) + ind\ (a \leq Nof\ K) * ind\ (a \leq K\ mod\ p) * ind\ (p \leq K\ mod\ p + a)$

proof –

have e : $ell\ p\ K\ a = 2 + ind\ (a \leq K\ mod\ p) + ind\ (p \leq K\ mod\ p + a)$
using $ell\text{-}eq[OF\ assms(1,2,4,5)]\ assms(3)$ **by** $simp$
have oc : $outerCount\ K\ p\ a = 2 + ind\ (a \leq K\ mod\ p) + ind\ (p \leq K\ mod\ p + a) - ind\ (a \leq Nof\ K)$
using $outerCount\text{-}add[OF\ assms(1,2,4,5)]\ e$ **by** $linarith$
have W : $Wn\ K\ p\ a\ i = Wc\ (2 + ind\ (a \leq K\ mod\ p) + ind\ (p \leq K\ mod\ p + a))\ (ind\ (a \leq Nof\ K))\ i$ **for** i
using $assms(4)\ e$ **by** ($simp\ add$: $Wn\text{-}def$)
show $?thesis$ **unfolding** $oc\ W$ **by** ($rule\ cls2[OF\ ind\text{-}le\text{-}one\ ind\text{-}le\text{-}one\ ind\text{-}le\text{-}one]$)
qed

lemma *class-high*:

assumes *prime p OuterHyp K p* $\neg K < 2 * p$ $1 \leq a \leq mHalf\ p$
shows $2 * (\sum i < outerCount\ K\ p\ a. Wn\ K\ p\ a\ i) =$
 $- 14 - 7 * (of\text{-}nat\ (ind\ (a \leq K \bmod p)) + of\text{-}nat\ (ind\ (p \leq$
 $K \bmod p + a))) + of\text{-}nat\ (ind\ (a \leq Nof\ K)) * (12 + 5 * (of\text{-}nat\ (ind$
 $(a \leq$
 $K \bmod p)) + of\text{-}nat\ (ind\ (p \leq K \bmod p + a)))) \wedge$
 $card\ \{i. i < outerCount\ K\ p\ a \wedge$
 $Wn\ K\ p\ a\ i = 0\} = 2 + ind\ (a \leq Nof\ K) * ind\ (a \leq K \bmod p) * ind\ (p$
 $\leq$
 $K \bmod p + a)$

proof –

have *e*: $ell\ p\ K\ a = 4 + ind\ (a \leq K \bmod p) + ind\ (p \leq K \bmod p + a)$
using *ell-eq*[*OF assms*(1,2,4,5)] *assms*(3) **by** *simp*
have *oc*: $outerCount\ K\ p\ a = 4 + ind\ (a \leq K \bmod p) + ind\ (p \leq K \bmod p +$
 $a) - ind\ (a \leq$
 $Nof\ K)$
using *outerCount-add*[*OF assms*(1,2,4,5)] *e* **by** *linarith*
have *W*: $Wn\ K\ p\ a\ i = Wc\ (4 + ind\ (a \leq K \bmod p) + ind\ (p \leq K \bmod p +$
 $a))\ (ind\ (a \leq$
 $Nof\ K))\ i$ **for** *i*
using *assms*(4) *e* **by** (*simp add: Wn-def*)
show *?thesis* **unfolding** *oc W* **by** (*rule cls4*[*OF ind-le-one ind-le-one ind-le-one*])
qed

The interval counts used in (4.14).

lemma *counts*:

assumes *prime p OuterHyp K p*
shows $card\ \{a \in \{1..mHalf\ p\}. a \leq K \bmod p\} + card\ \{a \in \{1..mHalf\ p\}. p \leq$
 $K \bmod p + a\} = K \bmod p \wedge$
 $card\ \{a \in \{1..mHalf\ p\}. a \leq Nof\ K \wedge$
 $a \leq K \bmod p\} = min\ (Nof\ K)\ (K \bmod p) \wedge$
 $card\ \{a \in \{1..mHalf\ p\}. a \leq Nof\ K \wedge$
 $p \leq K \bmod p + a\} = Nof\ K + K \bmod p + 1 - p \wedge$
 $card\ \{a \in \{1..mHalf\ p\}. a \leq Nof\ K\} = Nof\ K \wedge$
 $card\ \{a \in \{1..mHalf\ p\}. (a \leq Nof\ K \wedge a \leq K \bmod p) \wedge$
 $p \leq K \bmod p + a\} = Nof\ K + K \bmod p + 1 - p$

proof –

obtain *m*: $2 * mHalf\ p + 1 = p$ **and** *N*: $2 * Nof\ K < p$ **using** *outer-facts*[*OF*
assms] **by** *blast*
have *p0*: $0 < p$ **using** *m* **by** *linarith*
have *vp*: $K \bmod p < p$ **using** *p0* **by** *simp*
have *Nm*: $Nof\ K \leq mHalf\ p$ **using** *N m* **by** *linarith*
have *vp1*: $Suc\ 0 \leq p - K \bmod p$ **using** *vp* **by** *linarith*
have *e1*: $\{a \in \{1..mHalf\ p\}. a \leq K \bmod p\} = \{1..min\ (mHalf\ p)\ (K \bmod p)\}$
by *auto*
have *e2*: $\{a \in \{1..mHalf\ p\}. p \leq K \bmod p + a\} = \{p - K \bmod p..mHalf\ p\}$
using *vp vp1*
by *auto*

```

have e3: {a ∈ {1..mHalf p}. a ≤ Nof K ∧
  a ≤ K mod p} = {1..min (Nof K) (K mod p)} using Nm by auto
have e4: {a ∈ {1..mHalf p}. a ≤ Nof K ∧
  p ≤ K mod p + a} = {p - K mod p..Nof K} using Nm vp vp1 by auto
have e5: {a ∈ {1..mHalf p}. a ≤ Nof K} = {1..Nof K} using Nm by auto
have e6: {a ∈ {1..mHalf p}. (a ≤ Nof K ∧ a ≤ K mod p) ∧
  p ≤ K mod p + a} = {p - K mod p..Nof K}
  using Nm vp N vp1 by auto
have min (mHalf p) (K mod p) + (Suc (mHalf p) - (p - K mod p)) = K mod p
  using vp m by (cases K mod p ≤ mHalf p; simp add: min-def; linarith)
moreover have Suc (Nof K) - (p - K mod p) = Nof K + K mod p + 1 - p
using vp by linarith
ultimately show ?thesis unfolding e1 e2 e3 e4 e5 e6 card-atLeastAtMost by
simp
qed

```

The table after (4.14): summing (4.12) and applying Lemma 4.2 gives exactly $\gamma\text{-}p\hat{\text{out}}$. Pure combinatorics of the classes (Lean: OuterCount.lean, 444 lines).

```

lemma ind-sum-nat: finite I ⇒
  (∑ a ∈ I. ind (P a)) = card {a ∈ I. P a}
  unfolding ind-def using sum.inter-filter[of I λ-. 1::nat P] by simp

```

```

lemma ind-sum-rat: finite I ⇒
  (∑ a ∈ I. (of-nat (ind (P a)) :: rat)) = of-nat (card {a ∈ I. P a})
  by (simp only: of-nat-sum[symmetric] ind-sum-nat)

```

```

lemma two-minus-ind: 2 - ind P + ind P = (2::nat)
  by (simp add: ind-def)

```

```

lemma gamma-arith-low:
  fixes X :: rat
  assumes S: 2 * X = - 7 * (of-nat c1 + of-nat c2) + 6 * (of-nat c3 + of-nat
c4)
  and c12: c1 + c2 = v and v: v = K - p and pK: p ≤ K and c3: c3 =
    min N v and c4: c4 = u
  and m: 2 * m + 1 = p and Nm: N ≤ 2 * m
  shows (of-int (- 7 * (int K - int p) + 6 * int (min N v + u) - 1 - min (int
r) (int p - 1 -
    int N + int u)) :: rat)
    = 2 * (- 1 / 2 + X) - of-nat (min r (2 * m - N + u))
proof -
  have k: int K - int p = int c1 + int c2 using c12 v pK by simp
  have x: int p - 1 - int N + int u = int (2 * m - N + u) unfolding
m[symmetric] using Nm
  by (simp add: of-nat-diff)
  have mn: min (int r) (int (2 * m - N + u)) = int (min r (2 * m - N + u))
by (simp add: min-def)
  have e: (of-int (- 7 * (int K - int p) + 6 * int (min N v + u) - 1 - min (int

```


$r) (int\ p - 1 -$
 $int\ N + int\ u)) :: rat)$
 $= - 7 * (of-nat\ c1 + of-nat\ c2) + 6 * (of-nat\ c3 + of-nat\ c4) - 1 - of-nat$
 $(min\ r\ (2 * m - N +$
 $u))$
unfolding $k\ x\ mn\ c3\ c4$ **by** *simp*
have $t: 2 * (- 1 / 2 + X) = 2 * X - 1$ **by** *simp*
show *?thesis* **unfolding** $e\ t\ S$ **by** *simp*
qed

lemma *gamma-arith-high*:

fixes $X :: rat$
assumes $S: 2 * X = - 14 * of-nat\ m - 7 * (of-nat\ c1 + of-nat\ c2) + 12 * of-nat\ c5 + 5 * (of-nat\ c3 + of-nat\ c4)$
and $c12: c1 + c2 = v$ **and** $v: v = K - 2 * p$ **and** $pK: 2 * p \leq K$ **and** $c3: c3 = min\ N\ v$ **and** $c4: c4 = u$
and $c5: c5 = N$ **and** $m: 2 * m + 1 = p$
shows $(of-int\ (- 7 * (int\ K - int\ p) + 3 + 12 * int\ N + 5 * int\ (min\ N\ v + u) - min\ (int\ r) (int\ p + int\ u)) :: rat)$
 $= 2 * (- 2 + X) - of-nat\ (min\ r\ (2 * m + 1 + u))$
proof -
have $k: int\ K - int\ p = int\ c1 + int\ c2 + int\ p$ **using** $c12\ v\ pK$ **by** *simp*
have $x: int\ p + int\ u = int\ (2 * m + 1 + u)$ **unfolding** $m[symmetric]$ **by** *simp*
have $mn: min\ (int\ r) (int\ (2 * m + 1 + u)) = int\ (min\ r\ (2 * m + 1 + u))$
by *(simp add: min-def)*
have $mp: (of-nat\ p :: rat) = 2 * of-nat\ m + 1$ **unfolding** $m[symmetric]$ **by** *simp*
have $e: (of-int\ (- 7 * (int\ K - int\ p) + 3 + 12 * int\ N + 5 * int\ (min\ N\ v + u) - min\ (int\ r) (int\ p + int\ u)) :: rat)$
 $= - 7 * (of-nat\ c1 + of-nat\ c2 + of-nat\ p) + 3 + 12 * of-nat\ c5 + 5 * (of-nat\ c3 + of-nat\ c4) - of-nat\ (min\ r\ (2 * m + 1 + u))$
unfolding $k\ x\ mn\ c3\ c4\ c5$ **by** *simp*
have $t: 2 * (- 2 + X) = 2 * X - 4$ **by** *simp*
show *?thesis* **unfolding** $e\ t\ S\ mp$ **by** *(simp add: algebra-simps)*
qed

theorem *gammaOut-eq-sum*:

assumes $p: prime\ p$ **and** $K: 40\ dvd\ K$ **and** $H: OuterHyp\ K\ p$
shows $of-int\ (gammaOut\ K\ p) = 2 * (\sum\ x \leftarrow outer-idx\ K\ p. wOut\ K\ p\ x) - of-nat\ (min\ (rOut\ K\ p) (length\ (filter\ (\lambda x. wOut\ K\ p\ x = 0) (outer-idx\ K\ p))))$
proof -
define m **where** $m = mHalf\ p$
define I **where** $I = \{1..m\}$
define u **where** $u = Nof\ K + K\ mod\ p + 1 - p$

```

obtain c12:  $\text{card } \{a \in I. a \leq K \bmod p\} + \text{card } \{a \in I. p \leq K \bmod p + a\} =$ 
 $K \bmod p$ 
and c3:  $\text{card } \{a \in I. a \leq \text{Nof } K \wedge a \leq K \bmod p\} = \min (\text{Nof } K) (K \bmod p)$ 
and c4:  $\text{card } \{a \in I. a \leq \text{Nof } K \wedge p \leq K \bmod p + a\} = u$ 
and c5:  $\text{card } \{a \in I. a \leq \text{Nof } K\} = \text{Nof } K$ 
and c6:  $\text{card } \{a \in I. (a \leq \text{Nof } K \wedge a \leq K \bmod p) \wedge$ 
 $p \leq K \bmod p + a\} = u$ 
using counts[OF p H] unfolding I-def m-def u-def by blast
have W0:  $(\sum x \leftarrow \text{outer-idx } K p. \text{wOut } K p x) = (\sum i <$ 
 $\text{outerCount } K p 0. \text{Wn } K p 0 i) + (\sum a \in I. \sum i < \text{outerCount } K p a. \text{Wn}$ 
 $K p a i)$ 
proof -
have u:  $\{..m\text{Half } p\} = \text{insert } 0 I \text{ by } (\text{auto simp: I-def m-def})$ 
show ?thesis unfolding sum-wOut u by (subst sum.insert) (simp-all add:
I-def)
qed
have Z0:  $\text{length } (\text{filter } (\lambda x. \text{wOut } K p x = 0) (\text{outer-idx } K p)) = \text{card } \{i. i <$ 
 $\text{outerCount } K p 0 \wedge \text{Wn } K p 0 i = 0\} + (\sum a \in I. \text{card } \{i. i <$ 
 $\text{outerCount } K p a \wedge \text{Wn } K p a i = 0\})$ 
proof -
have u:  $\{..m\text{Half } p\} = \text{insert } 0 I \text{ by } (\text{auto simp: I-def m-def})$ 
show ?thesis unfolding card-zero-wOut u by (subst sum.insert) (simp-all add:
I-def)
qed
show ?thesis
proof (cases  $K < 2 * p$ )
case True
have v:  $K \bmod p = K - p$  using outer-facts[OF p H] True by blast
have oc0:  $\text{outerCount } K p 0 = 1$  by (simp add: outerCount-zero[OF p H]
True)
have z0:  $(\sum i < \text{outerCount } K p 0. \text{Wn } K p 0 i) = - 1 / 2$  and zc:  $\text{card } \{i.$ 
 $i <$ 
 $\text{outerCount } K p 0 \wedge \text{Wn } K p 0 i = 0\} = 0$ 
by (simp-all add: oc0 Wn-def)
have S:  $2 * (\sum a \in I. \sum i < \text{outerCount } K p a. \text{Wn } K p a i) =$ 
 $- 7 * (\text{of-nat } (\text{card } \{a \in I. a \leq K \bmod p\}) + \text{of-nat } (\text{card } \{a \in I. p \leq$ 
 $K \bmod p + a\}))$ 
 $+ 6 * (\text{of-nat } (\text{card } \{a \in I. a \leq \text{Nof } K \wedge$ 
 $a \leq K \bmod p\}) + \text{of-nat } (\text{card } \{a \in I. a \leq \text{Nof } K \wedge p \leq K \bmod p + a\}))$ 
proof -
have cl:  $2 * (\sum i < \text{outerCount } K p a. \text{Wn } K p a i) =$ 
 $- 7 * (\text{of-nat } (\text{ind } (a \leq K \bmod p)) + \text{of-nat } (\text{ind } (p \leq K \bmod p + a)))$ 
 $+ 6 * (\text{of-nat } (\text{ind } (a \leq \text{Nof } K \wedge$ 
 $a \leq K \bmod p)) + \text{of-nat } (\text{ind } (a \leq \text{Nof } K \wedge p \leq K \bmod p + a)))$ 
if a:  $a \in I$  for a
proof -
have a1:  $1 \leq a \leq m\text{Half } p$  using a by (simp-all add: I-def m-def)
have 2:  $2 * (\sum i < \text{outerCount } K p a. \text{Wn } K p a i) =$ 
 $- 7 * (\text{of-nat } (\text{ind } (a \leq K \bmod p)) + \text{of-nat } (\text{ind } (p \leq$ 

```

```

       $K \bmod p + a))) + 6 * \text{of-nat} (\text{ind} (a \leq \text{Nof } K)) * (\text{of-nat} (\text{ind} (a \leq$ 
       $K \bmod p)) + \text{of-nat} (\text{ind} (p \leq K \bmod p + a)))$ 
    using class-low[OF p H True a1] by blast
  also have  $6 * \text{of-nat} (\text{ind} (a \leq \text{Nof } K)) * (\text{of-nat} (\text{ind} (a \leq$ 
     $K \bmod p)) + \text{of-nat} (\text{ind} (p \leq K \bmod p + a))) =$ 
     $6 * (\text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge$ 
       $a \leq K \bmod p)) + \text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge p \leq K \bmod p + a)) :: \text{rat})$ 
    by (simp only: ind-mult[symmetric] of-nat-mult) (simp add: algebra-simps)
  finally show ?thesis .
qed
have  $2 * (\sum a \in I. \sum i < \text{outerCount } K \ p \ a. \text{Wn } K \ p \ a \ i) = (\sum a \in$ 
   $I. 2 * (\sum i < \text{outerCount } K \ p \ a. \text{Wn } K \ p \ a \ i))$ 
  by (simp add: sum-distrib-left)
also have  $\dots = (\sum a \in I. - 7 * (\text{of-nat} (\text{ind} (a \leq$ 
     $K \bmod p)) + \text{of-nat} (\text{ind} (p \leq K \bmod p + a)))$ 
     $+ 6 * (\text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge$ 
       $a \leq K \bmod p)) + \text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge p \leq K \bmod p + a))))$ 
    by (rule sum.cong[OF refl]) (rule cl)
  also have  $\dots = - 7 * (\text{of-nat} (\text{card } \{a \in I. a \leq K \bmod p\}) + \text{of-nat} (\text{card}$ 
     $\{a \in$ 
       $I. p \leq K \bmod p + a\}))$ 
       $+ 6 * (\text{of-nat} (\text{card } \{a \in I. a \leq \text{Nof } K \wedge$ 
         $a \leq K \bmod p\}) + \text{of-nat} (\text{card } \{a \in I. a \leq \text{Nof } K \wedge$ 
           $p \leq K \bmod p + a\}))$ 
    proof -
      have fI: finite I by (simp add: I-def)
      have  $(\sum a \in I. - 7 * (\text{of-nat} (\text{ind} (a \leq K \bmod p)) + \text{of-nat} (\text{ind} (p \leq$ 
         $K \bmod p + a)))$ 
         $+ 6 * (\text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge$ 
           $a \leq K \bmod p)) + \text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge p \leq K \bmod p + a)))) =$ 
         $- 7 * ((\sum a \in I. \text{of-nat} (\text{ind} (a \leq K \bmod p))) + (\sum a \in$ 
           $I. \text{of-nat} (\text{ind} (p \leq K \bmod p + a))))$ 
         $+ 6 * ((\sum a \in I. \text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge$ 
           $a \leq K \bmod p))) + (\sum a \in I. \text{of-nat} (\text{ind} (a \leq \text{Nof } K \wedge$ 
             $p \leq K \bmod p + a)))) :: \text{rat})$ 
        by (simp only: sum.distrib sum-distrib-left[symmetric])
      thus ?thesis by (simp only: ind-sum-rat[OF fI])
    qed
  finally show ?thesis .
qed
have Z: length (filter ( $\lambda x. \text{wOut } K \ p \ x = 0$ ) (outer-idx K p)) =  $2 * m - \text{Nof}$ 
 $K + u$ 
proof -
  have cl: card  $\{i. i < \text{outerCount } K \ p \ a \wedge$ 
     $\text{Wn } K \ p \ a \ i = 0\} = 2 - \text{ind} (a \leq \text{Nof } K) + \text{ind} ((a \leq \text{Nof } K \wedge$ 
       $a \leq K \bmod p) \wedge p \leq K \bmod p + a)$ 
    if a:  $a \in I$  for a
  proof -
    have a1:  $1 \leq a \leq mHalf \ p$  using a by (simp-all add: I-def m-def)

```

```

    show ?thesis using class-low[OF p H True a1] by (simp add: ind-mult)
  qed
  have fI: finite I by (simp add: I-def)
  have (∑ a ∈ I. card {i. i < outerCount K p a ∧ Wn K p a i = 0}) =
    (∑ a ∈ I. (2 - ind (a ≤ Nof K)) + ind ((a ≤ Nof K ∧
      a ≤ K mod p) ∧ p ≤ K mod p + a))
    by (rule sum.cong[OF refl]) (rule cl)
  also have ... = (∑ a ∈ I. 2 - ind (a ≤ Nof K)) + u
    by (simp only: sum.distrib ind-sum-nat[OF fI] c6)
  finally have s: (∑ a ∈ I. card {i. i < outerCount K p a ∧
    Wn K p a i = 0}) = (∑ a ∈ I. 2 - ind (a ≤ Nof K)) + u .
  have (∑ a ∈ I. 2 - ind (a ≤ Nof K)) + (∑ a ∈ I. ind (a ≤ Nof K)) =
    (∑ a ∈ I. (2::nat))
    by (simp only: sum.distrib[symmetric] two-minus-ind)
  moreover have (∑ a ∈ I. ind (a ≤ Nof K)) = Nof K using c5
    by (simp only: ind-sum-nat[OF fI])
  moreover have (∑ a ∈ I. (2::nat)) = 2 * m by (simp add: I-def)
  ultimately have (∑ a ∈ I. 2 - ind (a ≤ Nof K)) + Nof K = 2 * m by
simp
  hence (∑ a ∈ I. 2 - ind (a ≤ Nof K)) = 2 * m - Nof K by linarith
  thus ?thesis unfolding Z0 zc s by simp
qed
show ?thesis
  unfolding W0 Z z0 gammaOut-def Let-def
  unfolding u-def[symmetric] if-P[OF True]
proof (rule gamma-arith-low[OF S c12 v])
  show p ≤ K using outer-facts[OF p H] by blast
  show card {a ∈ I. a ≤ Nof K ∧ a ≤ K mod p} = min (Nof K) (K mod p)
    by (rule c3)
  show card {a ∈ I. a ≤ Nof K ∧ p ≤ K mod p + a} = u by (rule c4)
  show 2 * m + 1 = p using outer-facts[OF p H] by (simp add: m-def)
  show Nof K ≤ 2 * m using outer-facts[OF p H] by (simp add: m-def)
qed
next
case False
  have v: K mod p = K - 2 * p using outer-facts[OF p H] False by auto
  have oc0: outerCount K p 0 = 2 by (simp add: outerCount-zero[OF p H]
False)
  have z0: (∑ i < outerCount K p 0. Wn K p 0 i) = - 2 and zc: card {i. i <
    outerCount K p 0 ∧ Wn K p 0 i = 0} = 1
  proof -
    have s01: {..<outerCount K p 0} = {0, 1} using oc0 by auto
    show (∑ i < outerCount K p 0. Wn K p 0 i) = - 2 by (simp add: s01
Wn-def oc0)
    have {i. i < outerCount K p 0 ∧ Wn K p 0 i = 0} = {1} using oc0 by
(auto simp: Wn-def)
    thus card {i. i < outerCount K p 0 ∧ Wn K p 0 i = 0} = 1 by simp
  qed
  have S: 2 * (∑ a ∈ I. ∑ i < outerCount K p a. Wn K p a i) =

```

```

    - 14 * of-nat m - 7 * (of-nat (card {a ∈ I. a ≤ K mod p}) + of-nat (card
{a ∈
    I. p ≤ K mod p + a}))
    + 12 * of-nat (card {a ∈ I. a ≤ Nof K})
    + 5 * (of-nat (card {a ∈ I. a ≤ Nof K ∧
    a ≤ K mod p}) + of-nat (card {a ∈ I. a ≤ Nof K ∧ p ≤ K mod p + a}))
  proof -
    have cl: 2 * (∑ i < outerCount K p a. Wn K p a i) =
      - 14 - 7 * (of-nat (ind (a ≤ K mod p)) + of-nat (ind (p ≤ K mod p +
a)))
      + 12 * of-nat (ind (a ≤ Nof K))
      + 5 * (of-nat (ind (a ≤ Nof K ∧
      a ≤ K mod p)) + of-nat (ind (a ≤ Nof K ∧ p ≤ K mod p + a)))
      if a: a ∈ I for a
    proof -
      have a1: 1 ≤ a a ≤ mHalf p using a by (simp-all add: I-def m-def)
      have 2 * (∑ i < outerCount K p a. Wn K p a i) =
        - 14 - 7 * (of-nat (ind (a ≤ K mod p)) + of-nat (ind (p ≤
        K mod p + a))) + of-nat (ind (a ≤ Nof K)) * (12 + 5 * (of-nat (ind
(a ≤
        K mod p)) + of-nat (ind (p ≤ K mod p + a))))
        using class-high[OF p H False a1] by blast
      also have of-nat (ind (a ≤ Nof K)) * (12 + 5 * (of-nat (ind (a ≤
        K mod p)) + of-nat (ind (p ≤ K mod p + a)))) =
        12 * of-nat (ind (a ≤ Nof K)) + 5 * (of-nat (ind (a ≤ Nof K ∧
        a ≤ K mod p)) + of-nat (ind (a ≤ Nof K ∧ p ≤ K mod p + a)) :: rat)
      by (simp only: ind-mult[symmetric] of-nat-mult) (simp add: algebra-simps)
      finally show ?thesis by simp
    qed
  have cI: card I = m by (simp add: I-def)
  have 2 * (∑ a ∈ I. ∑ i < outerCount K p a. Wn K p a i) = (∑ a ∈
    I. 2 * (∑ i < outerCount K p a. Wn K p a i))
    by (simp add: sum-distrib-left)
  also have ... = (∑ a ∈ I. - 14 - 7 * (of-nat (ind (a ≤
    K mod p)) + of-nat (ind (p ≤ K mod p + a)))
    + 12 * of-nat (ind (a ≤ Nof K))
    + 5 * (of-nat (ind (a ≤ Nof K ∧
    a ≤ K mod p)) + of-nat (ind (a ≤ Nof K ∧ p ≤ K mod p + a))))
    by (rule sum.cong[OF refl]) (rule cl)
  also have ... = - 14 * of-nat m - 7 * (of-nat (card {a ∈ I. a ≤
    K mod p}) + of-nat (card {a ∈ I. p ≤ K mod p + a}))
    + 12 * of-nat (card {a ∈ I. a ≤ Nof K})
    + 5 * (of-nat (card {a ∈ I. a ≤ Nof K ∧
    a ≤ K mod p}) + of-nat (card {a ∈ I. a ≤ Nof K ∧
    p ≤ K mod p + a}))
    using cI by (simp add: sum.distrib sum-subtractf sum-distrib-left[symmetric]
ind-sum-rat
    I-def)
  finally show ?thesis .

```

```

qed
have Z: length (filter (λx. wOut K p x = 0) (outer-idx K p)) = 2 * m + 1 +
u
proof -
  have cl: card {i. i < outerCount K p a ∧
    Wn K p a i = 0} = 2 + ind ((a ≤ Nof K ∧ a ≤ K mod p) ∧
    p ≤ K mod p + a)
  if a: a ∈ I for a
  proof -
    have a1: 1 ≤ a a ≤ mHalf p using a by (simp-all add: I-def m-def)
    show ?thesis using class-high[OF p H False a1] by (simp add: ind-mult)
  qed
  have fI: finite I by (simp add: I-def)
  have (∑ a ∈ I. card {i. i < outerCount K p a ∧ Wn K p a i = 0}) =
    (∑ a ∈ I. 2 + ind ((a ≤ Nof K ∧ a ≤ K mod p) ∧
    p ≤ K mod p + a))
  by (rule sum.cong[OF refl]) (rule cl)
  also have ... = (∑ a ∈ I. (2::nat)) + u
  by (simp only: sum.distrib ind-sum-nat[OF fI] c6)
  also have ... = 2 * m + u by (simp add: I-def)
  finally show ?thesis unfolding Z0 zc by simp
qed
show ?thesis
  unfolding W0 Z z0 gammaOut-def Let-def
  unfolding u-def[symmetric] if-not-P[OF False]
proof (rule gamma-arith-high[OF S c12 v])
  show 2 * p ≤ K using False by simp
  show card {a ∈ I. a ≤ Nof K ∧ a ≤ K mod p} = min (Nof K) (K mod p)
  by (rule c3)
  show card {a ∈ I. a ≤ Nof K ∧ p ≤ K mod p + a} = u by (rule c4)
  show card {a ∈ I. a ≤ Nof K} = Nof K by (rule c5)
  show 2 * m + 1 = p using outer-facts[OF p H] by (simp add: m-def)
qed
qed
qed

```

22.9 Lemma 4.2

The weight arithmetic behind Lemma 4.2: for nonpositive half-integer weights and any U , $\sum_{i \in U} (w_i + 1/2) \leq \min(|U|, z)/2$, z the number of zero weights.

lemma *weight-loss-nat*:

```

fixes n :: nat and w :: nat ⇒ 'a :: linordered-field
assumes ∧i. i < n ⇒ w i ≤ 0 ∧i. i < n ⇒
  ∃ k::int. 2 * w i = of-int k U ⊆ {.. $n$ }
shows (∑ i ∈ U. w i + 1 / 2) ≤ of-nat (min (card U) (card {i. i < n ∧
  w i = 0})) / 2
proof -
  define Z where Z = {i. i < n ∧ w i = 0}

```

```

have U: finite U by (rule finite-subset[OF assms(3)]) simp
have Zf: finite Z by (simp add: Z-def)
have neg:  $w\ i + 1 / 2 \leq 0$  if  $i: i \in U - Z$  for  $i$ 
proof -
  have n:  $i < n$  and w0:  $w\ i \neq 0$  using  $i$  assms(3) by (auto simp: Z-def)
  obtain k :: int where  $k: 2 * w\ i = \text{of-int } k$  using assms(2)[OF n] by blast
  have  $2 * w\ i \leq 0$  using assms(1)[OF n] by simp
  hence  $\text{of-int } k \leq (0 :: 'a)$  by (simp only: k)
  hence  $k \leq 0$  by simp
  moreover have  $k \neq 0$  using k w0 by auto
  ultimately have  $k \leq -1$  by simp
  hence  $2 * w\ i \leq -1$  using k by (metis of-int-le-iff of-int-minus of-int-1)
  thus ?thesis by simp
qed
have split:  $(\sum i \in U. w\ i + 1 / 2) = (\sum i \in$ 
   $U \cap Z. w\ i + 1 / 2) + (\sum i \in U - Z. w\ i + 1 / 2)$ 
  by (rule sum.Int-Diff[OF U])
have p1:  $(\sum i \in U \cap Z. w\ i + 1 / 2) = \text{of-nat } (\text{card } (U \cap Z)) / 2$ 
  by (simp add: Z-def)
have p2:  $(\sum i \in U - Z. w\ i + 1 / 2) \leq 0$  by (rule sum-nonpos) (use neg in
blast)
have  $\text{card } (U \cap Z) \leq \min (\text{card } U) (\text{card } Z)$ 
  using U Zf by (simp add: card-mono)
hence p3:  $\text{of-nat } (\text{card } (U \cap Z)) / 2 \leq \text{of-nat } (\min (\text{card } U) (\text{card } Z)) / (2 ::$ 
   $'a)$ 
  by (intro divide-right-mono) simp-all
show ?thesis unfolding Z-def[symmetric] using split p1 p2 p3 by linarith
qed

```

lemma weight-loss:

```

fixes n :: nat and w :: nat  $\Rightarrow 'a :: \text{linordered-field}$ 
assumes  $\bigwedge i. i < n \Rightarrow w\ i \leq 0 \wedge i. i < n \Rightarrow$ 
   $\exists k :: \text{int}. 2 * w\ i = \text{of-int } k$   $U \subseteq \{..<n\}$ 
shows  $(\sum i \in U. w\ i + 1 / 2) \leq \text{of-nat } (\min (\text{card } U) (\text{card } \{i. i < n \wedge$ 
   $w\ i = 0\})) / 2$ 
using assms by (rule weight-loss-nat)

```

A matrix with a nontrivial linear relation among its rows has determinant zero (over an integral domain: multilinearity in the row carrying a nonzero coefficient).

lemma det-zero-of-row-relation:

```

fixes A :: 'a :: idom mat
assumes A:  $A \in \text{carrier-mat } n\ n$  and i0:  $i0 < n$  and c0:  $c\ i0 \neq 0$ 
  and rel:  $\bigwedge j. j < n \Rightarrow (\sum i = 0..<n. c\ i * A\ \$\$ (i, j)) = 0$ 
shows  $\text{Determinant.det } A = 0$ 
proof -
  have At:  $\text{transpose-mat } A \in \text{carrier-mat } n\ n$  using A by force
  define v where  $v = \text{vec } n\ c$ 
  have v:  $v \in \text{carrier-vec } n$  unfolding v-def by simp

```

```

have v0: v ≠ 0_v n
proof
  assume v = 0_v n
  hence v $ i0 = 0 using i0 by simp
  thus False using i0 c0 unfolding v-def by fastforce
qed
have Av: transpose-mat A *_v v = 0_v n
proof (rule eq-vecI)
  fix j assume j: j < dim-vec (0_v n :: 'a vec)
  have (∑ i = 0..

```

More than $\text{rank } L$ rows of a rational matrix L satisfy a nontrivial linear relation.

lemma rows-dependent:

```

assumes L: L ∈ carrier-mat n n and T: T ⊆ {0..

```



```

    have (F * L) $$ (t, j) = (∑ i = 0.. $n$ . (if i = xs ! t then 1 else 0) * L $$ (i,
j))
    using t' j' Fc L unfolding F-def by (auto simp: scalar-prod-def index-mult-mat)
    also have ... = (∑ i = 0.. $n$ . if i = xs ! t then L $$ (xs ! t, j) else 0)
    by (fastforce intro: sum.cong)
    also have ... = L $$ (xs ! t, j) using xn[OF t'] by (fastforce simp: sum.delta)
    finally show (F * L) $$ (t, j) = G $$ (t, j) using t' j' unfolding G-def by
auto
next
    show dim-row (F * L) = dim-row G using Fc L Gc by simp
    show dim-col (F * L) = dim-col G using Fc L Gc by force
qed
have rG: vec-space.rank k G ≤ vec-space.rank n L unfolding FL[symmetric]
    by (rule rank-mat-mul-left[OF Fc L])
define M where M = G * transpose-mat G
have GTc: transpose-mat G ∈ carrier-mat n k using Gc by simp
have Mc: M ∈ carrier-mat k k unfolding M-def using Gc GTc by force
have rM: vec-space.rank k M ≤ vec-space.rank k G unfolding M-def
    by (rule vec-space.rank-mat-mul-right[OF Gc GTc])
have rk: vec-space.rank k M < k using rM rG r unfolding k-def by linarith
have dM: Determinant.det M = 0 using vec-space.low-rank-det-zero[OF Mc] rk
by fastforce
obtain u where u: u ∈ carrier-vec k u ≠ 0_v k M *_v u = 0_v k
    using det-0-iff-vec-prod-zero[OF Mc] dM by blast
define w where w = transpose-mat G *_v u
have wc: w ∈ carrier-vec n unfolding w-def using GTc u(1) by simp
have Gw: G *_v w = 0_v k using u(3) Gc GTc u(1) unfolding M-def w-def by
force
have ww: w · w = 0
proof -
    have w · w = u · (G *_v w)
    using transpose-vec-mult-scalar[OF Gc wc u(1)] unfolding w-def by meson
    also have ... = 0 using Gw u(1) by fastforce
    finally show ?thesis .
qed
have w0: w $ j = 0 if j: j < n for j
proof -
    have s: w · w = (∑ i = 0.. $n$ . w $ i * w $ i) using wc unfolding
scalar-prod-def
    by simp
    have ∀ i ∈ {0.. $n$ }. w $ i * w $ i = 0
    using ww s sum-nonneg-eq-0-iff[of {0.. $n$ } λi. w $ i * w $ i] by auto
    thus ?thesis using j by force
qed
have wj: w $ j = (∑ t = 0.. $k$ . u $ t * L $$ (xs ! t, j)) if j: j < n for j
    using j Gc u(1) unfolding w-def G-def scalar-prod-def
    by (fastforce simp: scalar-prod-def mult commute)
have bij: bij-betw (λt. xs ! t) {0.. $k$ } T using dxs xsx lxs
    by (simp add: bij-betw-nth atLeast0LessThan)

```

```

define  $c$  where  $c\ i = (\text{if } i \in T \text{ then } u\ \$\ (\text{the-inv-into } \{0..<k\} \ (!)\ xs)\ i) \text{ else } 0)$ 
for  $i$ 
  have  $c0: \forall i. i \notin T \longrightarrow c\ i = 0$  unfolding  $c\text{-def}$  by presburger
  have  $cne: \exists i \in T. c\ i \neq 0$ 
  proof –
    obtain  $t$  where  $t: t < k \ u\ \$\ t \neq 0$  using  $u(1,2)$  by force
    have  $xs\ !\ t \in T$  using  $t(1)$   $sxs\ lxs$  by force
    moreover have  $\text{the-inv-into } \{0..<k\} \ (!)\ xs \ (xs\ !\ t) = t$  using  $t(1)$  bij
      by (force simp: the-inv-into-f-f bij-betw-imp-inj-on)
    ultimately show  $?thesis$  using  $t(2)$  unfolding  $c\text{-def}$  by metis
  qed
  have  $csum: (\sum i = 0..<n. c\ i * L\ \$\$ (i, j)) = (\sum t = 0..<k. u\ \$\ t * L\ \$\$ (xs\ !\ t, j))$ 
  for  $j$ 
  proof –
    have  $(\sum i = 0..<n. c\ i * L\ \$\$ (i, j)) = (\sum i \in T. c\ i * L\ \$\$ (i, j))$ 
      using  $T\ c0\ finT$  by (force intro: sum.mono-neutral-right)
    also have  $\dots = (\sum t = 0..<k. c\ (xs\ !\ t) * L\ \$\$ (xs\ !\ t, j))$ 
      using sum.reindex-bij-betw[OF bij, symmetric] by fast
    also have  $\dots = (\sum t = 0..<k. u\ \$\ t * L\ \$\$ (xs\ !\ t, j))$ 
      proof (rule sum.cong[OF refl])
        fix  $t$  assume  $t \in \{0..<k\}$ 
        hence  $t: t < k$  by simp
        have  $xs\ !\ t \in T$  using  $t\ xs\ lxs$  by force
        moreover have  $\text{the-inv-into } \{0..<k\} \ (!)\ xs \ (xs\ !\ t) = t$  using  $t$  bij
          by (force simp: the-inv-into-f-f bij-betw-imp-inj-on)
        ultimately show  $c\ (xs\ !\ t) * L\ \$\$ (xs\ !\ t, j) = u\ \$\ t * L\ \$\$ (xs\ !\ t, j)$ 
      qed
    unfolding  $c\text{-def}$ 
      by presburger
    qed
  finally show  $?thesis$  .
  qed
  show  $?thesis$  using  $c0\ cne\ csum\ wj\ w0$  by auto
qed

```

Multilinear expansion of the determinant of a sum over rows: for every subset T of the rows, take the rows in T from B and the others from A .

lemma *det-add-rows-expand*:

```

fixes  $A\ B :: 'a :: \text{comm-ring-1 mat}$ 
assumes  $A: A \in \text{carrier-mat } n\ n$  and  $B: B \in \text{carrier-mat } n\ n$ 
shows  $\text{Determinant.det } (A + B) =$ 
   $(\sum T \in \text{Pow } \{0..<n\}. \text{Determinant.det } (\text{Matrix.mat } n\ n\ (\lambda(i, j). \text{if } i \in T \text{ then } B\ \$\$ (i, j) \text{ else } A\ \$\$ (i, j))))$ 
proof –
  let  $?I = \{0..<n\}$ 
  let  $?M = \lambda T. \text{Matrix.mat } n\ n\ (\lambda(i, j). \text{if } i \in T \text{ then } B\ \$\$ (i, j) \text{ else } A\ \$\$ (i, j))$ 
  have  $AB: A + B \in \text{carrier-mat } n\ n$  using  $A\ B$  by force
  have  $M: ?M\ T \in \text{carrier-mat } n\ n$  for  $T$  by fastforce

```

```

have e0:  $(\prod i = 0..<n. (A + B) \$\$ (i, p i)) = (\prod i = 0..<n. B \$\$ (i, p i) + A \$\$ (i, p i))$ 
  if  $p$ :  $p$  permutes  $?I$  for  $p$ 
proof (rule prod.cong[OF refl])
  fix  $i$  assume  $i$ :  $i \in ?I$ 
  have  $pi$ :  $p i < n$  using permutes-in-image[OF  $p$ ]  $i$  by simp
  show  $(A + B) \$\$ (i, p i) = B \$\$ (i, p i) + A \$\$ (i, p i)$  using  $i$   $pi$   $A$   $B$  by
simp
qed
have e1:  $(\prod i = 0..<n. B \$\$ (i, p i) + A \$\$ (i, p i)) =$ 
   $(\sum T \in Pow ?I. (\prod i \in T. B \$\$ (i, p i)) * (\prod i \in ?I - T. A \$\$ (i, p i)))$ 
  for  $p$ 
  using prod-add[of  $?I$ ] by fast
have e2:  $(\prod i \in T. B \$\$ (i, p i)) * (\prod i \in ?I - T. A \$\$ (i, p i)) = (\prod i = 0..<n. ?M T \$\$ (i, p i))$ 
  if  $p$ :  $p$  permutes  $?I$  and  $T$ :  $T \in Pow ?I$  for  $p$   $T$ 
proof -
  have  $TI$ :  $T \subseteq ?I$  using  $T$  by blast
  have  $(\prod i = 0..<n. ?M T \$\$ (i, p i)) =$ 
     $(\prod i \in ?I - T. ?M T \$\$ (i, p i)) * (\prod i \in T. ?M T \$\$ (i, p i))$ 
    using prod.subset-diff[OF  $TI$ ] by blast
  moreover have  $(\prod i \in T. ?M T \$\$ (i, p i)) = (\prod i \in T. B \$\$ (i, p i))$ 
  proof (rule prod.cong[OF refl])
    fix  $i$  assume  $i$ :  $i \in T$ 
    have  $pi$ :  $p i < n$   $i < n$  using permutes-in-image[OF  $p$ ]  $i$   $TI$  by auto
    show  $?M T \$\$ (i, p i) = B \$\$ (i, p i)$  using  $pi$   $i$  by simp
  qed
  moreover have  $(\prod i \in ?I - T. ?M T \$\$ (i, p i)) = (\prod i \in ?I - T. A \$\$ (i, p$ 
 $i))$ 
  proof (rule prod.cong[OF refl])
    fix  $i$  assume  $i$ :  $i \in ?I - T$ 
    have  $pi$ :  $p i < n$   $i < n$  using permutes-in-image[OF  $p$ ]  $i$  by auto
    show  $?M T \$\$ (i, p i) = A \$\$ (i, p i)$  using  $pi$   $i$  by simp
  qed
  ultimately show  $?thesis$  using mult.commute by auto
qed
have Determinant.det  $(A + B) = (\sum p \in \{p. p \text{ permutes } ?I\}. \text{signof } p * (\prod i = 0..<n. (A + B) \$\$ (i, p i)))$ 
  using det-def'[OF  $AB$ ] by meson
also have  $\dots = (\sum p \in \{p. p \text{ permutes } ?I\}. \sum T \in Pow ?I. \text{signof } p * (\prod i = 0..<n. ?M T \$\$ (i, p i)))$ 
proof (rule sum.cong[OF refl])
  fix  $p$  assume  $p \in \{p. p \text{ permutes } ?I\}$ 
  hence  $p$ :  $p$  permutes  $?I$  by blast
  have  $\text{signof } p * (\prod i = 0..<n. (A + B) \$\$ (i, p i)) =$ 
     $\text{signof } p * (\sum T \in Pow ?I. (\prod i \in T. B \$\$ (i,$ 
 $p i)) * (\prod i \in ?I - T. A \$\$ (i, p i)))$ 
    using e0[OF  $p$ ]  $e1$  by presburger
  also have  $\dots = (\sum T \in Pow ?I. \text{signof } p * ((\prod i \in T. B \$\$ (i,$ 

```

```

    p i)) * (∏ i ∈ ?I - T. A $$ (i, p i)))
  using sum-distrib-left by blast
  also have ... = (∑ T ∈ Pow ?I. signof p * (∏ i = 0..<n. ?M T $$ (i, p i)))
  using e2[OF p] by auto
  finally show signof p * (∏ i = 0..<n. (A + B) $$ (i, p i)) =
    (∑ T ∈ Pow ?I. signof p * (∏ i = 0..<n. ?M T $$ (i, p i))) .
qed
also have ... = (∑ T ∈ Pow ?I. ∑ p ∈ {p. p permutes ?I}. signof p * (∏ i =
  0..<n. ?M T $$ (i, p i)))
  using sum.swap by fast
  also have ... = (∑ T ∈ Pow ?I. Determinant.det (?M T))
  using det-def'[OF M] by presburger
  finally show ?thesis .
qed

```

Lemma 4.2: if $v(A_{ij}) \geq w_i + w_j$ with nonpositive half-integer weights, L is integral of rank at most r , and exactly z weights are zero, then $v(\det(A + p \wedge 1L)) \geq 2 \sum w_i - \min(r, z)$ (Gauss valuation; L constant). Proof by the row-multilinear expansion (*Determinant.det-linear-rows-sum*), not by complementary minors.

theorem *det-vpGge-of-weighted-entries*:

```

  assumes p: prime p and A: A ∈ carrier-mat n n and L: L ∈ carrier-mat n n
  and hw: ∧ i. i < n ⇒
    w i ≤ 0 and hw2: ∧ i. i < n ⇒ ∃ k::int. 2 * w i = of-int k
  and hA: ∧ i j. i < n ⇒ j < n ⇒
    vpGge p (A $$ (i, j)) (w i + w j)
  and hL: ∧ i j. i < n ⇒ j < n ⇒
    L $$ (i, j) ≠ 0 ⇒ 0 ≤ vp p (L $$ (i, j))
  and hr: vec-space.rank n L ≤ r
  shows vpGge p (Determinant.det (A + map-mat (λc. [:c / of-nat p:] L))
    (2 * (∑ i < n. w i) - of-nat (min r (card {i. i < n ∧ w i = 0}))))

```

proof –

```

  define L' where L' = map-mat (λc. [:c / of-nat p:] L)
  define z where z = card {i. i < n ∧ w i = 0}
  define bnd where bnd = 2 * (∑ i < n. w i) - of-nat (min r z)
  have L'c: L' ∈ carrier-mat n n using L by (simp add: L'-def)
  have expand: Determinant.det (A + L') =
    (∑ T ∈ Pow {0..<n}. Determinant.det (Matrix.mat n n (λ(i, j).
      if i ∈ T then L' $$ (i, j) else A $$ (i, j))))
  by (rule det-add-rows-expand[OF A L'c])
  have trm: ∧ T. T ⊆ {0..<n} ⇒
    vpGge p (Determinant.det (Matrix.mat n n (λ(i, j).
      if i ∈ T then L' $$ (i, j) else A $$ (i, j)))) bnd

```

proof –

```

  fix T assume T: T ⊆ {0..<n}
  define P where P = Matrix.mat n n (λ(i, j).
    if i ∈ T then L' $$ (i, j) else A $$ (i, j))
  have Pc: P ∈ carrier-mat n n by (simp add: P-def)
  show vpGge p (Determinant.det P) bnd

```

```

proof (cases r < card T)
  case True
    — the  $p-1$  L rows in T are dependent, so this determinant vanishes
    obtain c :: nat  $\Rightarrow$  rat where c0:  $\bigwedge i. i \notin T \Rightarrow$ 
      c i = 0 and cne:  $\exists i \in T. c i \neq 0$ 
      and rel:  $\bigwedge j. j < n \Rightarrow (\sum i = 0..<n. c i * L \$\$ (i, j)) = 0$ 
      using rows-dependent[OF L T] True hr by auto
    have Determinant.det P = 0
  proof —
    obtain i0 where i0T: i0  $\in T$  and ci0: c i0  $\neq 0$  using cne by blast
    have i0: i0 < n using i0T T by auto
    have csum:  $(\sum i \in S. [ :f i : ]) = [ : \sum i \in S. f i : ]$ 
      for S and f :: nat  $\Rightarrow$  rat
      by (induction S rule: infinite-finite-induct) simp-all
    have rel':  $(\sum i = 0..<n. [ :c i : ] * P \$\$ (i, j)) = 0$  if j: j < n for j
    proof —
      have t:  $[ :c i : ] * P \$\$ (i, j) = [ :c i * L \$\$ (i, j) / \text{of-nat } p : ]$ 
        if i: i  $\in \{0..<n\}$  for i
        using i j L c0[of i] by (cases i  $\in T$ ) (simp-all add: P-def L'-def)
      have  $(\sum i = 0..<n. [ :c i : ] * P \$\$ (i, j)) = (\sum i =$ 
         $0..<n. [ :c i * L \$\$ (i, j) / \text{of-nat } p : ])$ 
        by (rule sum.cong[OF refl]) (rule t)
      also have ... =  $[ : (\sum i = 0..<n. c i * L \$\$ (i, j)) / \text{of-nat } p : ]$ 
        by (simp only: csum sum-divide-distrib)
      also have ... = 0 using rel[OF j] by simp
      finally show ?thesis .
    qed
    show ?thesis by (rule det-zero-of-row-relation[OF Pc i0, where c =
       $\lambda i. [ :c i : ]$ ) (use ci0 rel' in simp-all)
  qed
thus ?thesis by (simp add: vpGge-zero)
next
  case False
  hence Tr: card T  $\leq r$  by simp
  show ?thesis
  proof (rule vpGge-det-of-terms[OF p Pc])
    fix  $\sigma$  assume  $\sigma: \sigma$  permutes  $\{0..<n\}$ 
    define J where J =  $\{i \in \{0..<n\}. i \in T\}$ 
    have JT: J = T unfolding J-def using T by auto
    have fac:  $\bigwedge i. i < n \Rightarrow$ 
       $\text{vpGge } p (P \$\$ (i, \sigma i))$  (if i  $\in T$  then - 1 else w i + w ( $\sigma i$ ))
    proof —
      fix i assume i: i < n
      have si:  $\sigma i < n$  using permutes-in-image[OF  $\sigma$ ] i by force
      show  $\text{vpGge } p (P \$\$ (i, \sigma i))$  (if i  $\in T$  then - 1 else w i + w ( $\sigma i$ ))
      proof (cases i  $\in T$ )
        case True
          have vL:  $\text{vge } p (L \$\$ (i, \sigma i))$  0 using hL[OF i si] unfolding vge-def
          by presburger
        case False

```

```

have vge p (L $$ (i, σ i) * (1 / of-nat p)) (0 + - 1)
  by (rule vge-mul[OF p vL vge-inv-p[OF p]])
moreover have P $$ (i, σ i) = [:L $$ (i, σ i) / of-nat p:]
  using True i si L unfolding P-def L'-def by force
ultimately show ?thesis using True vpGge-C by force
next
case False
have e: P $$ (i, σ i) = A $$ (i, σ i) using False i si unfolding P-def
  by force
show ?thesis using hA[OF i si] e False by presburger
qed
qed
have vpGge p (∏ i = 0..

```

```

    by (rule weight-loss[OF hw hw2]) (use permutes-in-image[OF  $\sigma$ ] T in
auto)
  have l2:  $(\sum i \in J. w\ i + 1 / 2) \leq \text{of-nat} (\min (\text{card } T) z) / 2$ 
    unfolding z-def JT by (rule weight-loss[OF hw hw2 T[unfolded
atLeast0LessThan]])
  have min (card T) z  $\leq$  min r z using Tr by simp
  hence mm:  $(\text{of-nat} (\min (\text{card } T) z) :: \text{rat}) \leq \text{of-nat} (\min r z)$ 
    by (simp only: of-nat-le-iff)
  show ?thesis unfolding split hJT unfolding bnd-def atLeast0LessThan
using l1 l2 mm
    by linarith
  qed
  ultimately show vpGge p  $(\prod i = 0..<n. P\ \$\$ (i, \sigma\ i))$  bnd by (rule
vpGge-mono)
  qed
  qed
  qed
  show ?thesis
    unfolding L'-def[symmetric] expand bnd-def[symmetric] z-def[symmetric]
    by (rule vpGge-sum[OF p]) (use trm in auto)
  qed

```

22.10 Proposition 4.3

lemma *degree-outerBasis-lt:*

```

  assumes prime p  $\not\equiv 0 \pmod{4}$  dvd K OuterHyp K p  $x \in \text{set} (\text{outer-id}\ x\ K\ p)$ 
  shows degree (outerBasis K p x)  $<$  hof K
  proof -
    have p: prime p and H: OuterHyp K p using assms(1,3) .
    define a where a = fst x
    define i where i = snd x
    have am:  $a \leq mHalf\ p$  and ia:  $i < \text{outerCount } K\ p\ a$  using assms(4) mem-outer-id
    by (auto simp: a-def i-def)
    have dS: degree (sqPoleProd S)  $\leq$  card S if finite S for S
    proof -
      have degree (sqPoleProd S)  $\leq$   $(\sum j \in S. \text{degree } (pX + [: \text{of-nat } j \wedge 2:]))$ 
      unfolding sqPoleProd-def using degree-prod-sum-le[OF that] by (simp add:
o-def)
      also have ... = card S using degree-pX-plus-const by simp
      finally show ?thesis .
    qed
    have dP: degree  $((pX + [: \text{of-nat } n \wedge 2:]) \wedge k) \leq k$  for n k
    using degree-power-le[of pX + [: \text{of-nat } n \wedge 2:] k] degree-pX-plus-const by simp
    have dQ: degree (oq K p x)  $\leq$  i
    proof (cases a = 0)
      case True
      thus ?thesis using dP by (simp add: oq-def Let-def a-def i-def)
    next
      case False

```

```

hence a1: 1 ≤ a by simp
show ?thesis
proof (cases i < ell p K a - 2)
  case True
  thus ?thesis using dP False by (simp add: oq-def Let-def a-def i-def)
next
  case big: False
  have cE: card {j ∈ outerClassPoles K p a. p < j} + 2 = ell p K a
    by (rule card-class-high-add-two[OF p H a1 am])
  have fE: finite {j ∈ outerClassPoles K p a. p < j} by (simp add: outer-
ClassPoles-def)
  have degree (oq K p x) = degree (sqPoleProd {j ∈ outerClassPoles K p a. p
<
    j} * (pX + [:of-nat a ^ 2:]) ^ (i - (ell p K a - 2)))
    using False big by (simp add: oq-def Let-def a-def i-def)
  also have ... ≤ degree (sqPoleProd {j ∈ outerClassPoles K p a. p <
j}) + degree ((pX + [:of-nat a ^ 2:]) ^ (i - (ell p K a - 2)))
    by (rule degree-mult-le)
  also have ... ≤ card {j ∈ outerClassPoles K p a. p < j} + (i - (ell p K a
- 2))
    using dS[OF fE] dP[of a i - (ell p K a - 2)] by linarith
  also have ... = i using cE big by linarith
  finally show ?thesis .
qed
qed
obtain k where k: K = 40 * k using assms(2) by (elim dvdE)
have ct: card (tailPoles K) = hof K by (simp add: tailPoles-def k hof-def Nof-def)
have sub: outerClassPoles K p a ⊆ tailPoles K
  by (auto simp: outerClassPoles-def tailPoles-def)
have fC: finite (outerClassPoles K p a) by (simp add: outerClassPoles-def)
have fT: finite (tailPoles K - outerClassPoles K p a) by (simp add: tailPoles-def)
have cd: card (tailPoles K - outerClassPoles K p a) = hof K - card (outerClassPoles
K p a)
  using card-Diff-subset[OF fC sub] ct by simp
have cle: card (outerClassPoles K p a) ≤ hof K using card-mono[OF - sub] ct
  by (simp add: tailPoles-def)
have ic: i < card (outerClassPoles K p a) using ia by (simp add: outer-
Count-def)
have degree (outerBasis K p x) ≤ degree (sqPoleProd (tailPoles K - outer-
ClassPoles K p a)) +
  degree (oq K p x)
  unfolding outerBasis-eq a-def by (rule degree-mult-le)
also have ... ≤ (hof K - card (outerClassPoles K p a)) + i using dS[OF fT]
dQ cd
  by linarith
also have ... < hof K using ic cle by linarith
finally show ?thesis .
qed

```

Helpers for the unimodularity of the outer basis: congruences modulo p of

polynomials and of Taylor coefficients. $\text{oroot } p \ a$ is the expansion point of class a (p for the zero class).

definition $\text{oroot} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat}$ **where** $\text{oroot } p \ a = (\text{if } a = 0 \text{ then } p \text{ else } a)$

lemma oroot-mod : $\text{oroot } p \ a \bmod p = a \bmod p$
by ($\text{simp add: oroot-def}$)

lemma oroot-add-mod : $(k + \text{oroot } p \ a) \bmod p = (k + a) \bmod p$
by ($\text{simp add: oroot-def}$)

lemma vpGge-tay1 :

assumes p : $\text{prime } p$ **and** P : $\text{vpGge } p \ P \ 1$ **and** r : $\text{vge } p \ r \ 0$
shows $\text{vge } p \ (\text{tay } P \ r \ i) \ 1$

proof –

define P' **where** $P' = \text{Polynomial.smult } (1 / \text{of-nat } p) \ P$
have pp : $P = \text{Polynomial.smult } (\text{of-nat } p) \ P'$ **using** p **by** ($\text{simp add: } P'\text{-def prime-gt-0-nat}$)
have P' : $\text{vpGge } p \ P' \ 0$

proof –

have $\text{vge } p \ (1 / \text{of-nat } p * \text{Polynomial.coeff } P \ k) \ (-1 + 1)$ **for** k
using $\text{vge-mul}[\text{OF } p \ \text{vge-inv-p}[\text{OF } p], \text{ of } \text{Polynomial.coeff } P \ k \ 1] \ P$ **by** ($\text{simp add: vpGge-iff}$)
thus $?thesis$ **by** ($\text{simp add: vpGge-iff } P'\text{-def}$)

qed

have $\text{vge } p \ (\text{of-nat } p * \text{tay } P' \ r \ i) \ (1 + 0)$
by ($\text{rule vge-mul}[\text{OF } p \ \text{vge-p-self}[\text{OF } p] \ \text{vge-tay}[\text{OF } p \ P' \ r]]$)
thus $?thesis$ **using** $pp \ \text{tay-smult}$ **by** simp

qed

lemma tay-diff : $\text{tay } (P - Q) \ r \ i = \text{tay } P \ r \ i - \text{tay } Q \ r \ i$
using $\text{tay-add}[\text{of } P - Q \ Q \ r \ i]$ **by** simp

lemma vpGge-diff : $\text{prime } p \implies \text{vpGge } p \ A \ b \implies \text{vpGge } p \ B \ b \implies \text{vpGge } p \ (A - B) \ b$
using $\text{vpGge-add}[\text{of } p \ A \ b - B] \ \text{vpGge-neg}[\text{of } p \ B \ b]$ **by** simp

lemma vpGge-prod-diff :

assumes p : $\text{prime } p$ **and** S : $\text{finite } S$
and hf : $\bigwedge j. j \in S \implies \text{vpGge } p \ (f \ j) \ 0$ **and** hg : $\bigwedge j. j \in S \implies \text{vpGge } p \ (g \ j) \ 0$
and hfg : $\bigwedge j. j \in S \implies \text{vpGge } p \ (f \ j - g \ j) \ 1$
shows $\text{vpGge } p \ ((\prod j \in S. f \ j) - (\prod j \in S. g \ j)) \ 1$
using $S \ hf \ hg \ hfg$

proof ($\text{induction } S \text{ rule: finite-induct}$)

case empty

show $?case$ **using** vpGge-zero **by** simp

next

case ($\text{insert } x \ F$)

```

have IH: vpGge p (( $\prod j \in F. f j$ ) - ( $\prod j \in F. g j$ )) 1 using insert by blast
have fF: vpGge p ( $\prod j \in F. f j$ ) 0
  using vpGge-prod[OF p insert(1), of f  $\lambda\cdot. 0$ ] insert(4) by force
have gx: vpGge p (g x) 0 using insert(5) by blast
have fgx: vpGge p (f x - g x) 1 using insert(6) by blast
have e: ( $\prod j \in \text{insert } x F. f j$ ) - ( $\prod j \in \text{insert } x F. g j$ ) =
  (f x - g x) * ( $\prod j \in F. f j$ ) + g x * (( $\prod j \in F. f j$ ) - ( $\prod j \in$ 
    F. g j))
  using insert(1,2) by (simp add: algebra-simps)
have t1: vpGge p ((f x - g x) * ( $\prod j \in F. f j$ )) (1 + 0)
  by (rule vpGge-mul[OF p fgx fF])
have t2: vpGge p (g x * (( $\prod j \in F. f j$ ) - ( $\prod j \in F. g j$ ))) (0 + 1)
  by (rule vpGge-mul[OF p gx IH])
show ?case unfolding e using vpGge-add[OF p] t1 t2 by simp
qed

lemma sq-cong-mod:
  assumes prime p j mod p = a mod p  $\vee$  (j + a) mod p = 0
  shows vge p (of-nat j  $\wedge$  2 - of-nat a  $\wedge$  2) 1
proof -
  have p: prime p using assms(1) .
  have j: j mod p = a mod p  $\vee$  (j + a) mod p = 0 using assms(2) .
  have int p dvd (int j - int a) * (int j + int a)
  proof (cases j mod p = a mod p)
    case True
    hence int j mod int p = int a mod int p by (metis of-nat-mod)
    hence int p dvd int j - int a by (simp add: mod-eq-dvd-iff)
    thus ?thesis by simp
  next
    case False
    hence (j + a) mod p = 0 using j by simp
    hence p dvd j + a by (rule mod-0-imp-dvd)
    hence int p dvd int j + int a by (simp flip: of-nat-add)
    thus ?thesis by simp
  qed
  then obtain w where w: (int j - int a) * (int j + int a) = int p * w by (elim
dvdE)
  have e: (of-nat j  $\wedge$  2 - of-nat a  $\wedge$  2 :: rat) = of-nat p * of-int w
  proof -
    have (of-nat j  $\wedge$  2 - of-nat a  $\wedge$  2 :: rat) = of-int ((int j - int a) * (int j + int
a))
    by (simp add: power2-eq-square algebra-simps)
    thus ?thesis unfolding w by simp
  qed
  have vge p (of-nat p * of-int w) (1 + 0)
  by (rule vge-mul[OF p vge-p-self[OF p] vge-of-int[OF p]])
  thus ?thesis unfolding e by simp
qed

```

lemma *sq-unit-mod*:

assumes *p*: prime *p* **and** *nc*: $\neg (k \bmod p = s \bmod p \vee (k + s) \bmod p = 0)$

shows $(\text{of-nat } k^2 - \text{of-nat } s^2 :: \text{rat}) \neq 0 \wedge$

$\text{vp } p (\text{of-nat } k^2 - \text{of-nat } s^2) = 0$

proof –

have *pr*: prime (int *p*) **using** *p* **by** (simp add: prime-nat-int-transfer)

define *z* :: int **where** *z* = (int *k* – int *s*) * (int *k* + int *s*)

have *zq*: $(\text{of-nat } k^2 - \text{of-nat } s^2 :: \text{rat}) = \text{of-int } z$

by (simp add: z-def power2-eq-square algebra-simps)

have $\neg \text{int } p \text{ dvd } (\text{int } k - \text{int } s)$

proof

assume int *p* dvd (int *k* – int *s*)

hence int *k* mod int *p* = int *s* mod int *p* **using** mod-eq-dvd-iff **by** blast

hence int (*k* mod *p*) = int (*s* mod *p*) **using** zmod-int[of *k* *p*] zmod-int[of *s* *p*]

by simp

hence *k* mod *p* = *s* mod *p* **by** simp

thus False **using** *nc* **by** blast

qed

moreover have $\neg \text{int } p \text{ dvd } (\text{int } k + \text{int } s)$

proof

assume int *p* dvd (int *k* + int *s*)

hence *p* dvd *k* + *s* **using** of-nat-add[symmetric] **by** presburger

hence (*k* + *s*) mod *p* = 0 **by** simp

thus False **using** *nc* **by** meson

qed

ultimately have *nz*: $\neg \text{int } p \text{ dvd } z$ **unfolding** z-def **using** prime-dvd-mult-iff[OF

pr] **by** blast

hence *z* $\neq 0$ **by** auto

thus ?thesis **unfolding** *zq* **using** *nz* **by** (simp add: vp-def qmultiplicity-of-int not-dvd-imp-multiplicity-0)

qed

lemma *class-disjoint-m*:

assumes *m*: $2 * m\text{Half } p + 1 = p$ **and** *a*: $a \leq m\text{Half } p$ **and** *c*: $c \leq$

$m\text{Half } p$ **and** *ac*: $a \neq c$

shows $\text{outerClassPoles } K \text{ } p \text{ } a \cap \text{outerClassPoles } K \text{ } p \text{ } c = \{\}$

proof –

{ fix *j* **assume** *ja*: $j \in \text{outerClassPoles } K \text{ } p \text{ } a$ **and** *jc*: $j \in \text{outerClassPoles } K \text{ } p \text{ } c$

have $j \bmod p = a \vee (1 \leq a \wedge j \bmod p = p - a)$

by (rule res-class[OF *a* *m*]) (use *ja* in ⟨simp add: mem-class⟩)

moreover have $j \bmod p = c \vee (1 \leq c \wedge j \bmod p = p - c)$

by (rule res-class[OF *c* *m*]) (use *jc* in ⟨simp add: mem-class⟩)

ultimately have False **using** *a* *c* *ac* *m* **by** (elim disjE conjE; linarith) }

thus ?thesis **by** blast

qed

lemma *class-prod-cong*:

assumes *p*: prime *p* **and** *S*: $S \subseteq \text{outerClassPoles } K \text{ } p \text{ } c$

shows $\text{vpGge } p (\text{sqPoleProd } S - (pX - [:- (\text{of-nat } (\text{oroot } p \text{ } c) ^ 2):]) ^ \text{card } S)$

```

1
proof -
  have fS: finite S using S finite-subset by (fastforce simp: outerClassPoles-def)
  have e:  $(pX - [:- (of\text{-}nat\ (oroot\ p\ c) \wedge 2):]) \wedge card\ S = (\prod j \in$ 
     $S. pX - [:- (of\text{-}nat\ (oroot\ p\ c) \wedge 2):])$ 
    using prod-constant by simp
  have r:  $vpGge\ p\ (pX - [:- (of\text{-}nat\ (oroot\ p\ c) \wedge 2):])\ 0$ 
    using vpGge-diff[OF p vpGge-pX vpGge-C[OF vge-neg[OF vge-pow[OF p vge-of-nat[OF
p]]]]] by simp
  have d:  $vpGge\ p\ ((pX + [:- of\text{-}nat\ j \wedge 2:]) - (pX - [:- (of\text{-}nat\ (oroot\ p\ c) \wedge 2):]))$ 
1
    if  $j: j \in S$  for  $j$ 
  proof -
    have jm:  $j\ mod\ p = oroot\ p\ c\ mod\ p \vee (j + oroot\ p\ c)\ mod\ p = 0$ 
      using j S oroot-mod oroot-add-mod by (auto simp: outerClassPoles-def)
    have v:  $vge\ p\ (of\text{-}nat\ j \wedge 2 - of\text{-}nat\ (oroot\ p\ c) \wedge 2)\ 1$  by (rule sq-cong-mod[OF
p jm])
    have  $(pX + [:- of\text{-}nat\ j \wedge 2:]) - (pX - [:- (of\text{-}nat\ (oroot\ p\ c) \wedge 2):]) =$ 
       $[:- of\text{-}nat\ j \wedge 2 - of\text{-}nat\ (oroot\ p\ c) \wedge 2:]$ 
      by simp
    thus ?thesis using vpGge-C[OF v] by simp
  qed
  show ?thesis unfolding e sqPoleProd-def using vpGge-prod-diff[OF p fS vpGge-pX-plus-sq[OF
p] r d]
    by fast
  qed

lemma oq-cong:
  assumes  $p$ : prime p
  and  $cH$ :  $1 \leq fst\ x \implies$ 
     $card\ \{j \in outerClassPoles\ K\ p\ (fst\ x).\ p < j\} = ell\ p\ K\ (fst\ x) - 2$ 
  shows  $vpGge\ p\ (oq\ K\ p\ x - (pX - [:- (of\text{-}nat\ (oroot\ p\ (fst\ x)) \wedge 2):]) \wedge snd\ x)$ 
1
  proof -
    obtain  $a\ i$  where  $x: x = (a, i)$  by (cases x)
    have  $ePa: pX - [:- (of\text{-}nat\ (oroot\ p\ a) \wedge 2):] = pX + [:- of\text{-}nat\ (oroot\ p\ a) \wedge 2:]$ 
by simp
    show ?thesis
    proof ( $cases\ a = 0 \vee i < ell\ p\ K\ a - 2$ )
      case True
        have  $oq\ K\ p\ x = (pX + [:- of\text{-}nat\ (oroot\ p\ a) \wedge 2:]) \wedge i$ 
          using True unfolding x oq-def Let-def fst-conv snd-conv by (auto simp:
oroot-def)
        thus ?thesis unfolding x fst-conv snd-conv ePa using vpGge-zero by simp
      next
        case False
        hence  $a1: 1 \leq a$  and  $il: ell\ p\ K\ a - 2 \leq i$  by auto
        define  $H$  where  $H = \{j \in outerClassPoles\ K\ p\ a.\ p < j\}$ 
        have  $cH': card\ H = ell\ p\ K\ a - 2$  using  $cH\ a1$  unfolding H-def x fst-conv

```

```

by blast
  have oqe:  $oq\ K\ p\ x = sqPoleProd\ H * (pX + [:of-nat\ a\ \wedge\ 2:]) \wedge (i - (ell\ p\ K\ a - 2))$ 
  using False unfolding x oq-def Let-def H-def fst-conv snd-conv by simp
  have oa:  $oroot\ p\ a = a$  using a1 unfolding oroot-def by simp
  have ePa2:  $pX - [:-(of-nat\ a\ \wedge\ 2):] = (pX + [:of-nat\ a\ \wedge\ 2:]) :: rat\ poly$  by
simp
  have ii:  $card\ H + (i - (ell\ p\ K\ a - 2)) = i$  using cH' il by fastforce
  have pe:  $(pX - [:-(of-nat\ (oroot\ p\ (fst\ x))\ \wedge\ 2):]) \wedge snd\ x =$ 
 $(pX - [:-(of-nat\ (oroot\ p\ a)\ \wedge\ 2):]) \wedge card\ H * (pX + [:of-nat\ a\ \wedge\ 2:]) \wedge$ 
 $(i - (ell\ p\ K\ a - 2))$ 
  unfolding x fst-conv snd-conv oa ePa2
  using power-add[ $of\ pX + [:of-nat\ a\ \wedge\ 2:]$  card H i - (ell p K a - 2)] ii by
simp
  have HC:  $H \subseteq outerClassPoles\ K\ p\ a$  unfolding H-def by blast
  have d:  $vpGge\ p\ (sqPoleProd\ H - (pX - [:-(of-nat\ (oroot\ p\ a)\ \wedge\ 2):]) \wedge card\ H)\ 1$ 
  by (rule class-prod-cong[OF p HC])
  have vpGge p ( $(sqPoleProd\ H - (pX - [:-(of-nat\ (oroot\ p\ a)\ \wedge\ 2):]) \wedge card\ H) * (pX +$ 
 $[:of-nat\ a\ \wedge\ 2:]) \wedge (i - (ell\ p\ K\ a - 2)))\ (1 + 0)$ 
  by (rule vpGge-mul[OF p d vpGge-pow0[OF p vpGge-pX-plus-sq[OF p]]])
  thus ?thesis unfolding oqe pe by (simp add: algebra-simps)
qed
qed

lemma oq-cong-hyp:
  assumes p: prime p and H: OuterHyp K p and x:  $x \in set\ (outer-idx\ K\ p)$ 
  shows  $vpGge\ p\ (oq\ K\ p\ x - (pX - [:-(of-nat\ (oroot\ p\ (fst\ x))\ \wedge\ 2):]) \wedge snd\ x)\ 1$ 
  proof (rule oq-cong[OF p])
    assume a1:  $1 \leq fst\ x$ 
    have am:  $fst\ x \leq mHalf\ p$  using x mem-outer-idx by blast
    show  $card\ \{j \in outerClassPoles\ K\ p\ (fst\ x).\ p < j\} = ell\ p\ K\ (fst\ x) - 2$ 
    using card-class-high-add-two[OF p H a1 am] by simp
  qed

```

Below the diagonal of the Taylor matrix of the outer basis the entries are divisible by p .

```

lemma outer-tay-below:
  assumes p: prime p and m:  $2 * mHalf\ p + 1 = p$ 
  and oqc:  $vpGge\ p\ (oq\ K\ p\ (a, j) - (pX - [:-(of-nat\ (oroot\ p\ a)\ \wedge\ 2):]) \wedge j)\ 1$ 
  and a:  $a \leq mHalf\ p$  and c:  $c \leq mHalf\ p$  and ci:  $i < outerCount\ K\ p\ c$ 
  and lex:  $c < a \vee (c = a \wedge i < j)$ 
  shows  $vge\ p\ (tay\ (outerBasis\ K\ p\ (a, j))\ (- (of-nat\ (oroot\ p\ c)\ \wedge\ 2))\ i)\ 1$ 
  proof -
    let ?r =  $- (of-nat\ (oroot\ p\ c)\ \wedge\ 2) :: rat$ 
    have r0:  $vge\ p\ ?r\ 0$  by (rule vge-neg[OF p vge-pow[OF p vge-of-nat[OF p]]])
  
```

```

have  $E$ :  $\text{outerBasis } K \ p \ (a, j) = \text{sqPoleProd } (\text{tailPoles } K - \text{outerClassPoles } K \ p \ a) * \text{oq } K \ p \ (a, j)$ 
using  $\text{outerBasis-eq}$ [ $\text{of } K \ p \ (a, j)$ ] by  $\text{simp}$ 
have  $\exists E'. \text{vpGge } p \ (\text{outerBasis } K \ p \ (a, j) - E') \ 1 \wedge \text{tay } E' \ ?r \ i = 0$ 
proof ( $\text{cases } c < a$ )
  case  $\text{True}$ 
    have  $dj$ :  $\text{outerClassPoles } K \ p \ c \subseteq \text{tailPoles } K - \text{outerClassPoles } K \ p \ a$ 
    using  $\text{class-disjoint-m}$ [ $\text{OF } m \ a \ c$ ]  $\text{True}$  unfolding  $\text{tailPoles-def}$   $\text{outerClassPoles-def}$ 
by  $\text{fastforce}$ 
    define  $D$  where  $D = \text{tailPoles } K - \text{outerClassPoles } K \ p \ a - \text{outerClassPoles } K \ p \ c$ 
    define  $R$  where  $R = \text{sqPoleProd } D * \text{oq } K \ p \ (a, j)$ 
    have  $u$ :  $\text{tailPoles } K - \text{outerClassPoles } K \ p \ a = \text{outerClassPoles } K \ p \ c \cup D$ 
using  $dj$ 
    unfolding  $D\text{-def}$  by  $\text{blast}$ 
    have  $fC$ :  $\text{finite } (\text{outerClassPoles } K \ p \ c)$  and  $fD$ :  $\text{finite } D$ 
    unfolding  $D\text{-def}$   $\text{tailPoles-def}$   $\text{outerClassPoles-def}$  by  $\text{auto}$ 
    have  $dD$ :  $\text{outerClassPoles } K \ p \ c \cap D = \{\}$  unfolding  $D\text{-def}$  by  $\text{blast}$ 
    have  $sp$ :  $\text{sqPoleProd } (\text{tailPoles } K - \text{outerClassPoles } K \ p \ a) = \text{sqPoleProd } (\text{outerClassPoles } K \ p \ c) * \text{sqPoleProd } D$ 
    unfolding  $u$  using  $\text{sqPoleProd-union}$ [ $\text{OF } fC \ fD \ dD$ ] by  $\text{blast}$ 
    have  $d$ :  $\text{vpGge } p \ (\text{sqPoleProd } (\text{outerClassPoles } K \ p \ c) - (pX - [:?r:])) \wedge \text{outerCount } K \ p \ c) \ 1$ 
    using  $\text{class-prod-cong}$ [ $\text{OF } p \ \text{subset-refl}, \text{ of } K \ c$ ] unfolding  $\text{outerCount-def}$  by  $\text{blast}$ 
    have  $R0$ :  $\text{vpGge } p \ R \ 0$  unfolding  $R\text{-def}$ 
    by ( $\text{rule } \text{vpGge-mul0}$ [ $\text{OF } p \ \text{vpGge-sqPoleProd}$ [ $\text{OF } p$ ]  $\text{oq-vge}$ [ $\text{OF } p$ ]])
    have  $\text{vpGge } p \ ((\text{sqPoleProd } (\text{outerClassPoles } K \ p \ c) - (pX - [:?r:])) \wedge \text{outerCount } K \ p \ c) * R) \ (1 + 0)$ 
    by ( $\text{rule } \text{vpGge-mul}$ [ $\text{OF } p \ d \ R0$ ])
    moreover have  $\text{outerBasis } K \ p \ (a, j) - (pX - [:?r:])) \wedge \text{outerCount } K \ p \ c * R =$ 
     $(\text{sqPoleProd } (\text{outerClassPoles } K \ p \ c) - (pX - [:?r:])) \wedge \text{outerCount } K \ p \ c$ 
     $* R$ 
    unfolding  $E \ sp \ R\text{-def}$  by ( $\text{simp add: algebra-simps}$ )
    moreover have  $\text{tay } ((pX - [:?r:])) \wedge \text{outerCount } K \ p \ c * R) \ ?r \ i = 0$ 
    by ( $\text{rule } \text{tay-pow-mult-lt}$ [ $\text{OF } ci$ ])
    ultimately show  $?thesis$  by  $\text{force}$ 
next
  case  $\text{False}$ 
    hence  $ca$ :  $c = a$  and  $ij$ :  $i < j$  using  $\text{lex}$  by  $\text{auto}$ 
    define  $P$  where  $P = \text{sqPoleProd } (\text{tailPoles } K - \text{outerClassPoles } K \ p \ a)$ 
    have  $P0$ :  $\text{vpGge } p \ P \ 0$  unfolding  $P\text{-def}$  by ( $\text{rule } \text{vpGge-sqPoleProd}$ [ $\text{OF } p$ ])
    have  $\text{vpGge } p \ (P * (\text{oq } K \ p \ (a, j) - (pX - [:?r:])) \wedge j) \ (0 + 1)$ 
    using  $\text{vpGge-mul}$ [ $\text{OF } p \ P0 \ oqc$ ]  $ca$  by  $\text{blast}$ 
    moreover have  $\text{outerBasis } K \ p \ (a, j) - (pX - [:?r:])) \wedge j * P =$ 
     $P * (\text{oq } K \ p \ (a, j) - (pX - [:?r:])) \wedge j$ 
    unfolding  $E \ P\text{-def}$  by ( $\text{simp add: algebra-simps}$ )

```

```

moreover have tay ((pX - [:?r:]) ^ j * P) ?r i = 0 by (rule tay-pow-mult-lt[OF
ij])
ultimately show ?thesis by force
qed
then obtain E' where E1: vpGge p (outerBasis K p (a, j) - E') 1 and E2:
tay E' ?r i = 0
by blast
have vge p (tay (outerBasis K p (a, j) - E') ?r i) 1 by (rule vpGge-tay1[OF p
E1 r0])
thus ?thesis using E2 tay-diff by simp
qed

```

The diagonal entries of the Taylor matrix of the outer basis are p -adic units.

lemma outer-tay-diag:

```

assumes p: prime p
and oqc: vpGge p (oq K p (a, j) - (pX - [:-(of-nat (oroot p a) ^ 2):]) ^ j) 1
shows tay (outerBasis K p (a, j)) (- (of-nat (oroot p a) ^ 2)) j ≠ 0 ∧
vp p (tay (outerBasis K p (a, j)) (- (of-nat (oroot p a) ^ 2)) j) = 0
proof -
let ?r = - (of-nat (oroot p a) ^ 2) :: rat
define P where P = sqPoleProd (tailPoles K - outerClassPoles K p a)
have E: outerBasis K p (a, j) = P * oq K p (a, j) unfolding P-def
using outerBasis-eq[OF K p (a, j)] by simp
have r0: vge p ?r 0 by (rule vge-neg[OF vge-pow[OF p vge-of-nat[OF p]]])
have P0: vpGge p P 0 unfolding P-def by (rule vpGge-sqPoleProd[OF p])
have d1: vpGge p (P * (oq K p (a, j) - (pX - [:?r:]) ^ j)) (0 + 1)
by (rule vpGge-mul[OF p P0 oqc])
have d2: outerBasis K p (a, j) - (pX - [:?r:]) ^ j * P = P * (oq K p (a, j) -
(pX - [:?r:]) ^ j)
unfolding E by (simp add: algebra-simps)
define u where u = tay (outerBasis K p (a, j) - (pX - [:?r:]) ^ j * P) ?r j
have u1: vge p u 1 unfolding u-def d2 using vpGge-tay1[OF p - r0] d1 by
simp
have t0: tay ((pX - [:?r:]) ^ j * P) ?r j = poly P ?r
using tay-pow-mult[of ?r j P 0] tay-0-eq-poly[of P ?r] by simp
have fT: finite (tailPoles K - outerClassPoles K p a) unfolding tailPoles-def
by blast
have fac: ?r + of-nat k ^ 2 ≠ 0 ∧ vp p (?r + of-nat k ^ 2) = 0
if k: k ∈ tailPoles K - outerClassPoles K p a for k
proof -
have nc: ¬ (k mod p = oroot p a mod p ∨ (k + oroot p a) mod p = 0)
using k oroot-mod oroot-add-mod unfolding tailPoles-def outerClassPoles-def
by auto
have e: ?r + of-nat k ^ 2 = of-nat k ^ 2 - of-nat (oroot p a) ^ 2 by simp
show ?thesis unfolding e by (rule sq-unit-mod[OF p nc])
qed
have pP: poly P ?r = (∏ k ∈ tailPoles K - outerClassPoles K p a. ?r + of-nat
k ^ 2)
unfolding P-def sqPoleProd-def using poly-prod poly-monom by (simp add:

```

```

poly-prod poly-monom)
  have w0: poly P ?r ≠ 0 unfolding pP using fac fT by simp
  have w1: vp p (poly P ?r) = 0
  proof -
    have vp p (poly P ?r) = (∑ k ∈
      tailPoles K - outerClassPoles K p a. vp p (?r + of-nat k ^ 2))
      unfolding pP using vp-prod-rat[OF p fT] fac by presburger
    also have ... = 0 using fac by simp
    finally show ?thesis .
  qed
  have tsum: tay (outerBasis K p (a, j)) ?r j = poly P ?r + u unfolding u-def
  tay-diff t0 by simp
  show ?thesis
  proof (cases u = 0)
    case True
      thus ?thesis using tsum w0 w1 by simp
    next
      case False
        have lt: vp p (poly P ?r) < vp p u using u1 False w1 unfolding vge-def by
        simp
        have poly P ?r + u ≠ 0 ∧ vp p (poly P ?r + u) = vp p (poly P ?r)
          by (rule vp-add-eq-of-lt[OF p w0 False lt])
        thus ?thesis using tsum w1 by simp
      qed
    qed
  qed

```

(4.11): the outer basis is $\mathbb{Z}$ - p -unimodular, by the Taylor-matrix criterion. Row (a, i) is expanded at $-a^2$ (at $-p^2$ for the zero class); below the diagonal the entries are divisible by p (the poles $j > p$ of a class satisfy $j^2 \equiv a^2$), and the diagonal entries are units.

theorem *outerBasis-unimodular*:

```

assumes p: prime p and K: 40 dvd K and H: OuterHyp K p
defines B ≡ Matrix.mat (hof K) (hof K) (λ(x,
  d). Polynomial.coeff (outerBasis K p (outer-idx K p ! x)) d)
shows Determinant.det B ≠ 0 ∧ vp p (Determinant.det B) = 0
proof -
  define h where h = hof K
  define idx where idx = outer-idx K p
  define rt where rt = (λy. if fst (idx ! y) =
    0 then - (of-nat p ^ 2) else - (of-nat (fst (idx ! y)) ^ 2) :: rat)
  define od where od = (λy. snd (idx ! y))
  define V :: rat mat where V = Matrix.mat h h (λ(d, y). tay (pX ^ d) (rt y)
    (od y))
  define T :: rat mat where T = B * V
  have len: length idx = h by (simp add: idx-def h-def length-outer-idx[OF p K
    H])
  have Bc: B ∈ carrier-mat h h and Vc: V ∈ carrier-mat h h
    by (simp-all add: B-def V-def h-def)
  have Tentry: ∧x y. x < h ⇒ y < h ⇒

```



```

    T $$ (x, y) = tay (outerBasis K p (idx ! x)) (rt y) (od y)
  proof -
    fix x y assume x: x < h and y: y < h
    have mem: idx ! x ∈ set (outer-idx K p) using x len unfolding idx-def by
    fastforce
    have dg: degree (outerBasis K p (idx ! x)) < h using degree-outerBasis-lt[OF
    p K H mem]
    unfolding h-def by meson
    have dim-row B = h dim-col B = h dim-row V = h dim-col V = h using Bc
    Vc by auto
    with x y have T $$ (x, y) = (∑ d ∈ {0..

```

```

oqc am cm ci lex])
qed
have unit:  $\bigwedge x. x < h \implies T \text{ \$\$ } (x, x) \neq 0 \wedge vp \ p \ (T \text{ \$\$ } (x, x)) = 0$ 
proof -
  fix x assume x:  $x < h$ 
  obtain a j where ax:  $idx \ ! \ x = (a, j)$  by (cases idx ! x)
  have mx:  $idx \ ! \ x \in set \ (outer\text{-}idx \ K \ p)$  using x len unfolding idx-def by simp
  have oqc:  $vpGge \ p \ (oq \ K \ p \ (a, j) - (pX - [: - (of\text{-}nat \ (oroot \ p \ a) \wedge 2):]) \wedge j) \ 1$ 
    using oq-cong-hyp[OF p H mx] ax by simp
  have rtx:  $rt \ x = - \ (of\text{-}nat \ (oroot \ p \ a) \wedge 2)$  unfolding rt-def ax oroot-def by
simp
  have Te:  $T \text{ \$\$ } (x, x) = tay \ (outerBasis \ K \ p \ (a, j)) \ (- \ (of\text{-}nat \ (oroot \ p \ a) \wedge 2)) \ j$ 
    using Tentry[OF x x] ax rtx unfolding od-def by simp
  show  $T \text{ \$\$ } (x, x) \neq 0 \wedge vp \ p \ (T \text{ \$\$ } (x, x)) = 0$  unfolding Te
    by (rule outer-tay-diag[OF p oqc])
qed
have Bi:  $\bigwedge x \ d. x < h \implies d < h \implies vge \ p \ (B \text{ \$\$ } (x, d)) \ 0$ 
  using vpGge-outerBasis[OF p] by (auto simp: B-def h-def vpGge-iff)
have Vi:  $\bigwedge d \ y. d < h \implies y < h \implies vge \ p \ (V \text{ \$\$ } (d, y)) \ 0$ 
proof -
  fix d y assume d:  $d < h$  and y:  $y < h$ 
  have sq:  $vge \ p \ (- \ (of\text{-}nat \ m \wedge 2)) \ 0$  for m by (rule vge-neg[OF vge-pow[OF p
vge-of-nat[OF p]]])
  have rt0:  $vge \ p \ (rt \ y) \ 0$  by (simp add: rt-def sq)
  have vge_p (of-nat (d choose od y) * rt y  $\wedge$  (d - od y)) (0 + 0)
    by (rule vge-mul[OF p vge-of-nat[OF p] vge-pow[OF p rt0]])
  hence vf:  $vge \ p \ (of\text{-}nat \ (d \ choose \ od \ y) * rt \ y \wedge (d - od \ y)) \ 0$  by simp
  have V  $\text{ \$\$ } (d, y) = tay \ (pX \wedge d) \ (rt \ y) \ (od \ y)$  using d y by (simp add: V-def)
  thus  $vge \ p \ (V \text{ \$\$ } (d, y)) \ 0$  unfolding tay-pX-pow using vf by (simp add:
vge-zero)
qed
show ?thesis
  unfolding h-def[symmetric]
  by (rule unimodular-criterion[OF p Bc Vc Bi Vi]) (use below unit in <simp-all
add:
  T-def[symmetric]>)
qed

lemma muModPoly-eq-sum:
  assumes degree Q  $\leq N$ 
  shows muModPoly p Q =  $(\sum e \leq N. Polynomial.coeff \ Q \ e * (muMono \ e - cCorr \ p \ e / of\text{-}nat \ p))$ 
  unfolding muModPoly-def
proof (rule sum.mono-neutral-left)
  show finite  $\{..N\}$  by force
  show  $\{..degree \ Q\} \subseteq \{..N\}$  using assms by blast
  show  $\forall i \in \{..N\} - \{..degree \ Q\}. Polynomial.coeff \ Q \ i * (muMono \ i - cCorr \ p \ i /$ 
/

```

$of\text{-}nat\ p) = 0$ **by** (*simp add: coeff-eq-0*)
qed

lemma *muModPoly-add*: $muModPoly\ p\ (P + Q) = muModPoly\ p\ P + muModPoly\ p\ Q$

proof –

define N **where** $N = \max\ (degree\ P)\ (degree\ Q)$
have $degree\ (P + Q) \leq N$ $degree\ P \leq N$ $degree\ Q \leq N$
using *degree-add-le-max[of P Q]* **unfolding** *N-def* **by** *auto*
hence $muModPoly\ p\ (P + Q) = (\sum\ e \leq$
 $N. Polynomial.coeff\ (P + Q)\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))$
 $muModPoly\ p\ P = (\sum\ e \leq N. Polynomial.coeff\ P\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))$
 $muModPoly\ p\ Q = (\sum\ e \leq N. Polynomial.coeff\ Q\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))$
using *muModPoly-eq-sum* **by** *auto*
moreover **have** $(\sum\ e \leq N. Polynomial.coeff\ (P + Q)\ e * (muMono\ e - cCorr\ p\ e /$
 $of\text{-}nat\ p)) =$
 $(\sum\ e \leq N. Polynomial.coeff\ P\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p)) +$
 $(\sum\ e \leq$
 $N. Polynomial.coeff\ Q\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))$
by (*simp add: distrib-right sum.distrib*)
ultimately **show** *?thesis* **by** *argo*
qed

lemma *muModPoly-smult*: $muModPoly\ p\ (Polynomial.smult\ c\ P) = c * muModPoly\ p\ P$

proof –

have $muModPoly\ p\ (Polynomial.smult\ c\ P) = (\sum\ e \leq$
 $degree\ P. Polynomial.coeff\ (Polynomial.smult\ c\ P)\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))$
using *degree-smult-le[of c P]* *muModPoly-eq-sum* **by** *blast*
moreover **have** $muModPoly\ p\ P = (\sum\ e \leq$
 $degree\ P. Polynomial.coeff\ P\ e * (muMono\ e - cCorr\ p\ e / of\text{-}nat\ p))$
using *muModPoly-eq-sum* **by** *fast*
ultimately **show** *?thesis* **by** (*force simp: sum-distrib-left mult.assoc*)
qed

lemma *muXmod-add*: $muXmod\ p\ S\ (P + Q) = muXmod\ p\ S\ P + muXmod\ p\ S\ Q$

unfolding *muXmod-eq* **by** (*force simp: poly-div-add-left muModPoly-add sqResidue-add smult-add-left sum.distrib algebra-simps*)

lemma *muXmod-C-mul*: $muXmod\ p\ S\ ([:c:] * P) = [:c:] * muXmod\ p\ S\ P$

unfolding *muXmod-eq* **by** (*fastforce simp: div-smult-left muModPoly-smult sqResidue-smult smult-add-right smult-smult smult-sum-right*)

```

lemma muXmod-zero:  $\text{muXmod } p \ S \ 0 = 0$ 
  using muXmod-C-mul[of p S 0 0] by simp

lemma muXmod-sum:  $\text{muXmod } p \ S \ (\sum i \in I. f \ i) = (\sum i \in I. \text{muXmod } p \ S \ (f \ i))$ 
proof (induction I rule: infinite-finite-induct)
  case (infinite A)
    thus ?case using muXmod-zero by simp
next
  case empty
    thus ?case using muXmod-zero by auto
next
  case (insert x F)
    thus ?case using muXmod-add by force
qed

```

The Gram identity with the length and degree facts as hypotheses, so that it also applies for $p > K$.

```

lemma outerGram-eq-conj-gen:
  assumes lenH:  $\text{length } (\text{outer-idx } K \ p) = \text{hof } K$ 
    and dgH:  $\bigwedge x. x \in \text{set } (\text{outer-idx } K \ p) \implies$ 
       $\text{degree } (\text{outerBasis } K \ p \ x) < \text{hof } K$ 
  defines  $B \equiv \text{Matrix.mat } (\text{hof } K) \ (\text{hof } K) \ (\lambda(x,$ 
     $d). \text{Polynomial.coeff } (\text{outerBasis } K \ p \ (\text{outer-idx } K \ p \ ! \ x)) \ d)$ 
  shows  $\text{outerGram } K \ p = \text{map-mat } (\lambda q. [:q:]) \ B \ * \ A_{\text{out}} \ K \ p \ *$ 
     $\text{transpose-mat } (\text{map-mat } (\lambda q. [:q:]) \ B)$ 
proof –
  let ?h =  $\text{hof } K$  and ?idx =  $\text{outer-idx } K \ p$ 
  let ?Q =  $\text{map-mat } (\lambda q. [:q:]) \ B$ 
  have len:  $\text{length } ?\text{idx} = ?h$  by (rule lenH)
  have dB:  $\text{dim-row } B = ?h \ \text{dim-col } B = ?h$  unfolding B-def by auto
  have dG:  $\text{dim-row } (A_{\text{out}} \ K \ p) = ?h \ \text{dim-col } (A_{\text{out}} \ K \ p) = ?h$  unfolding Aout-def
by auto
  have sm:  $\text{muXmod } p \ S \ (\text{Polynomial.smult } a \ P) = \text{Polynomial.smult } a \ (\text{muXmod } p \ S \ P)$  for  $S \ a \ P$ 
    using muXmod-C-mul[of p S a P] by fastforce
  have qe:  $\text{outerBasis } K \ p \ (?\text{idx} \ ! \ x) = (\sum k < ?h. \text{Polynomial.smult } (B \ \$\$ \ (x,$ 
     $k)) \ (pX \ ^k))$ 
    if  $x: x < ?h$  for  $x$ 
  proof –
    have mem:  $?\text{idx} \ ! \ x \in \text{set } ?\text{idx}$  using  $x \ \text{len}$  by fastforce
    have dg:  $\text{degree } (\text{outerBasis } K \ p \ (?\text{idx} \ ! \ x)) \leq ?h - 1$ 
      using dgH[OF mem] by force
    have e:  $\{.. ?h - 1\} = \{.. < ?h\}$  using  $x$  by fastforce
    have  $\text{outerBasis } K \ p \ (?\text{idx} \ ! \ x) =$ 
       $(\sum k \leq ?h - 1. \text{Polynomial.monom } (\text{Polynomial.coeff } (\text{outerBasis } K \ p$ 
         $(?\text{idx} \ ! \ x)) \ k)$ 
       $k)$ 
    by (rule sym[OF poly-as-sum-of-monoms'[OF dg]])

```

also have $\dots = (\sum k < ?h. \text{Polynomial.monom} (\text{Polynomial.coeff} (\text{outerBasis } K p (?idx ! x)) k) k)$
unfolding $e \dots$
also have $\dots = (\sum k < ?h. \text{Polynomial.smult} (B \$\$ (x, k)) (pX \wedge k))$
proof (*rule sum.cong[OF refl]*)
fix k **assume** $k: k \in \{..<?h\}$
show $\text{Polynomial.monom} (\text{Polynomial.coeff} (\text{outerBasis } K p (?idx ! x)) k) k$
 $=$
 $\text{Polynomial.smult} (B \$\$ (x, k)) (pX \wedge k)$
unfolding $pX\text{-pow-monom smult-monom}$ **using** $x k$ **unfolding** $B\text{-def}$ **by**
force
qed
finally show $?thesis$.
qed
have $pq: Dm (Nof K) \wedge 6 * \text{outerBasis } K p (?idx ! x) * \text{outerBasis } K p (?idx ! y)$
 $=$
 $(\sum k < ?h. \sum l < ?h. \text{Polynomial.smult} (B \$\$ (x, k)) * B \$\$ (y, l)) (Dm (Nof K) \wedge 6 * pX \wedge (k + l))$
if $x: x < ?h$ **and** $y: y < ?h$ **for** $x y$
proof –
have $pq0: \text{outerBasis } K p (?idx ! x) * \text{outerBasis } K p (?idx ! y) = (\sum k < ?h. \sum l < ?h. \text{Polynomial.smult} (B \$\$ (x, k)) (pX \wedge k) * \text{Polynomial.smult} (B \$\$ (y, l)) (pX \wedge l))$
unfolding $qe[OF x] qe[OF y]$ **by** (*rule sum-product*)
have $Dm (Nof K) \wedge 6 * \text{outerBasis } K p (?idx ! x) * \text{outerBasis } K p (?idx ! y)$
 $=$
 $Dm (Nof K) \wedge 6 * (\text{outerBasis } K p (?idx ! x) * \text{outerBasis } K p (?idx ! y))$
by (*simp only: mult.assoc*)
also have $\dots = (\sum k < ?h. \sum l < ?h. \text{Polynomial.smult} (B \$\$ (x, k)) * B \$\$ (y, l)) (Dm (Nof K) \wedge 6 * pX \wedge (k + l))$
unfolding $pq0$ *sum-distrib-left*
proof (*intro sum.cong refl*)
fix $k l$
show $Dm (Nof K) \wedge 6 * (\text{Polynomial.smult} (B \$\$ (x, k)) (pX \wedge k) * \text{Polynomial.smult} (B \$\$ (y, l)) (pX \wedge l)) = \text{Polynomial.smult} (B \$\$ (x, k)) * B \$\$ (y, l) (Dm (Nof K) \wedge 6 * pX \wedge (k + l))$
by (*force simp: power-add mult-ac*)
qed
finally show $?thesis$.
qed
have *entry: outerGram* $K p \$\$ (x, y) = (?Q * Aout K p * \text{transpose-mat } ?Q) \$\$ (x, y)$
if $x: x < ?h$ **and** $y: y < ?h$ **for** $x y$

```

proof –
  have outerGram K p $$ (x, y) =
    muXmod p {1..K} (Dm (Nof K) ^ 6 * outerBasis K p (?idx ! x) * outerBasis
K p (?idx ! y))
  using x y len unfolding outerGram-def Let-def by force
  also have ... = ( $\sum k < ?h. \sum l <$ 
    ?h. [B $$ (x, k):] * Aout K p $$ (k, l) * [B $$ (y, l):])
  unfolding pq[OF x y] muXmod-sum sm
  proof (intro sum.cong refl)
  fix k l assume k ∈ {..?h} l ∈ {..?h}
  thus Polynomial.smult (B $$ (x, k) * B $$ (y,
    l)) (muXmod p {1..K} (Dm (Nof K) ^ 6 * pX ^ (k + l))) =
    [B $$ (x, k):] * Aout K p $$ (k, l) * [B $$ (y, l):] unfolding Aout-def
    by (force simp: mult-ac)
  qed
  also have ... = (?Q * Aout K p * transpose-mat ?Q) $$ (x, y)
  proof –
    have m: (?Q * Aout K p * transpose-mat ?Q) $$ (x, y) =
      ( $\sum l < ?h. (\sum k < ?h. [B \$$(x, k):] * Aout K p \$$(k, l)) * [B \$$(y,
l):])$ )
    using x y dB dG unfolding scalar-prod-def by (force simp: scalar-prod-def
atLeast0LessThan)
    show ?thesis unfolding m sum-distrib-right by (rule sum.swap)
    qed
    finally show ?thesis .
  qed
  show ?thesis
  proof (rule eq-matI)
    fix x y
    assume x: x < dim-row (?Q * Aout K p * transpose-mat ?Q) and y: y <
      dim-col (?Q * Aout K p * transpose-mat ?Q)
    have xh: x < ?h and yh: y < ?h using x y dB by auto
    show outerGram K p $$ (x, y) = (?Q * Aout K p * transpose-mat ?Q) $$ (x,
y)
    by (rule entry[OF xh yh])
  next
    show dim-row (outerGram K p) = dim-row (?Q * Aout K p * transpose-mat
    ?Q) using len dB dG
    unfolding outerGram-def Let-def by force
    show dim-col (outerGram K p) = dim-col (?Q * Aout K p * transpose-mat ?Q)
using len dB dG
    unfolding outerGram-def Let-def by force
  qed
qed

lemma outerGram-eq-conj:
  assumes p: prime p and K: 40 dvd K and H: OuterHyp K p
  defines B ≡ Matrix.mat (hof K) (hof K) (λ(x,
    d). Polynomial.coeff (outerBasis K p (outer-idr K p ! x)) d)

```

shows $outerGram\ K\ p = map\text{-}mat\ (\lambda q. [:q:])\ B * Aout\ K\ p * transpose\text{-}mat\ (map\text{-}mat\ (\lambda q. [:q:])\ B)$
unfolding $B\text{-}def$ **by** $(rule\ outerGram\text{-}eq\text{-}conj\text{-}gen[OF\ length\text{-}outer\text{-}idx[OF\ p\ K\ H])$
 $degree\text{-}outerBasis\text{-}lt[OF\ p\ K\ H])$

lemma $gammaOut\text{-}eq\text{-}sum\text{-}idx$:
assumes $prime\ p\ 40\ dvd\ K\ OuterHyp\ K\ p$
shows $of\text{-}int\ (gammaOut\ K\ p) =$
 $2 * (\sum\ x < length\ (outer\text{-}idx\ K\ p). wOut\ K\ p\ (outer\text{-}idx\ K\ p\ !\ x))$
 $- of\text{-}nat\ (min\ (rOut\ K\ p)\ (card\ \{x. x < length\ (outer\text{-}idx\ K\ p)\} \wedge$
 $wOut\ K\ p\ (outer\text{-}idx\ K\ p\ !\ x) = 0))$
using $gammaOut\text{-}eq\text{-}sum[OF\ assms]$ **by** $(simp\ add: sum\text{-}list\text{-}sum\text{-}nth\ length\text{-}filter\text{-}conv\text{-}card\ atLeast0LessThan)$

Proposition 4.3: under (4.9), $v\text{-}p\hat{G}(\Delta\text{-}K) \geq \gamma\text{-}p\hat{out}$. Assembly: change to the outer basis, split $G = A + p\hat{-}\hat{1}L$, bound the entries, apply Lemma 4.2, identify the bound.

lemma $vge\text{-}mat\text{-}mult0$:
assumes p : $prime\ p$ **and** A : $A \in carrier\text{-}mat\ n\ n$ **and** B : $B \in carrier\text{-}mat\ n\ n$
and hA : $\forall i < n. \forall j < n. vge\ p\ (A\ \$\$ (i,$
 $j))\ 0$ **and** hB : $\forall i < n. \forall j < n. vge\ p\ (B\ \$\$ (i, j))\ 0$
and i : $i < n$ **and** j : $j < n$
shows $vge\ p\ ((A * B)\ \$\$ (i, j))\ 0$
proof –
have e : $(A * B)\ \$\$ (i, j) = (\sum\ k = 0..<n. A\ \$\$ (i, k) * B\ \$\$ (k, j))$
using $A\ B\ i\ j$ **by** $(simp\ add: scalar\text{-}prod\text{-}def)$
show $?thesis$ **unfolding** e
proof $(rule\ vge\text{-}sum[OF\ p])$
fix k **assume** $k \in \{0..<n\}$
hence k : $k < n$ **by** $simp$
show $vge\ p\ (A\ \$\$ (i, k) * B\ \$\$ (k, j))\ 0$
using $vge\text{-}mul[OF\ p\ -, of\ A\ \$\$ (i, k)\ 0\ B\ \$\$ (k, j)\ 0]$ $hA\ hB\ i\ j\ k$ **by** $simp$
qed $simp$

qed

lemma $mat\text{-}mult\text{-}entry$:

assumes $A \in carrier\text{-}mat\ n\ m\ C \in carrier\text{-}mat\ m\ k\ i < n\ j < k$
shows $(A * C)\ \$\$ (i, j) = (\sum\ l = 0..<m. A\ \$\$ (i, l) * C\ \$\$ (l, j))$
using $assms$ **by** $(auto\ simp: scalar\text{-}prod\text{-}def)$

theorem $DeltaK\text{-}vpGge\text{-}outer$:

assumes p : $prime\ p$ **and** K : $40\ dvd\ K$ **and** H : $OuterHyp\ K\ p$
shows $vpGge\ p\ (DeltaK\ K)\ (of\text{-}int\ (gammaOut\ K\ p))$

proof –

define h **where** $h = hof\ K$
define B **where** $B = Matrix.mat\ (hof\ K)\ (hof\ K)\ (\lambda(x,$
 $d). Polynomial.coeff\ (outerBasis\ K\ p\ (outer\text{-}idx\ K\ p\ !\ x))\ d)$
define Bp **where** $Bp = map\text{-}mat\ (\lambda q. [:q:])\ B$
define w **where** $w = (\lambda x. wOut\ K\ p\ (outer\text{-}idx\ K\ p\ !\ x))$
have len : $length\ (outer\text{-}idx\ K\ p) = h$ **by** $(simp\ add: h\text{-}def\ length\text{-}outer\text{-}idx[OF\ p$

$K H]$)
have Bc : $B \in \text{carrier-mat } h \ h$ **and** Bpc : $Bp \in \text{carrier-mat } h \ h$ **and** Gc : $GK \ K \in \text{carrier-mat } h \ h$
and Ac : $Aout \ K \ p \in \text{carrier-mat } h \ h$ **and** OGc : $outerGram \ K \ p \in \text{carrier-mat } h \ h$
by (*simp-all add: B-def Bp-def GK-def Aout-def outerGram-def h-def len Let-def*)
have uni : $\text{Determinant.det } B \neq 0 \wedge \forall p \ p \ (\text{Determinant.det } B) = 0$
unfolding B -def **by** (*rule outerBasis-unimodular[OF p K H]*)
obtain L **where** Lc : $L \in \text{carrier-mat } h \ h$
and GL : $GK \ K = Aout \ K \ p + \text{map-mat } (\lambda c. [:c / \text{of-nat } p:]) \ L$
and $Lint$: $\bigwedge i \ j. i < h \implies j < h \implies L \ \$\$ (i, j) \neq 0 \implies 0 \leq \text{vp } p \ (L \ \$\$ (i, j))$
and $Lrank$: $\text{vec-space.rank } h \ L \leq rOut \ K \ p$
using GK -outer-decomposition[OF p K H] **unfolding** h -def **by** *blast*
define $L2$ **where** $L2 = B * L * \text{transpose-mat } B$
have $L2c$: $L2 \in \text{carrier-mat } h \ h$ **using** $Bc \ Lc$ **by** (*simp add: L2-def*)
have $conj$: $Bp * GK \ K * \text{transpose-mat } Bp = \text{outerGram } K \ p + \text{map-mat } (\lambda c. [:c / \text{of-nat } p:]) \ L2$
proof –
define f **where** $f = (\lambda c. [:c / \text{of-nat } p:] :: \text{rat poly})$
have sum-pC : $(\sum i \in S. [:g \ i:]) = [: \sum i \in S. g \ i:]$
for S **and** $g :: \text{nat} \Rightarrow \text{rat}$
by (*induction S rule: infinite-finite-induct*) *simp-all*
have MLc : $\text{map-mat } f \ L \in \text{carrier-mat } h \ h$ **using** Lc **by** *auto*
have BTc : $\text{transpose-mat } Bp \in \text{carrier-mat } h \ h$ **using** Bpc **by** *force*
have OGc : $\text{outerGram } K \ p = Bp * Aout \ K \ p * \text{transpose-mat } Bp$
using outerGram-eq-conj [OF p K H] **unfolding** B -def[*symmetric*] Bp -def[*symmetric*]
by *meson*
have $M1$: $Bp * \text{map-mat } f \ L * \text{transpose-mat } Bp = \text{map-mat } f \ L2$
proof (*rule eq-matI*)
fix $x \ y$ **assume** x : $x < \text{dim-row } (\text{map-mat } f \ L2)$ **and** y : $y < \text{dim-col } (\text{map-mat } f \ L2)$
have xh : $x < h$ **and** yh : $y < h$ **using** $x \ y \ L2c$ **by** *auto*
have $e1$: $(Bp * \text{map-mat } f \ L * \text{transpose-mat } Bp) \ \$\$ (x, y) =$
 $(\sum l = 0..<h. (\sum k = 0..<h. [:B \ \$\$ (x, k):] * [:L \ \$\$ (k, l) / \text{of-nat } p:] * [:B \ \$\$ (y, l):]))$
proof –
have BMc : $Bp * \text{map-mat } f \ L \in \text{carrier-mat } h \ h$ **using** $Bpc \ MLc$ **by** *fastforce*
have $(Bp * \text{map-mat } f \ L * \text{transpose-mat } Bp) \ \$\$ (x, y) = (\sum l = 0..<h. (Bp * \text{map-mat } f \ L) \ \$\$ (x, l) * \text{transpose-mat } Bp \ \$\$ (l, y))$
by (*rule mat-mult-entry*[OF $BMc \ BTc \ xh \ yh$])
also have $\dots = (\sum l = 0..<h. (\sum k = 0..<h. Bp \ \$\$ (x, k) * \text{map-mat } f \ L \ \$\$ (k, l)) * \text{transpose-mat } Bp \ \$\$ (l, y))$
proof (*rule sum.cong*[OF *refl*])
fix l **assume** $l \in \{0..<h\}$
hence l : $l < h$ **by** *simp*
show $(Bp * \text{map-mat } f \ L) \ \$\$ (x, l) * \text{transpose-mat } Bp \ \$\$ (l, y) = (\sum k$
 $=$


```

    0.. $h$ .  $Bp \text{ } \$\$ (x, k) * \text{map-mat } f \text{ } L \text{ } \$\$ (k, l)) * \text{transpose-mat } Bp \text{ } \$\$ (l, y)$ 
    using mat-mult-entry[OF Bpc MLc xh l] by argo
  qed
  also have ... =  $(\sum l = 0.. $h$ . (\sum k =$ 
    0.. $h$ .  $[:B \text{ } \$\$ (x, k):] * [:L \text{ } \$\$ (k, l) / \text{of-nat } p:] * [:B \text{ } \$\$ (y, l):])$ 
  proof (rule sum.cong[OF refl])
    fix  $l$  assume  $l \in \{0.. $h\}$ 
    hence  $l: l < h$  by simp
    have  $t: \text{transpose-mat } Bp \text{ } \$\$ (l, y) = [:B \text{ } \$\$ (y, l):]$  using  $l \text{ } yh \text{ } Bpc \text{ } Bc$ 
  unfolding Bp-def
    by auto
    have  $s: (\sum k = 0.. $h$ .  $Bp \text{ } \$\$ (x, k) * \text{map-mat } f \text{ } L \text{ } \$\$ (k, l)) = (\sum k =$ 
    0.. $h$ .  $[:B \text{ } \$\$ (x, k):] * [:L \text{ } \$\$ (k, l) / \text{of-nat } p:]$ )
    proof (rule sum.cong[OF refl])
      fix  $k$  assume  $k \in \{0.. $h\}$ 
      hence  $k: k < h$  by simp
      show  $Bp \text{ } \$\$ (x, k) * \text{map-mat } f \text{ } L \text{ } \$\$ (k, l) =$ 
         $[:B \text{ } \$\$ (x, k):] * [:L \text{ } \$\$ (k, l) / \text{of-nat } p:]$ 
      using  $k \text{ } l \text{ } xh \text{ } Bc \text{ } Lc$  unfolding Bp-def f-def by force
    qed
    show  $(\sum k = 0.. $h$ .  $Bp \text{ } \$\$ (x, k) * \text{map-mat } f \text{ } L \text{ } \$\$ (k, l)) * \text{transpose-mat}$ 
 $Bp \text{ } \$\$ (l,$ 
       $y) =$ 
       $(\sum k = 0.. $h$ .  $[:B \text{ } \$\$ (x, k):] * [:L \text{ } \$\$ (k, l) / \text{of-nat } p:] * [:B \text{ } \$\$ (y,$ 
 $l):])$ 
    unfolding  $s \text{ } t \text{ } ..$ 
  qed
  finally show ?thesis .
  qed
  have  $e2: \text{map-mat } f \text{ } L2 \text{ } \$\$ (x, y) = [:(\sum l = 0.. $h$ . (\sum k =$ 
    0.. $h$ .  $B \text{ } \$\$ (x, k) * L \text{ } \$\$ (k, l)) * B \text{ } \$\$ (y, l)) / \text{of-nat } p:]$ 
  proof -
    have  $BLc: B * L \in \text{carrier-mat } h \text{ } h$  using  $Bc \text{ } Lc$  by force
    have  $BTc': \text{transpose-mat } B \in \text{carrier-mat } h \text{ } h$  using  $Bc$  by force
    have  $L2 \text{ } \$\$ (x, y) = (\sum l = 0.. $h$ .  $(B * L) \text{ } \$\$ (x, l) * \text{transpose-mat } B \text{ } \$\$$ 
 $(l, y))$ 
    unfolding L2-def by (rule mat-mult-entry[OF BLc BTc' xh yh])
    also have ... =  $(\sum l = 0.. $h$ . (\sum k =$ 
    0.. $h$ .  $B \text{ } \$\$ (x, k) * L \text{ } \$\$ (k, l)) * B \text{ } \$\$ (y, l))$ 
    proof (rule sum.cong[OF refl])
      fix  $l$  assume  $l \in \{0.. $h\}$ 
      hence  $l: l < h$  by simp
      show  $(B * L) \text{ } \$\$ (x, l) * \text{transpose-mat } B \text{ } \$\$ (l, y) = (\sum k =$ 
    0.. $h$ .  $B \text{ } \$\$ (x, k) * L \text{ } \$\$ (k, l)) * B \text{ } \$\$ (y, l)$ 
      using mat-mult-entry[OF Bc Lc xh l]  $l \text{ } yh \text{ } Bc$  by force
    qed
    finally have  $L2 \text{ } \$\$ (x, y) = (\sum l = 0.. $h$ . (\sum k =$ 
    0.. $h$ .  $B \text{ } \$\$ (x, k) * L \text{ } \$\$ (k, l)) * B \text{ } \$\$ (y, l))$  .
    thus ?thesis using  $xh \text{ } yh \text{ } L2c$  unfolding f-def by fastforce$$$$$$$ 
```

```

qed
have e3: (∑ l = 0..

```

```

    show dim-row (Bp * map-mat f L * transpose-mat Bp) = dim-row (map-mat
f L2) using Bpc L2c
    by fastforce
    show dim-col (Bp * map-mat f L * transpose-mat Bp) = dim-col (map-mat f
L2) using BTc L2c
    by fastforce
  qed
  have Bp * GK K * transpose-mat Bp =
    Bp * Aout K p * transpose-mat Bp + Bp * map-mat f L * transpose-mat Bp
  proof -
    have d1: Bp * (Aout K p + map-mat f L) = Bp * Aout K p + Bp * map-mat
f L
    by (rule mult-add-distrib-mat[OF Bpc Ac MLc])
    have d2: (Bp * Aout K p + Bp * map-mat f L) * transpose-mat Bp =
      Bp * Aout K p * transpose-mat Bp + Bp * map-mat f L * transpose-mat Bp
    by (rule add-mult-distrib-mat[OF mult-carrier-mat[OF Bpc Ac] mult-carrier-mat[OF
Bpc MLc]
      BTc])
    show ?thesis unfolding GL f-def[symmetric] d1 d2 ..
  qed
  also have ... = outerGram K p + map-mat f L2 unfolding OGe[symmetric]
M1 ..
  finally show ?thesis unfolding f-def .
  qed
  have L2int:  $\bigwedge i j. i < h \implies j < h \implies$ 
    L2 $$ (i, j)  $\neq 0 \implies 0 \leq vp\ p\ (L2\ \$\$ (i, j))$ 
  proof -
    fix i j assume i:  $i < h$  and j:  $j < h$  and nz: L2 $$ (i, j)  $\neq 0$ 
    have vB:  $\forall i < h. \forall j < h. vge\ p\ (B\ \$\$ (i, j))\ 0$ 
    using vpGge-outerBasis[OF p] by (auto simp: B-def h-def vpGge-iff)
    have vL:  $\forall i < h. \forall j < h. vge\ p\ (L\ \$\$ (i, j))\ 0$  using Lint by (auto simp: vge-def)
    have vBT:  $\forall i < h. \forall j < h. vge\ p\ (transpose-mat\ B\ \$\$ (i, j))\ 0$  using vB Bc by
auto
    have BL:  $\forall i < h. \forall j < h. vge\ p\ ((B * L)\ \$\$ (i, j))\ 0$ 
    using vge-mat-mult0[OF p Bc Lc] vB vL by blast
    have vge p ((B * L * transpose-mat B) $$ (i, j)) 0
    by (rule vge-mat-mult0[OF p - - BL vBT i j]) (use Bc Lc in auto)
    thus  $0 \leq vp\ p\ (L2\ \$\$ (i, j))$  using nz by (simp add: L2-def vge-def)
  qed
  have L2rank:  $vec-space.rank\ h\ L2 \leq rOut\ K\ p$ 
  proof -
    have  $vec-space.rank\ h\ L2 \leq vec-space.rank\ h\ (B * L)$  unfolding L2-def
    by (rule vec-space.rank-mat-mul-right) (use Bc Lc in auto)
    also have  $\dots \leq vec-space.rank\ h\ L$  by (rule rank-mat-mul-left[OF Bc Lc])
    also note Lrank
    finally show ?thesis .
  qed
  have h42:  $vpGge\ p\ (Determinant.det\ (outerGram\ K\ p + map-mat\ (\lambda c. [c\ /\ of-nat\ p:] L2))$ 

```

```

      (2 * (∑ i < h. w i) - of-nat (min (rOut K p) (card {i. i < h ∧ w i
= 0}))))
    by (rule det-vpGge-of-weighted-entries[OF p OGc L2c])
      (use wOut-nonpos wOut-halfInt outerGram-entry-bound[OF p K H] L2int
L2rank in ⟨auto
      simp: w-def len⟩)
    have vpGge p (DeltaK K) (2 * (∑ i < h. w i) - of-nat (min (rOut K p) (card
{i. i < h ∧
      w i = 0}))))
      unfolding DeltaK-def
    by (rule det-gram-transfer[OF p Gc Bc conjunct1[OF uni] conjunct2[OF uni]])
(use h42 conj in
  ⟨simp add: Bp-def⟩)
  thus ?thesis using gammaOut-eq-sum-idx[OF p K H] by (simp add: w-def len)
qed

```

22.11 Proposition 4.3 for $p > K$

```

lemma muX-eq-muXmod-of-lt:
  assumes prime p degree (P div sqPoleProd S) < 2 * p - 3
  shows muX S P = muXmod p S P
proof -
  have (∑ e ≤ degree (P div sqPoleProd S). Polynomial.coeff (P div sqPoleProd
S) e *
    cCorr p e) = 0
  proof (rule sum.neutral, intro ballI)
    fix e assume e ∈ {..degree (P div sqPoleProd S)}
    hence e < 2 * p - 3 using assms(2) by simp
    thus Polynomial.coeff (P div sqPoleProd S) e * cCorr p e = 0 by (simp add:
cCorr-of-lt)
  qed
  thus ?thesis using muX-eq-muXmod-add[OF assms(1), of S P] by simp
qed

```

For $p > K \geq 40$: p is odd and ≥ 7 , class $a \geq 1$ poles are among a , $p - a$, the zero class is empty.

```

lemma large-facts:
  assumes prime p 40 dvd K 0 < K K < p
  shows 7 ≤ p ∧ odd p ∧ 2 * mHalf p + 1 = p ∧ outerCount K p 0 = 0 ∧
    (∀ a j. a ≤ mHalf p ⟶
      j ∈ outerClassPoles K p a ⟶ (j = a ∨ j = p - a) ∧
      Nof K + 1 ≤ j ∧ j ≤ K) ∧
    (∀ a. a ≤ mHalf p ⟶ ell p K a ≤ 2) ∧
    (∀ a. {j ∈ outerClassPoles K p a. p < j} = {})
proof -
  obtain k where k: K = 40 * k using assms(2) by (elim dvdE)
  have 40 ≤ K using assms(3) k by simp
  hence p7: 7 ≤ p using assms(4) by simp
  have o: odd p using prime-odd-nat[OF assms(1)] p7 by simp

```

then obtain t where $t: p = 2 * t + 1$ by $(elim\ oddE)$
have $m: 2 * mHalf\ p + 1 = p$ by $(simp\ add: mHalf-def\ t)$
have $jp: j < p \wedge Nof\ K + 1 \leq j \wedge j \leq K$ if $j \in outerClassPoles\ K\ p\ a$
for $a\ j$
using $that\ assms(4)$ by $(auto\ simp: outerClassPoles-def)$
have $outerClassPoles\ K\ p\ 0 = \{\}$
proof $(rule\ equals0I)$
fix j assume $j: j \in outerClassPoles\ K\ p\ 0$
hence $j\ mod\ p = 0$ by $(auto\ simp: outerClassPoles-def)$
moreover have $j < p\ 1 \leq j$ using $jp[OF\ j]$ by $auto$
ultimately show $False$ by $simp$
qed
hence $c0: outerCount\ K\ p\ 0 = 0$ by $(simp\ add: outerCount-def)$
have $cl: \forall a\ j. a \leq mHalf\ p \longrightarrow$
 $j \in outerClassPoles\ K\ p\ a \longrightarrow (j = a \vee j = p - a) \wedge$
 $Nof\ K + 1 \leq j \wedge j \leq K$
proof $(intro\ allI\ impI)$
fix $a\ j$ assume $a: a \leq mHalf\ p$ and $j: j \in outerClassPoles\ K\ p\ a$
have $j\ mod\ p = a \vee (1 \leq a \wedge j\ mod\ p = p - a)$
by $(rule\ res-class[OF\ a\ m])$ $(use\ j\ in\ \langle simp\ add: outerClassPoles-def \rangle)$
thus $(j = a \vee j = p - a) \wedge Nof\ K + 1 \leq j \wedge j \leq K$ using $jp[OF\ j]$ by $auto$
qed
have $el: \forall a. a \leq mHalf\ p \longrightarrow ell\ p\ K\ a \leq 2$
proof $(intro\ allI\ impI)$
fix a assume $a: a \leq mHalf\ p$
have $\{j \in \{1..K\}. j\ mod\ p = a\ mod\ p \vee (j + a)\ mod\ p = 0\} \subseteq \{a, p - a\}$
proof
fix j assume $j: j \in \{j \in \{1..K\}. j\ mod\ p = a\ mod\ p \vee (j + a)\ mod\ p = 0\}$
have $j\ mod\ p = a \vee (1 \leq a \wedge j\ mod\ p = p - a)$
by $(rule\ res-class[OF\ a\ m])$ $(use\ j\ in\ simp)$
moreover have $j < p$ using $j\ assms(4)$ by $auto$
ultimately show $j \in \{a, p - a\}$ by $auto$
qed
hence $ell\ p\ K\ a \leq card\ \{a, p - a\}$ unfolding $ell-def$ by $(rule\ card-mono[rotated])$
 $simp$
also have $card\ \{a, p - a\} \leq 2$ by $(simp\ add: card-insert-if)$
finally show $ell\ p\ K\ a \leq 2$.
qed
have $e: \forall a. \{j \in outerClassPoles\ K\ p\ a. p < j\} = \{\}$ using jp by $fastforce$
show $?thesis$ using $p7\ o\ m\ c0\ cl\ el\ e$ by $blast$
qed

For $p > K$: the length of the outer index list, the degree bound and the congruence of the class factors.

lemma $length-outer-idx-large$:

assumes p : prime p and K : $40\ dvd\ K$ and $K0$: $0 < K$ and Kp : $K < p$

shows $length\ (outer-idx\ K\ p) = hof\ K$

proof -

obtain k where k : $K = 40 * k$ using K by $(elim\ dvdE)$

```

have p2: 2 < p using K0 Kp k by simp
have hk: hof K = 37 * k and Nk: Nof K = 3 * k unfolding k hof-def Nof-def
by simp-all
have length (outer-idx K p) = (∑ a ∈ set [0..<mHalf p + 1]. outerCount K p
a)
  unfolding outer-idx-def length-concat map-map
  by (simp add: o-def sum-list-distinct-conv-sum-set)
also have set [0..<mHalf p + 1] = {..mHalf p} by auto
also have (∑ a ≤ mHalf p. outerCount K p a) = card {Nof K + 1..K}
  unfolding outerCount-def outerClassPoles-def by (rule sum-classes-card[OF p
p2]) simp
also have ... = hof K using hk Nk k by simp
finally show ?thesis .
qed

```

lemma *degree-outerBasis-lt-large:*

```

assumes p: prime p and K: 40 dvd K and K0: 0 < K and Kp: K < p and x:
x ∈
  set (outer-idx K p)
shows degree (outerBasis K p x) < hof K
proof -
  define a where a = fst x
  define i where i = snd x
  have ia: i < outerCount K p a using x mem-outer-idx by (auto simp: a-def
i-def)
  have dS: degree (sqPoleProd S) ≤ card S if finite S for S
  proof -
    have degree (sqPoleProd S) ≤ (∑ j ∈ S. degree (pX + [:of-nat j ^ 2:]))
    unfolding sqPoleProd-def using degree-prod-sum-le[OF that] by (simp add:
o-def)
    also have ... = card S using degree-pX-plus-const by simp
    finally show ?thesis .
  qed
  have dP: degree ((pX + [:of-nat n ^ 2:]) ^ k) ≤ k for n k
  using degree-power-le[of pX + [:of-nat n ^ 2:] k] degree-pX-plus-const by simp
  have dQ: degree (oq K p x) ≤ i
  proof (cases a = 0)
    case True
    thus ?thesis using dP by (simp add: oq-def Let-def a-def i-def)
  next
    case False
    show ?thesis
    proof (cases i < ell p K a - 2)
      case True
      thus ?thesis using dP False by (simp add: oq-def Let-def a-def i-def)
    next
      case big: False
      have cE: {j ∈ outerClassPoles K p a. p < j} = {} using large-facts[OF p K
K0 Kp]

```

```

    by blast
    have fE: finite {j ∈ outerClassPoles K p a. p < j} by (simp add: outer-
ClassPoles-def)
    have degree (oq K p x) = degree (sqPoleProd {j ∈ outerClassPoles K p a. p
<
      j} * (pX + [:of-nat a ^ 2:]) ^ (i - (ell p K a - 2)))
    using False big by (simp add: oq-def Let-def a-def i-def)
    also have ... ≤ degree (sqPoleProd {j ∈ outerClassPoles K p a. p <
      j}) + degree ((pX + [:of-nat a ^ 2:]) ^ (i - (ell p K a - 2)))
    by (rule degree-mult-le)
    also have ... ≤ card {j ∈ outerClassPoles K p a. p < j} + (i - (ell p K a
- 2))
    using dS[OF fE] dP[of a i - (ell p K a - 2)] by linarith
    also have ... ≤ i unfolding cE by simp
    finally show ?thesis .
  qed
qed
obtain k where k: K = 40 * k using K by (elim dvdE)
have ct: card (tailPoles K) = hof K by (simp add: tailPoles-def k hof-def Nof-def)
have sub: outerClassPoles K p a ⊆ tailPoles K
  by (auto simp: outerClassPoles-def tailPoles-def)
have fC: finite (outerClassPoles K p a) by (simp add: outerClassPoles-def)
have fT: finite (tailPoles K - outerClassPoles K p a) by (simp add: tailPoles-def)
have cd: card (tailPoles K - outerClassPoles K p a) = hof K - card (outerClassPoles
K p a)
  using card-Diff-subset[OF fC sub] ct by simp
have cle: card (outerClassPoles K p a) ≤ hof K using card-mono[OF - sub] ct
  by (simp add: tailPoles-def)
have ic: i < card (outerClassPoles K p a) using ia by (simp add: outer-
Count-def)
have degree (outerBasis K p x) ≤ degree (sqPoleProd (tailPoles K - outer-
ClassPoles K p a)) +
  degree (oq K p x)
  unfolding outerBasis-eq a-def by (rule degree-mult-le)
also have ... ≤ (hof K - card (outerClassPoles K p a)) + i using dS[OF fT]
dQ cd
  by linarith
also have ... < hof K using ic cle by linarith
finally show ?thesis .
qed

lemma oq-cong-large:
  assumes p: prime p and K: 40 dvd K and K0: 0 < K and Kp: K < p and x:
x ∈
    set (outer-id x K p)
  shows vpGge p (oq K p x - (pX - [: - (of-nat (oroot p (fst x)) ^ 2):]) ^ snd x)
1
proof (rule oq-cong[OF p])
  assume 1 ≤ fst x

```

```

have am: fst x ≤ mHalf p using x mem-outer-idx by blast
have e: {j ∈ outerClassPoles K p (fst x). p < j} = {} and l2: ell p K (fst x) ≤
2
  using large-facts[OF p K K0 Kp] am by auto
show card {j ∈ outerClassPoles K p (fst x). p < j} = ell p K (fst x) - 2 using
l2
  unfolding e by simp
qed

lemma outerBasis-unimodular-large:
  assumes p: prime p and K: 40 dvd K and K0: 0 < K and Kp: K < p
  defines B ≡ Matrix.mat (hof K) (hof K) (λ(x,
    d). Polynomial.coeff (outerBasis K p (outer-idx K p ! x)) d)
  shows Determinant.det B ≠ 0 ∧ ∀p p (Determinant.det B) = 0
proof -
  define h where h = hof K
  define idx where idx = outer-idx K p
  define rt where rt = (λy. if fst (idx ! y) =
    0 then - (of-nat p ^ 2) else - (of-nat (fst (idx ! y)) ^ 2) :: rat)
  define od where od = (λy. snd (idx ! y))
  define V :: rat mat where V = Matrix.mat h h (λ(d, y). tay (pX ^ d) (rt y)
(od y))
  define T :: rat mat where T = B * V
  have len: length idx = h unfolding idx-def h-def by (rule length-outer-idx-large[OF
p K K0 Kp])
  have Bc: B ∈ carrier-mat h h and Vc: V ∈ carrier-mat h h
    by (simp-all add: B-def V-def h-def)
  have Tentry: ∧x y. x < h ⇒ y < h ⇒
    T $$ (x, y) = tay (outerBasis K p (idx ! x)) (rt y) (od y)
  proof -
    fix x y assume x: x < h and y: y < h
    have mem: idx ! x ∈ set (outer-idx K p) using x len unfolding idx-def by
fastforce
    have dg: degree (outerBasis K p (idx ! x)) < h
      using degree-outerBasis-lt-large[OF p K K0 Kp mem] unfolding h-def by
meson
    have dim-row B = h dim-col B = h dim-row V = h dim-col V = h using Bc
Vc by auto
    with x y have T $$ (x, y) = (∑ d ∈ {0..<h}. B $$ (x, d) * V $$ (d, y))
    unfolding T-def scalar-prod-def by (auto simp: scalar-prod-def index-mult-mat)
    also have ... = (∑ d < h. Polynomial.coeff (outerBasis K p (idx ! x)) d *
      tay (pX ^ d) (rt y) (od y))
    proof (rule sum.cong)
      show {0..<h} = {..<h} by fastforce
      fix d assume d: d ∈ {..<h}
      show B $$ (x, d) * V $$ (d, y) = Polynomial.coeff (outerBasis K p (idx !
x)) d *
        tay (pX ^ d) (rt y) (od y)
      using x y d unfolding B-def V-def idx-def h-def by force
    end
  end

```



```

qed
also have ... = tay (outerBasis K p (idx ! x)) (rt y) (od y)
  by (rule tay-eq-sum[OF dg, symmetric])
finally show T $$ (x, y) = tay (outerBasis K p (idx ! x)) (rt y) (od y) .
qed
have below:  $\bigwedge x y. y < x \implies x < h \implies vge\ p\ (T\ \$\$ (x, y))\ 1$ 
proof -
  fix x y assume yx:  $y < x$  and x:  $x < h$ 
  have y:  $y < h$  using yx x by simp
  obtain a j where ax:  $idx\ !\ x = (a, j)$  by (cases idx ! x)
  obtain c i where cy:  $idx\ !\ y = (c, i)$  by (cases idx ! y)
  have mx:  $idx\ !\ x \in set\ (outer\ idx\ K\ p)$  and my:  $idx\ !\ y \in set\ (outer\ idx\ K\ p)$ 
    using x y len unfolding idx-def by auto
  have am:  $a \leq mHalf\ p$  using mx ax mem-outer-idx by simp
  have cm:  $c \leq mHalf\ p$  and ci:  $i < outerCount\ K\ p\ c$  using my cy mem-outer-idx
  by auto
  have lex:  $c < a \vee (c = a \wedge i < j)$  using outer-idx-lex[of y x K p] yx x len ax
  cy
    unfolding idx-def by simp
  have m:  $2 * mHalf\ p + 1 = p$  using large-facts[OF p K K0 Kp] by blast
  have oqc:  $vpGge\ p\ (oq\ K\ p\ (a, j) - (pX - [:-(of\ nat\ (oroot\ p\ a) \wedge 2):]) \wedge j)\ 1$ 
    using oq-cong-large[OF p K K0 Kp mx] ax by simp
  have rty:  $rt\ y = - (of\ nat\ (oroot\ p\ c) \wedge 2)$  unfolding rt-def cy oroot-def by
  simp
  have Te:  $T\ \$\$ (x, y) = tay\ (outerBasis\ K\ p\ (a, j))\ (- (of\ nat\ (oroot\ p\ c) \wedge 2))\ i$ 
    using Tentry[OF x y] ax rty cy unfolding od-def by simp
  show vge p (T $$ (x, y)) 1 unfolding Te by (rule outer-tay-below[OF p m
  oqc am cm ci lex])
qed
have unit:  $\bigwedge x. x < h \implies T\ \$\$ (x, x) \neq 0 \wedge vp\ p\ (T\ \$\$ (x, x)) = 0$ 
proof -
  fix x assume x:  $x < h$ 
  obtain a j where ax:  $idx\ !\ x = (a, j)$  by (cases idx ! x)
  have mx:  $idx\ !\ x \in set\ (outer\ idx\ K\ p)$  using x len unfolding idx-def by simp
  have oqc:  $vpGge\ p\ (oq\ K\ p\ (a, j) - (pX - [:-(of\ nat\ (oroot\ p\ a) \wedge 2):]) \wedge j)\ 1$ 
    using oq-cong-large[OF p K K0 Kp mx] ax by simp
  have rtx:  $rt\ x = - (of\ nat\ (oroot\ p\ a) \wedge 2)$  unfolding rt-def ax oroot-def by
  simp
  have Te:  $T\ \$\$ (x, x) = tay\ (outerBasis\ K\ p\ (a, j))\ (- (of\ nat\ (oroot\ p\ a) \wedge 2))\ j$ 
    using Tentry[OF x x] ax rtx unfolding od-def by simp
  show  $T\ \$\$ (x, x) \neq 0 \wedge vp\ p\ (T\ \$\$ (x, x)) = 0$  unfolding Te
    by (rule outer-tay-diag[OF p oqc])
qed
have Bi:  $\bigwedge x d. x < h \implies d < h \implies vge\ p\ (B\ \$\$ (x, d))\ 0$ 
  using vpGge-outerBasis[OF p] by (auto simp: B-def h-def vpGge-iff)
have Vi:  $\bigwedge d y. d < h \implies y < h \implies vge\ p\ (V\ \$\$ (d, y))\ 0$ 
proof -

```

```

    fix d y assume d: d < h and y: y < h
    have sq: vge p (-(of-nat m ^ 2)) 0 for m by (rule vge-neg[OF vge-pow[OF p
vge-of-nat[OF p]]])
    have rt0: vge p (rt y) 0 by (simp add: rt-def sq)
    have vge p (of-nat (d choose od y) * rt y ^ (d - od y)) (0 + 0)
      by (rule vge-mul[OF p vge-of-nat[OF p] vge-pow[OF p rt0]])
    hence vf: vge p (of-nat (d choose od y) * rt y ^ (d - od y)) 0 by simp
    have V $$$ (d, y) = tay (pX ^ d) (rt y) (od y) using d y by (simp add: V-def)
    thus vge p (V $$$ (d, y)) 0 unfolding tay-pX-pow using vf by (simp add:
vge-zero)
  qed
  show ?thesis
    unfolding h-def[symmetric]
    by (rule unimodular-criterion[OF p Bc Vc Bi Vi]) (use below unit in <simp-all
add:
  T-def[symmetric]>)
  qed

lemma G-eq-Aout-large:
  assumes prime p 40 dvd K 0 < K K < p
  shows GK K = Aout K p
  proof -
    have p: prime p using assms(1) .
    obtain n where Kn: K = 40 * n using assms(2) by (elim dvdE)
    have hN: Nof K = 3 * n and hh: hof K = 37 * n by (simp-all add: Kn Nof-def
hof-def)
    have dS: degree (sqPoleProd {1..K}) = K
    proof -
      have sqPoleProd {1..K} = Dm K by (simp add: Dm-def sqPoleProd-def)
      thus ?thesis by (simp add: degree-Dm)
    qed
    show ?thesis
      proof (rule eq-matI)
        fix i j assume i: i < dim-row (Aout K p) and j: j < dim-col (Aout K p)
        have ih: i < hof K j < hof K using i j by (simp-all add: Aout-def)
        have dN: degree (Dm (Nof K) ^ 6 * pX ^ (i + j)) ≤ 6 * Nof K + (i + j)
        proof -
          have degree (Dm (Nof K) ^ 6 * pX ^ (i + j)) ≤
            degree (Dm (Nof K) ^ 6) + degree (pX ^ (i + j)) by (rule degree-mult-le)
          also have degree (Dm (Nof K) ^ 6) ≤ degree (Dm (Nof K)) * 6 by (rule
degree-power-le)
          also have degree ((pX :: rat poly) ^ (i + j)) ≤ degree (pX :: rat poly) * (i +
j)
            by (rule degree-power-le)
          finally show ?thesis by (simp add: degree-Dm degree-monom-eq)
        qed
      qed
    have dd: degree ((Dm (Nof K) ^ 6 * pX ^ (i + j)) div sqPoleProd {1..K}) ≤
      degree (Dm (Nof K) ^ 6 * pX ^ (i + j)) - degree (sqPoleProd {1..K})
      by (rule degree-div-sub[OF sqPoleProd-nonzero]) simp

```

```

    have dq: degree ((Dm (Nof K) ^ 6 * pX ^ (i + j)) div sqPoleProd {1..K}) <
2 * p - 3
    using dd dN dS ih hN hh Kn assms(3,4) by linarith
    have e: muX {1..K} (Dm (Nof K) ^ 6 * pX ^ (i + j)) =
muXmod p {1..K} (Dm (Nof K) ^ 6 * pX ^ (i + j))
    by (rule muX-eq-muXmod-of-lt[OF p dq])
    show GK K $$ (i, j) = Aout K p $$ (i, j) using ih e by (simp add: GK-def
Aout-def)
    qed (simp-all add: GK-def Aout-def)
  qed

```

lemma *large-class-entry*:

```

    assumes p: prime p and K: 40 dvd K and K0: 0 < K and Kp: K < p and
a1: 1 ≤
    a and am: a ≤ mHalf p and R: vpGge p R 0
    shows vpGge p (muXmod p (outerClassPoles K p a) R) 0
  proof -
    obtain p7: 7 ≤ p and m: 2 * mHalf p + 1 = p
    and cl: ∀ a j. a ≤ mHalf p ⟶
j ∈ outerClassPoles K p a ⟶ (j = a ∨ j = p - a) ∧
Nof K + 1 ≤ j ∧ j ≤ K
    using large-facts[OF p K K0 Kp] by blast
    have p2: p ≠ 2 using p7 by simp
    have a2: 2 * a < p using m am by linarith
    have ap: a < p and b1: 1 ≤ p - a and bp: p - a < p using a2 a1 by auto
    have sub: outerClassPoles K p a ⊆ {a, p - a} using cl am by blast
    have single: vpGge p (muXmod p {j} R) 0 if j1: 1 ≤ j and jp: j < p for j
    proof (rule muXmod-vge-of-residues[OF p p7 - R order.refl])
      show finite {j} by simp
      fix k assume k ∈ {j}
      hence k: k = j by simp
      have vge p (sqResidue {j} R j) 0 unfolding sqResidue-def
      by (simp add: eval-vge0[OF p R vge-neg-sq[OF p]])
      from vpGge-mul[OF p vpGge-C[OF this] muPole-vge-small[OF p j1 jp p2]]
      show vpGge p ([:sqResidue {j} R k:] * muPole k) 0 unfolding k by simp
    qed
  qed
  show ?thesis
  proof (cases a ∈ outerClassPoles K p a)
    case aC: True
    show ?thesis
    proof (cases p - a ∈ outerClassPoles K p a)
      case True
      have outerClassPoles K p a = {a, p - a} using sub aC True by blast
      thus ?thesis using two-pole-vge[OF p p7 a1 a2 R] by simp
    next
      case False
      have outerClassPoles K p a = {a} using sub aC False by blast
      thus ?thesis using single[OF a1 ap] by simp
    qed
  qed

```

```

next
  case naC: False
  show ?thesis
  proof (cases p - a ∈ outerClassPoles K p a)
    case True
    have outerClassPoles K p a = {p - a} using sub naC True by blast
    thus ?thesis using single[OF b1 bp] by simp
  next
    case False
    have outerClassPoles K p a = {} using sub naC False by blast
    thus ?thesis using vpGge-C[OF muModPoly-vge[OF p p7 R]] by (simp add:
muXmod-empty)
  qed
qed
qed

lemma outerGram-int-large:
  assumes p: prime p and K: 40 dvd K and K0: 0 < K and Kp: K < p and x:
x <
  length (outer-idx K p) and y: y < length (outer-idx K p)
  shows vpGge p (outerGram K p $$ (x, y)) 0
proof -
  obtain p7: 7 ≤ p and m: 2 * mHalf p + 1 = p and c0: outerCount K p 0 = 0
  and cl: ∀ a j. a ≤ mHalf p ⟶
    j ∈ outerClassPoles K p a ⟶ (j = a ∨ j = p - a) ∧
    Nof K + 1 ≤ j ∧ j ≤ K
  using large-facts[OF p K K0 Kp] by fast
  define ix where ix = outer-idx K p ! x
  define iy where iy = outer-idx K p ! y
  define a where a = fst ix
  define c where c = fst iy
  have ix: ix ∈ set (outer-idx K p) and iys: iy ∈ set (outer-idx K p)
  by (simp-all add: ix-def iy-def x y)
  have am: a ≤ mHalf p and xi: snd ix < outerCount K p a and cm: c ≤
  mHalf p and yi: snd iy < outerCount K p c
  using mem-outer-idx ix iys by (auto simp: a-def c-def)
  have a1: 1 ≤ a using xi c0 by (cases a = 0) auto
  have finC: ∧a. finite (outerClassPoles K p a) and subC: ∧a. outerClassPoles
K p a
  ⊆ tailPoles K
  by (auto simp: outerClassPoles-def tailPoles-def)
  have entry: outerGram K p $$ (x, y) =
  muXmod p {1..K} (Dm (Nof K) ^ 6 * (sqPoleProd (tailPoles K - outer-
ClassPoles K p a) *
  oq K p ix) * (sqPoleProd (tailPoles K - outerClassPoles K p c) * oq K p iy))
  using outerGram-apply[OF x y] outerBasis-eq[of K p ix] outerBasis-eq[of K p
iy]
  by (simp add: ix-def iy-def a-def c-def)
  have int0: vpGge p (Dm (Nof K) ^ 5 * sqPoleProd (tailPoles K - outer-

```

```

ClassPoles K p a)) 0
  by (intro vpGge-mul0[OF p] vpGge-pow0[OF p] vpGge-Dm[OF p] vpGge-sqPoleProd[OF
p])
  show ?thesis
  proof (cases a = c)
    case False
      have ac:  $a + c < p$  using am cm m by linarith
      have  $\{a, p - a\} \cap \{c, p - c\} = \{\}$  using False ac by auto
      moreover have outerClassPoles K p a  $\subseteq \{a, p - a\}$  outerClassPoles K p c  $\subseteq$ 
 $\{c, p - c\}$  using cl am cm by blast+
      ultimately have disj:  $\text{outerClassPoles } K \text{ } p \text{ } a \cap \text{outerClassPoles } K \text{ } p \text{ } c = \{\}$ 
by blast
      have outerGram K p  $\$ (x, y) = [\text{muModPoly } p \text{ } (Dm \text{ } (Nof \text{ } K) \text{ } ^5 * \text{sqPoleProd}$ 
 $(\text{tailPoles } K -$ 
 $(\text{outerClassPoles } K \text{ } p \text{ } a \cup \text{outerClassPoles } K \text{ } p \text{ } c)) * \text{oq } K \text{ } p \text{ } ix * \text{oq } K \text{ } p \text{ } iy):]$ 
      unfolding entry by (rule cross-class-cancel[OF finC finC disj subC subC])
      thus ?thesis
      using vpGge-C[OF muModPoly-vge[OF p p7
vpGge-mul0[OF p vpGge-mul0[OF p vpGge-mul0[OF p vpGge-pow0[OF p
vpGge-Dm[OF p]]
vpGge-sqPoleProd[OF p]] oq-vge[OF p]] oq-vge[OF p]]]]
      by simp
    next
      case True
      have same:  $\text{outerGram } K \text{ } p \text{ } \$ (x, y) =$ 
 $\text{muXmod } p \text{ } (\text{outerClassPoles } K \text{ } p \text{ } a) \text{ } (Dm \text{ } (Nof \text{ } K) \text{ } ^5 * \text{sqPoleProd } (\text{tailPoles}$ 
 $K -$ 
 $\text{outerClassPoles } K \text{ } p \text{ } a) * \text{oq } K \text{ } p \text{ } ix * \text{oq } K \text{ } p \text{ } iy)$ 
      unfolding entry True[symmetric] by (rule same-class-cancel[OF finC subC])
      have R:  $\text{vpGge } p \text{ } (Dm \text{ } (Nof \text{ } K) \text{ } ^5 * \text{sqPoleProd } (\text{tailPoles } K - \text{outerClassPoles}$ 
 $K \text{ } p \text{ } a) *$ 
 $\text{oq } K \text{ } p \text{ } ix * \text{oq } K \text{ } p \text{ } iy) \text{ } 0$ 
      by (rule vpGge-mul0[OF p vpGge-mul0[OF p int0 oq-vge[OF p]] oq-vge[OF
p]])
      show ?thesis unfolding same by (rule large-class-entry[OF p K K0 Kp a1 am
R])
    qed
  qed

```

Proposition 4.3, second part: $v \cdot p^{\wedge G}(\Delta - K) \geq 0$ for $p > K$.

```

theorem DeltaK-vpGge-large-prime:
  assumes p: prime p and K:  $40 \text{ dvd } K$  and Kp:  $K < p$ 
  shows vpGge p (DeltaK K) 0
proof (cases K = 0)
  case True
    have DeltaK 0 = 1 by (simp add: DeltaK-def GK-def hof-def)
    thus ?thesis using True by (simp add: vpGge-one)
  next
    case False

```

```

hence K0: 0 < K by simp
define h where h = hof K
define B where B = Matrix.mat (hof K) (hof K) (λ(x,
  d). Polynomial.coeff (outerBasis K p (outer-idx K p ! x)) d)
have len: length (outer-idx K p) = h
proof -
  obtain k where k: K = 40 * k using K by (elim dvdE)
  have p2: 2 < p using K0 Kp k by simp
  have hk: hof K = 37 * k and Nk: Nof K = 3 * k unfolding k hof-def Nof-def
by simp-all
  have length (outer-idx K p) = (∑ a ∈ set [0.. $mHalf$  p + 1]. outerCount K
p a)
    unfolding outer-idx-def length-concat map-map
    by (simp add: o-def sum-list-distinct-conv-sum-set)
  also have set [0.. $mHalf$  p + 1] = {.. $mHalf$  p} by auto
  also have (∑ a ≤  $mHalf$  p. outerCount K p a) = card {Nof K + 1..K}
    unfolding outerCount-def outerClassPoles-def by (rule sum-classes-card[OF
p p2]) simp
  also have ... = h using hk Nk k by (simp add: h-def)
  finally show ?thesis .
qed
have Bc: B ∈ carrier-mat h h and Gc: GK K ∈
  carrier-mat h h and OGc: outerGram K p ∈ carrier-mat h h
  by (simp-all add: B-def GK-def outerGram-def h-def len Let-def)
have uni: Determinant.det B ≠ 0 ∧ vp p (Determinant.det B) = 0
  unfolding B-def by (rule outerBasis-unimodular-large[OF p K K0 Kp])
have conj: outerGram K p = map-mat (λq. [:q:]) B * GK K *
  transpose-mat (map-mat (λq. [:q:]) B)
proof -
  have lenH: length (outer-idx K p) = hof K using len unfolding h-def by blast
  have outerGram K p = map-mat (λq. [:q:]) B * Aout K p *
    transpose-mat (map-mat (λq. [:q:]) B)
  unfolding B-def using outerGram-eq-conj-gen[OF lenH degree-outerBasis-lt-large[OF
p K K0 Kp]]
    by blast
  thus ?thesis using G-eq-Aout-large[OF p K K0 Kp] by simp
qed
have gramv: vpGge p (Determinant.det (outerGram K p)) 0
  by (rule vpGge-det-of-terms[OF p OGc]) (use outerGram-int-large[OF p K K0
Kp] len in <auto
  intro!: vpGge-prod[OF p, where b =
    λ-. 0, simplified] simp: permutes-in-image>)
show ?thesis
  unfolding DeltaK-def
  by (rule det-gram-transfer[OF p Gc Bc conjunct1[OF uni] conjunct2[OF uni]])
(use gramv conj in
  simp)
qed

```

end

theory *Zeta5-Integral-Representation*

imports *Zeta5-Outer-Range Vandermonde-Gram-Determinants Laplace-Sin-Cot-Series*
HOL-Real-Asymp.Real-Asymp

begin

23 The determinant construction and the moment representation of $\zeta(5)$

Section 2 of the paper: degree and leading coefficient of Δ_K (2.9) through the Vandermonde determinant, the positive integral representation $\mu_{\zeta(5)}(P) = \int P w$ of Proposition 2.2 (*muX-zeta5-integral*: moments from Euler's formula for $\zeta(2e + 2)$ and the pole values from the Laplace transform of *sin* and the cotangent expansion), and the positivity $\Delta_K(\zeta(5)) > 0$. Lean theories *Section2*, *Degree29*, *Prop22*.

$G\text{-}K$ may be computed with the cancelled denominator $D\text{-}K/D\text{-}N$ and numerator $D\text{-}N^5$.

lemma *G-eq-cancelled*:

$i < \text{hof } K \implies j < \text{hof } K \implies$

$GK\ K\ \$\$ (i, j) = \text{muX } \{Nof\ K + 1..K\} (Dm\ (Nof\ K) \wedge 5 * pX \wedge (i + j))$

proof –

assume $i: i < \text{hof } K$ **and** $j: j < \text{hof } K$

define N **where** $N = Nof\ K$

have $N \leq K$ **unfolding** $N\text{-def } Nof\text{-def}$ **by** *linarith*

hence $U: \{1..K\} = \{1..N\} \cup \{N + 1..K\}$ **by** *auto*

have $D: Dm\ N = \text{sqPoleProd } \{1..N\}$ **by** (*simp add: Dm-def sqPoleProd-def*)

have $e: Dm\ N \wedge 6 * pX \wedge (i + j) = (Dm\ N \wedge 5 * pX \wedge (i + j)) * \text{sqPoleProd } \{1..N\}$

unfolding D **by** (*simp add: power-Suc ac-simps numeral-eq-Suc*)

have $GK\ K\ \$\$ (i, j) = \text{muX } \{1..K\} (Dm\ N \wedge 6 * pX \wedge (i + j))$

using $i\ j$ **by** (*simp add: GK-def N-def*)

also have $\dots = \text{muX } (\{1..N\} \cup \{N + 1..K\}) ((Dm\ N \wedge 5 * pX \wedge (i + j)) * \text{sqPoleProd } \{1..N\})$

unfolding $U\ e\ ..$

also have $\dots = \text{muX } \{N + 1..K\} (Dm\ N \wedge 5 * pX \wedge (i + j))$

by (*rule muX-mul-cancel-set*) *auto*

finally show *?thesis* **by** (*simp add: N-def*)

qed

$G\text{-}K(X)$ has entries affine in X .

lemma *degree-muPole-le*: $\text{degree } (\text{muPole } j) \leq 1$

unfolding *muPole-def*

by (*force simp: degree-monom-le intro: degree-add-le degree-diff-le order-trans[OF degree-smult-le]*)

```

lemma muX-degree-le:
  assumes finite S
  shows  $\text{degree } (\text{muX } S \ P) \leq 1$ 
proof –
  have  $\text{degree } (\sum j \in S. [:sqResidue \ S \ P \ j:] * \text{muPole } j) \leq 1$ 
  proof (rule degree-sum-le[OF assms])
    fix j
    have  $\text{degree } ([:sqResidue \ S \ P \ j:] * \text{muPole } j) \leq \text{degree } (\text{muPole } j)$ 
    by (simp add: degree-smult-le)
    thus  $\text{degree } ([:sqResidue \ S \ P \ j:] * \text{muPole } j) \leq 1$  using degree-muPole-le[of j]
  by simp
  qed
  thus ?thesis unfolding muX-def by (intro degree-add-le) simp-all
qed

```

23.1 (2.9): degree and leading coefficient of Δ - K

A product of polynomials of degree ≤ 1 has degree $\leq \text{card } I$, and its top coefficient is the product of the X -coefficients.

```

lemma coeff-prod-degree-le-1:
  assumes  $\text{finite } I \wedge i. i \in I \implies \text{degree } (f \ i) \leq 1$ 
  shows  $\text{degree } (\prod i \in I. f \ i) \leq \text{card } I \wedge$ 
     $\text{Polynomial.coeff } (\prod i \in I. f \ i) (\text{card } I) = (\prod i \in$ 
       $I. \text{Polynomial.coeff } (f \ i) \ 1)$ 
  using assms
proof (induction I rule: finite-induct)
  case empty
  show ?case by simp
next
  case (insert a I)
  have IH:  $\text{degree } (\prod i \in I. f \ i) \leq \text{card } I$ 
   $\text{Polynomial.coeff } (\prod i \in I. f \ i) (\text{card } I) = (\prod i \in$ 
     $I. \text{Polynomial.coeff } (f \ i) \ 1)$ 
  using insert.IH insert.prem by auto
  have da:  $\text{degree } (f \ a) \leq 1$  using insert.prem by simp
  have deg:  $\text{degree } (f \ a * (\prod i \in I. f \ i)) \leq \text{Suc } (\text{card } I)$ 
  using degree-mult-le[of f a \prod i \in I. f i] da IH(1) by linarith
  have top:  $\text{Polynomial.coeff } (f \ a * (\prod i \in I. f \ i)) (\text{Suc } (\text{card } I)) =$ 
     $\text{Polynomial.coeff } (f \ a) \ 1 * \text{Polynomial.coeff } (\prod i \in I. f \ i) (\text{card } I)$ 
  proof –
  have  $\text{Polynomial.coeff } (f \ a * (\prod i \in I. f \ i)) (\text{Suc } (\text{card } I)) =$ 
     $(\sum k \leq \text{Suc } (\text{card } I). \text{Polynomial.coeff } (f \ a) \ k * \text{Polynomial.coeff } (\prod i \in$ 
       $I. f \ i) (\text{Suc } (\text{card } I) - k))$ 
  by (rule coeff-mult)
  also have  $\dots = (\sum k \in$ 
     $\{1\}. \text{Polynomial.coeff } (f \ a) \ k * \text{Polynomial.coeff } (\prod i \in I. f \ i) (\text{Suc } (\text{card } I)$ 
     $- k))$ 
  proof (rule sum.mono-neutral-right)
  show  $\forall k \in \{.. \text{Suc } (\text{card } I)\} - \{1\}.$ 

```



```

    Polynomial.coeff (f a) k * Polynomial.coeff (∏ i ∈ I. f i) (Suc (card I)
- k) =
    0
  proof
    fix k assume k: k ∈ {..Suc (card I)} - {1}
    show Polynomial.coeff (f a) k * Polynomial.coeff (∏ i ∈
      I. f i) (Suc (card I) - k) = 0
    proof (cases k = 0)
      case True
        have degree (∏ i ∈ I. f i) < Suc (card I) using IH(1) by linarith
        hence Polynomial.coeff (∏ i ∈ I. f i) (Suc (card I)) = 0 by (rule
coeff-eq-0)
        thus ?thesis using True by simp
      next
        case False
          hence degree (f a) < k using k da by auto
          hence Polynomial.coeff (f a) k = 0 by (rule coeff-eq-0)
          thus ?thesis by simp
        qed
      qed
    qed auto
    finally show ?thesis by simp
  qed
  show ?case using insert.hyps deg top IH by simp
qed

```

$A + X B$ as a matrix of affine polynomials.

definition *affine-mat* :: *rat mat* $\Rightarrow$ *rat mat* $\Rightarrow$ *rat poly mat* **where**
affine-mat $A B = \text{Matrix.mat } (\text{dim-row } A) (\text{dim-col } A) (\lambda(i, j). [: A \text{ $$$ } (i, j), B \text{ $$$ } (i, j) :])$

lemma *affine-mat-carrier*: $A \in \text{carrier-mat } n \ n \implies$
affine-mat $A B \in \text{carrier-mat } n \ n$
by (*simp add: affine-mat-def*)

$\det(A + X B)$ has degree $\leq n$ and X^n -coefficient $\det B$.

lemma *det-affine-mat*:

assumes $A: A \in \text{carrier-mat } n \ n$ **and** $B: B \in \text{carrier-mat } n \ n$

shows $\text{degree } (\text{Determinant.det } (\text{affine-mat } A B)) \leq n \wedge$

$\text{Polynomial.coeff } (\text{Determinant.det } (\text{affine-mat } A B)) \ n = \text{Determinant.det}$

B

proof –

let $?M = \text{affine-mat } A B$

have $Mc: ?M \in \text{carrier-mat } n \ n$ **using** A **by** (*rule affine-mat-carrier*)

have $\text{trm}: \bigwedge \sigma. \sigma \text{ permutes } \{0..<n\} \implies$

$\text{degree } (\prod i = 0..<n. ?M \text{ $$$ } (i, \sigma i)) \leq n \wedge$

$\text{Polynomial.coeff } (\prod i = 0..<n. ?M \text{ $$$ } (i, \sigma i)) \ n = (\prod i = 0..<n. B \text{ $$$ } (i, \sigma i))$

```

proof –
  fix  $\sigma :: \text{nat} \Rightarrow \text{nat}$ 
  assume  $\sigma: \sigma \text{ permutes } \{0..<n\}$ 
  have  $e: ?M \text{ \>\> } (i, \sigma i) = [A \text{ \>\> } (i, \sigma i), B \text{ \>\> } (i, \sigma i):]$ 
    if  $i: i \in \{0..<n\}$  for  $i$ 
  proof –
    have  $\sigma i \in \{0..<n\}$  using permutes-in-image[OF  $\sigma$ ]  $i$  by simp
    thus ?thesis using  $i$   $A$  by (simp add: affine-mat-def)
  qed
  have  $dl: \text{degree } [:a, b:] \leq 1$  for  $a \ b :: \text{rat}$ 
    using degree-pCons-le[of  $a$   $[:b:]$ ] by simp
  have  $\text{degree } (\prod i = 0..<n. ?M \text{ \>\> } (i, \sigma i)) \leq \text{card } \{0..<n\} \wedge$ 
     $\text{Polynomial.coeff } (\prod i = 0..<n. ?M \text{ \>\> } (i, \sigma i)) (\text{card } \{0..<n\}) =$ 
     $(\prod i = 0..<n. \text{Polynomial.coeff } (?M \text{ \>\> } (i, \sigma i)) 1)$ 
    by (rule coeff-prod-degree-le-1) (simp-all add:  $e$   $dl$ )
  moreover have  $(\prod i = 0..<n. \text{Polynomial.coeff } (?M \text{ \>\> } (i, \sigma i)) 1) = (\prod i$ 
=
     $0..<n. B \text{ \>\> } (i, \sigma i))$ 
    by (rule prod.cong) (simp-all add:  $e$ )
  ultimately show  $\text{degree } (\prod i = 0..<n. ?M \text{ \>\> } (i, \sigma i)) \leq n \wedge$ 
     $\text{Polynomial.coeff } (\prod i = 0..<n. ?M \text{ \>\> } (i, \sigma i)) n = (\prod i =$ 
     $0..<n. B \text{ \>\> } (i, \sigma i))$ 
    by simp
  qed
  have det-expand: Determinant.det  $?M =$ 
     $(\sum \sigma \in \{\sigma. \sigma \text{ permutes } \{0..<n\}\}. \text{signof } \sigma * (\prod i =$ 
     $0..<n. ?M \text{ \>\> } (i, \sigma i)))$ 
    using  $Mc$  by (simp add: Determinant.det-def')
  have detB: Determinant.det  $B =$ 
     $(\sum \sigma \in \{\sigma. \sigma \text{ permutes } \{0..<n\}\}. \text{signof } \sigma * (\prod i =$ 
     $0..<n. B \text{ \>\> } (i, \sigma i)))$ 
    using  $B$  by (simp add: Determinant.det-def')
  have  $\text{degree } (\text{Determinant.det } ?M) \leq n$ 
    unfolding det-expand
  proof (rule degree-sum-le)
    show finite  $\{\sigma. \sigma \text{ permutes } \{0..<n\}\}$  by (simp add: finite-permutations)
  next
    fix  $\sigma$  assume  $\sigma \in \{\sigma. \sigma \text{ permutes } \{0..<n\}\}$ 
    hence  $\sigma: \sigma \text{ permutes } \{0..<n\}$  by simp
    have  $\text{degree } (\text{signof } \sigma * (\prod i = 0..<n. ?M \text{ \>\> } (i, \sigma i))) \leq$ 
     $\text{degree } (\prod i = 0..<n. ?M \text{ \>\> } (i, \sigma i))$ 
    unfolding of-int-poly by (simp add: degree-smult-le)
    thus  $\text{degree } (\text{signof } \sigma * (\prod i = 0..<n. ?M \text{ \>\> } (i, \sigma i))) \leq n$ 
    using trm[OF  $\sigma$ ] by linarith
  qed
  moreover have  $\text{Polynomial.coeff } (\text{Determinant.det } ?M) n = \text{Determinant.det}$ 
 $B$ 
  proof –
    have  $\text{Polynomial.coeff } (\text{Determinant.det } ?M) n =$ 

```

```

      (∑ σ ∈ {σ. σ permutes {0..<n}}. Polynomial.coeff (signof
        σ * (∏ i = 0..<n. ?M $$ (i, σ i))) n)
    unfolding det-expand by (rule coeff-sum)
  also have ... = (∑ σ ∈
    {σ. σ permutes {0..<n}}. signof σ * (∏ i =
      0..<n. B $$ (i, σ i)))
  proof (rule sum.cong[OF refl])
    fix σ assume σ ∈ {σ. σ permutes {0..<n}}
    hence σ: σ permutes {0..<n} by simp
    show Polynomial.coeff (signof σ * (∏ i = 0..<n. ?M $$ (i, σ i))) n =
      signof σ * (∏ i = 0..<n. B $$ (i, σ i))
    unfolding of-int-poly using trm[OF σ] by simp
  qed
  also have ... = Determinant.det B using detB by simp
  finally show ?thesis .
qed
ultimately show ?thesis by simp
qed

```

The constant and X -coefficient matrices of G - K .

definition $GA :: nat \Rightarrow rat\ mat$ **where**

$GA\ K = Matrix.mat\ (hof\ K)\ (hof\ K)\ (\lambda(i, j). Polynomial.coeff\ (GK\ K\ \$\$ (i, j))\ 0)$

definition $GB :: nat \Rightarrow rat\ mat$ **where**

$GB\ K = Matrix.mat\ (hof\ K)\ (hof\ K)\ (\lambda(i, j). Polynomial.coeff\ (GK\ K\ \$\$ (i, j))\ 1)$

lemma *degree-le-1-pCons*:

assumes d : $degree\ p \leq 1$

shows $p = [:Polynomial.coeff\ p\ 0,\ Polynomial.coeff\ p\ 1:]$

proof (rule poly-eqI)

fix n

show $Polynomial.coeff\ p\ n = Polynomial.coeff\ [:Polynomial.coeff\ p\ 0,\ Polynomial.coeff\ p\ 1:]\ n$

proof (cases $n \leq 1$)

case *True*

thus ?thesis **by** (cases n) *auto*

next

case *False*

hence n : $n = Suc\ (Suc\ (n - 2))$ **by** *arith*

have $degree\ p < n$ **using** d *False* **by** *linarith*

hence $Polynomial.coeff\ p\ n = 0$ **by** (rule *coeff-eq-0*)

moreover **have** $Polynomial.coeff\ [:Polynomial.coeff\ p\ 0,\ Polynomial.coeff\ p\ 1:]\ n = 0$

by (*subst* n) *simp*

ultimately **show** ?thesis **by** *simp*

qed

qed

lemma *GK-eq-affine*: $GK\ K = \text{affine-mat}\ (GA\ K)\ (GB\ K)$
proof (*rule eq-matI*)
 fix $i\ j$
 assume $i < \text{dim-row}\ (\text{affine-mat}\ (GA\ K)\ (GB\ K))\ j < \text{dim-col}\ (\text{affine-mat}\ (GA\ K)\ (GB\ K))$
 hence ij : $i < \text{hof}\ K\ j < \text{hof}\ K$ **by** (*simp-all add: affine-mat-def GA-def*)
 have e : $GK\ K\ \$\$ (i, j) = \text{muX}\ \{1..K\}\ (Dm\ (Nof\ K) \wedge 6 * pX \wedge (i + j))$ **using**
 ij
 by (*simp add: GK-def*)
 have $\text{degree}\ (GK\ K\ \$\$ (i, j)) \leq 1$ **unfolding** e **by** (*rule muX-degree-le*) *simp*
 hence $GK\ K\ \$\$ (i, j) = [\text{Polynomial.coeff}\ (GK\ K\ \$\$ (i, j))\ 0, \text{Polynomial.coeff}\ (GK\ K\ \$\$ (i, j))\ 1:]$
 by (*rule degree-le-1-pCons*)
 thus $GK\ K\ \$\$ (i, j) = \text{affine-mat}\ (GA\ K)\ (GB\ K)\ \$\$ (i, j)$
 using ij **by** (*simp add: affine-mat-def GA-def GB-def*)
qed (*simp-all add: affine-mat-def GA-def GK-def*)

lemma *muPole-coeff-one*: $\text{Polynomial.coeff}\ (\text{muPole}\ j)\ 1 = \text{of-nat}\ j \wedge 4$
unfolding *muPole-def* **by** *simp*

The residue weight at the pole j .

definition $wt :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{rat}$ **where**

$$wt\ K\ j = \text{of-nat}\ j \wedge 4 * \text{poly}\ (Dm\ (Nof\ K))\ (-\ (\text{of-nat}\ j \wedge 2)) \wedge 5 /$$

$$(\prod_{k \in \{Nof\ K + 1..K\} - \{j\}} \text{of-nat}\ k \wedge 2 - \text{of-nat}\ j \wedge 2)$$

lemma *GB-apply*:

assumes $i < \text{hof}\ K\ j < \text{hof}\ K$
shows $GB\ K\ \$\$ (i, j) =$
 $(\sum_{l \in \{Nof\ K + 1..K\}} (-\ (\text{of-nat}\ l \wedge 2)) \wedge i * (-\ (\text{of-nat}\ l \wedge 2)) \wedge j * wt\ K\ l)$
proof –
 have $GB\ K\ \$\$ (i, j) = \text{Polynomial.coeff}\ (\text{muX}\ \{Nof\ K + 1..K\}\ (Dm\ (Nof\ K) \wedge 5 * pX \wedge (i + j)))\ 1$
 using *assms* **by** (*simp add: GB-def G-eq-cancelled*)
 also have $\dots = (\sum_{l \in \{Nof\ K + 1..K\}} \text{sqResidue}\ \{Nof\ K + 1..K\}\ (Dm\ (Nof\ K) \wedge 5 * pX \wedge (i + j))\ l * \text{of-nat}\ l \wedge 4)$
unfolding *muX-def* **by** (*simp add: coeff-sum muPole-coeff-one[unfolded One-nat-def]*)
 also have $\dots = (\sum_{l \in \{Nof\ K + 1..K\}} (-\ (\text{of-nat}\ l \wedge 2)) \wedge i * (-\ (\text{of-nat}\ l \wedge 2)) \wedge j * wt\ K\ l)$
by (*intro sum.cong refl*) (*simp add: sqResidue-def wt-def poly-monom power-add mult-ac*)
finally show *?thesis* .
qed

The Vandermonde sign identity $\prod -t \prod -\{s \neq t\} (v\ t - v\ s) = (-1)^{\wedge\{n(n-1)/2\}}$
 $(\prod -\{i < j\} (v\ j - v\ i))^{\wedge 2}$.

lemma *prod-erase-sub*:

fixes $v :: \text{nat} \Rightarrow \text{rat}$

shows $(\prod t < n. \prod s \in \{..<n\} - \{t\}. v\ t - v\ s) =$
 $(-1) \wedge (n * (n - 1) \text{ div } 2) * (\prod i < n. \prod j \in \{j. i < j \wedge$
 $j < n\}. v\ j - v\ i) \wedge 2$

proof –
have *split*: $\bigwedge t. t < n \implies \{..<n\} - \{t\} = \{..<t\} \cup \{j. t < j \wedge$
 $j < n\}$ **by** *auto*
have *e1*: $(\prod t < n. \prod s \in \{..<n\} - \{t\}. v\ t - v\ s) =$
 $(\prod t < n. \prod s < t. v\ t - v\ s) * (\prod t < n. \prod s \in \{j. t <$
 $j \wedge j < n\}. v\ t - v\ s)$
unfolding *prod.distrib[symmetric]*
proof (*rule prod.cong[OF refl]*)
fix *t* **assume** $t \in \{..<n\}$
hence $t < n$ **by** *simp*
have *fin*: *finite* $\{j. t < j \wedge j < n\}$ **by** (*rule finite-subset[of - {..<n}]*) *auto*
show $(\prod s \in \{..<n\} - \{t\}. v\ t - v\ s) = (\prod s < t. v\ t - v\ s) * (\prod s \in$
 $\{j. t < j \wedge j < n\}. v\ t - v\ s)$
unfolding *split[OF t]* **by** (*rule prod.union-disjoint*) (*auto simp: fin*)
qed

have *e2*: $(\prod t < n. \prod s < t. v\ t - v\ s) = (\prod i < n. \prod j \in \{j. i <$
 $j \wedge j < n\}. v\ j - v\ i)$

proof –
have *L1*: $(\prod t < n. \prod s < t. v\ t - v\ s) = (\prod (t, s) \in$
 $\text{Sigma } \{..<n\} (\lambda t. \{..<t\}). v\ t - v\ s)$
by (*rule prod.Sigma*) *auto*
have *L2*: $(\prod i < n. \prod j \in \{j. i < j \wedge j < n\}. v\ j - v\ i) =$
 $(\prod (i, j) \in \text{Sigma } \{..<n\} (\lambda i. \{j. i < j \wedge j < n\}). v\ j - v\ i)$
by (*rule prod.Sigma*) *auto*
have *L3*: $(\prod (t, s) \in \text{Sigma } \{..<n\} (\lambda t. \{..<t\}). v\ t - v\ s) =$
 $(\prod (i, j) \in \text{Sigma } \{..<n\} (\lambda i. \{j. i < j \wedge j < n\}). v\ j - v\ i)$
by (*rule prod.reindex-bij-witness[where i = prod.swap and j = prod.swap]*)
auto

show *?thesis* **unfolding** *L1 L2 L3 ..*

qed

have *e3*: $(\prod t < n. \prod s \in \{j. t < j \wedge j < n\}. v\ t - v\ s) =$
 $(-1) \wedge (n * (n - 1) \text{ div } 2) * (\prod i < n. \prod j \in \{j. i < j \wedge$
 $j < n\}. v\ j - v\ i)$

proof –
have *card*: *card* $\{j. t < j \wedge j < n\} = n - \text{Suc } t$ **for** *t*
proof –
have $\{j. t < j \wedge j < n\} = \{t < ..<n\}$ **by** *auto*
thus *?thesis* **by** *simp*
qed

have *inner*: $(\prod s \in \{j. t < j \wedge j < n\}. v\ t - v\ s) =$
 $(-1) \wedge (n - \text{Suc } t) * (\prod j \in \{j. t < j \wedge j < n\}. v\ j - v\ t)$ **for** *t*

proof –
have $(\prod s \in \{j. t < j \wedge j < n\}. v\ t - v\ s) = (\prod s \in \{j. t < j \wedge$
 $j < n\}. (-1) * (v\ s - v\ t))$
by *simp*
also have $\dots = (-1) \wedge (n - \text{Suc } t) * (\prod j \in \{j. t < j \wedge$

```

      j < n}. v j - v t)
    by (subst prod.distrib) (simp add: card)
  finally show ?thesis .
qed
have sum: (∑ t < n. n - Suc t) = n * (n - 1) div 2
proof -
  have (∑ t < n. n - Suc t) = (∑ t < n. t)
    using sum.nat-diff-reindex[of λx. x n] by simp
  also have ... = ∑ {0..<n} by (simp add: lessThan-atLeast0)
  also have ... = n * (n - 1) div 2 by (simp add: Sum-Ico-nat)
  finally show ?thesis .
qed
have (∏ t < n. ∏ s ∈ {j. t < j ∧ j < n}. v t - v s) =
  (∏ t < n. (-1) ^ (n - Suc t)) * (∏ i < n. ∏ j ∈ {j. i < j ∧
    j < n}. v j - v i)
  unfolding inner by (rule prod.distrib)
also have (∏ t < n. (-1 :: rat) ^ (n - Suc t)) = (-1) ^ (∑ t < n. n - Suc
t)
  by (rule power-sum[symmetric])
  finally show ?thesis unfolding sum .
qed
show ?thesis unfolding e1 e2 e3 by (simp add: power2-eq-square)
qed

lemma Dm-eval-ne-zero: Nof K < j ⟹ poly (Dm (Nof K)) (- (of-nat j ^ 2)) ≠
0
proof -
  assume j: Nof K < j
  have fac: poly (pX + [:of-nat k ^ 2:]) (- (of-nat j ^ 2)) ≠ 0 if k: k ∈ {1..Nof
K}
  for k
  proof -
    have k < j using k j by auto
    hence lt: (of-nat k :: rat) ^ 2 < of-nat j ^ 2 by (intro power-strict-mono) auto
    have poly (pX + [:of-nat k ^ 2:]) (- (of-nat j ^ 2)) = (of-nat k ^ 2 - of-nat j
^ 2 :: rat)
      by (simp add: poly-monom)
    thus ?thesis using lt ⟨k < j⟩ by simp
  qed
  show ?thesis unfolding Dm-def poly-prod prod-zero-iff[OF finite-atLeastAtMost]
using fac by blast
qed

lemma hof-eq-diff: 40 dvd K ⟹ hof K = K - Nof K
  by (auto simp: hof-def Nof-def elim!: dvdE)

lemma field-cancel-aux:
  fixes d c P :: rat
  assumes d: d ≠ 0 and c: c * c = 1

```

shows $d * (P / (c * d \wedge 2)) * d = c * P$
proof –
 have $c0: c \neq 0$ **using** c **by** *auto*
 have $d * (P / (c * d \wedge 2)) * d = P / c$ **using** $d\ c0$ **by** (*simp add: field-simps power2-eq-square*)
 also have $\dots = c * (c * P) / c$ **using** c **by** (*simp add: mult.assoc[symmetric]*)
 also have $\dots = c * P$ **by** (*rule nonzero-mult-div-cancel-left[OF c0]*)
 finally **show** *?thesis* .
qed

lemma *det-GB*:

assumes $40\ \text{dvd}\ K$
 shows $\text{Determinant.det}\ (GB\ K) = (-1)^{\wedge} (\text{hof}\ K * (\text{hof}\ K - 1) \text{div}\ 2) * (\prod j = \text{Nof}\ K + 1..K. \text{of-nat}\ j^{\wedge} 4 * \text{poly}\ (Dm\ (\text{Nof}\ K)) (- (\text{of-nat}\ j^{\wedge} 2)))^{\wedge} 5)$

proof –
 define $h\ N$ **where** $h = \text{hof}\ K$ **and** $N = \text{Nof}\ K$
 have $hN: h = K - N$ **using** *hof-eq-diff[OF assms]* **by** (*simp add: h-def N-def*)
 have $NK: N \leq K$ **by** (*simp add: N-def Nof-def*)
 define xp **where** $xp\ t = N + 1 + t$ **for** t — the poles, indexed by $t < h$
 define x **where** $x\ t = - (\text{of-nat}\ (xp\ t)^{\wedge} 2 :: \text{rat})$ **for** t
 define V **where** $V = \text{Matrix.mat}\ h\ h\ (\lambda(t, i). x\ t^{\wedge} i)$
 define D **where** $D = \text{Matrix.mat}\ h\ h\ (\lambda(t, s). \text{if}\ t = s \text{ then}\ wt\ K\ (xp\ t) \text{ else}\ 0)$
 have $\text{img}: \{N + 1..K\} = xp\ ' \{..<h\}$ **and** $\text{inj}: \text{inj-on}\ xp\ \{..<h\}$
proof –
 show $\{N + 1..K\} = xp\ ' \{..<h\}$
proof
 show $xp\ ' \{..<h\} \subseteq \{N + 1..K\}$ **using** $hN\ NK$ **by** (*auto simp: xp-def*)
 show $\{N + 1..K\} \subseteq xp\ ' \{..<h\}$
proof
 fix k **assume** $k: k \in \{N + 1..K\}$
 hence $k = xp\ (k - (N + 1))$ $k - (N + 1) \in \{..<h\}$ **using** hN **by** (*auto simp: xp-def*)
 thus $k \in xp\ ' \{..<h\}$ **by** (*rule image-eqI*)
qed
qed
 show $\text{inj-on}\ xp\ \{..<h\}$ **by** (*rule inj-onI*) (*simp add: xp-def*)
qed
 have $x\text{-inj}: \text{inj-on}\ x\ \{..<h\}$
proof (*rule inj-onI*)
 fix $a\ b$ **assume** $a \in \{..<h\}\ b \in \{..<h\}$ **and** $e: x\ a = x\ b$
 from e **have** $xp\ a = xp\ b$ **by** (*simp add: x-def*)
 thus $a = b$ **by** (*simp add: xp-def*)
qed
 have $Vc: V \in \text{carrier-mat}\ h\ h$ **and** $Dc: D \in \text{carrier-mat}\ h\ h$
by (*simp-all add: V-def D-def*)
 have $GB\text{-eq}: GB\ K = \text{transpose-mat}\ V * D * V$
proof –
 have $VD: \text{transpose-mat}\ V * D = \text{Matrix.mat}\ h\ h\ (\lambda(i, t). x\ t^{\wedge} i * wt\ K\ (xp\ t))$

```

t))
proof (rule eq-matI)
  fix a b assume a: a < dim-row (Matrix.mat h h (λ(i, t). x t ^ i * wt K (xp
t)))
    and b: b < dim-col (Matrix.mat h h (λ(i, t). x t ^ i * wt K (xp t)))
    have (transpose-mat V * D) $$ (a, b) = (∑ s ∈ {0..using a b Vc Dc by (simp add: scalar-prod-def)
    also have ... = (∑ s ∈ {0..using a b by (intro sum.cong refl) (simp add: V-def D-def)
    also have ... = x b ^ a * wt K (xp b) using b by simp
    finally show (transpose-mat V * D) $$ (a, b) =
      Matrix.mat h h (λ(i, t). x t ^ i * wt K (xp t)) $$ (a, b)
      using a b by simp
qed (use Vc Dc in simp-all)
show ?thesis
proof (rule eq-matI)
  fix i j assume i: i < dim-row (transpose-mat V * D * V) and j: j <
dim-col (transpose-mat V * D * V)
  have ih: i < h and jh: j < h using i j Vc Dc by auto
  have (transpose-mat V * D * V) $$ (i, j) = (∑ t ∈
{0..unfolding VD using ih jh Vc by (simp add: scalar-prod-def V-def)
  also have ... = (∑ t ∈
{..unfolding lessThan-atLeast0 by (intro sum.cong refl) (simp add: x-def
mult-ac)
  also have ... = (∑ l ∈
{N + 1..K}. (- (of-nat l ^ 2)) ^ i * (- (of-nat l ^ 2)) ^ j * wt K l)
    unfolding img sum.reindex[OF inj] by (simp add: o-def)
  also have ... = GB K $$ (i, j) using GB-apply[of i K j] ih jh by (simp add:
h-def N-def)
  finally show GB K $$ (i, j) = (transpose-mat V * D * V) $$ (i, j) by (rule
sym)
  qed (use Vc Dc in ⟨simp-all add: GB-def h-def⟩)
qed
have detV: Determinant.det V = (∏ i < h. ∏ j ∈ {j. i < j ∧
j < h}. x j - x i)
  unfolding V-def by (rule det-vandermonde-jnf)
have detV-ne: Determinant.det V ≠ 0
proof -
  have nz: x j - x i ≠ 0 if i < h i < j j < h for i j
  proof
    assume x j - x i = 0
    hence x j = x i by simp
    hence j = i using inj-onD[OF x-inj] that by simp
    thus False using that by simp
  qed

```



```

    have fin: finite {j. i < j ∧ j < h} for i by (rule finite-subset[of - {..

```

```

qed
have num: (∏ t < h. of-nat (xp t) ^ 4 * poly (Dm N) (- (of-nat (xp t) ^ 2))
^ 5) =
  (∏ j = N + 1..K. of-nat j ^ 4 * poly (Dm N) (- (of-nat j ^ 2)) ^ 5)
  unfolding img prod.reindex[OF inj] by (simp add: o-def)
  have Determinant.det (GB K) = Determinant.det V * Determinant.det D *
  Determinant.det V
  proof -
    have Tc: transpose-mat V ∈ carrier-mat h h using Vc by simp
    have TDc: transpose-mat V * D ∈ carrier-mat h h using Tc Dc by (rule
  mult-carrier-mat)
    show ?thesis unfolding GB-eq det-mult[OF TDc Vc] det-mult[OF Tc Dc]
  det-transpose[OF Vc] ..
  qed
  also have ... = (-1) ^ (h * (h - 1) div 2) *
    (∏ j = N + 1..K. of-nat j ^ 4 * poly (Dm N) (- (of-nat j ^ 2)) ^ 5)
  proof -
    have s: ((-1 :: rat) ^ (h * (h - 1) div 2)) * (-1) ^ (h * (h - 1) div 2) = 1
    by (simp add: power-mult-distrib[symmetric])
    show ?thesis unfolding detD prod-wt num by (rule field-cancel-aux[OF detV-ne
  s])
  qed
  finally show ?thesis by (simp add: h-def N-def)
qed

```

(2.9): Δ - K has degree h and leading coefficient $(-1)^{\{h(h-1)/2\}} \prod \{-j=N+1\}^K j^4 D-N(-j^2)^5$. The X -part of G - K is $V^T \text{diag}(w) V$ with V Vandermonde.

theorem *Delta-degree-lead-coeff*:

```

assumes 40 dvd K
shows degree (DeltaK K) = hof K ∧
  lead-coeff (DeltaK K) = (-1) ^ (hof K * (hof K - 1) div 2) *
  (∏ j = Nof K + 1..K. of-nat j ^ 4 * poly (Dm (Nof K)) (- (of-nat j ^
  2)) ^ 5)
proof -
  have Ac: GA K ∈ carrier-mat (hof K) (hof K) and Bc: GB K ∈ carrier-mat
  (hof K) (hof K)
  by (simp-all add: GA-def GB-def)
  have D: DeltaK K = Determinant.det (affine-mat (GA K) (GB K))
  by (simp add: DeltaK-def GK-eq-affine)
  have deg: degree (DeltaK K) ≤ hof K
  and cf: Polynomial.coeff (DeltaK K) (hof K) = Determinant.det (GB K)
  using det-affine-mat[OF Ac Bc] by (simp-all add: D)
  have ne: Determinant.det (GB K) ≠ 0
  unfolding det-GB[OF assms]
  using Dm-eval-ne-zero[of K] by auto
  have degree (DeltaK K) = hof K
  using deg le-degree[of DeltaK K hof K] cf ne by simp
  thus ?thesis using cf det-GB[OF assms] by simp

```

qed

lemma *SK-pos*: $0 < SK\ K$
 unfolding *SK-def*
 by (force intro: divide-pos-pos mult-pos-pos prod-pos zero-less-power fact-gt-zero)

23.2 Proposition 2.2: the positive integral representation

lemma *weight-summable*:
 assumes $0 < y$
 shows *summable* ($\lambda l. \text{real } (l + 1) ^ 4 * \exp (- 2 * \pi * \text{real } (l + 1) * y)$)
 proof -
 define *s* where $s = \exp (- \pi * y)$
 have *s0*: $0 < s$ and *s1*: $s < 1$ using *assms* by (auto simp: *s-def*)
 define *C* where $C = 256 / (\pi * y) ^ 4$
 have *C0*: $0 < C$ using *assms* by (simp add: *C-def*)
 have *bound*: $\text{real } (l + 1) ^ 4 * \exp (- 2 * \pi * \text{real } (l + 1) * y) \leq C * s ^ (l + 1)$ for *l*
 proof -
 define *x* where $x = \pi * y * \text{real } (l + 1)$
 have *x0*: $0 \leq x$ using *assms* by (simp add: *x-def*)
 have $(x / 4) ^ 4 \leq (1 + x / 4) ^ 4$ using *x0* by (intro power-mono) simp-all
 also have $\dots \leq \exp (x / 4) ^ 4$
 by (intro power-mono exp-ge-add-one-self) (simp add: *x0*)
 also have $\dots = \exp x$ by (simp add: exp-of-nat-mult[symmetric])
 finally have *e*: $(x / 4) ^ 4 \leq \exp x$.
 have $\text{real } (l + 1) ^ 4 = C * (x / 4) ^ 4$ using *assms* unfolding *C-def* *x-def*
 by (auto simp: power-mult-distrib power-divide pi-gt-zero)
 also have $\dots \leq C * \exp x$ using *C0* *e* by (intro mult-left-mono) simp-all
 finally have *A*: $\text{real } (l + 1) ^ 4 \leq C * \exp x$.
 have *B*: $\exp (- 2 * \pi * \text{real } (l + 1) * y) = s ^ (l + 1) * \exp (- x)$
 proof -
 have *e1*: $s ^ (l + 1) = \exp (\text{real } (l + 1) * (- \pi * y))$
 unfolding *s-def* by (rule exp-of-nat-mult[symmetric])
 have $s ^ (l + 1) * \exp (- x) = \exp (\text{real } (l + 1) * (- \pi * y) + - x)$
 by (subst *e1*) (simp add: exp-add[symmetric])
 also have $\text{real } (l + 1) * (- \pi * y) + - x = - 2 * \pi * \text{real } (l + 1) * y$
 by (simp add: *x-def* algebra-simps)
 finally show ?thesis ..
 qed
 have $\text{real } (l + 1) ^ 4 * \exp (- 2 * \pi * \text{real } (l + 1) * y) =$
 $(\text{real } (l + 1) ^ 4 * \exp (- x)) * s ^ (l + 1)$
 by (subst *B*) (simp add: mult-ac)
 also have $\dots \leq (C * \exp x * \exp (- x)) * s ^ (l + 1)$
 using *A* *s0* by (intro mult-right-mono mult-right-mono[OF *A*]) simp-all
 also have $\dots = C * s ^ (l + 1)$ by (simp add: exp-minus)
 finally show ?thesis .
 qed
 have *summable* ($\lambda l. s ^ \text{Suc } l$) using *summable-geometric*[of *s*] *s0* *s1*

by (simp add: summable-Suc-iff)
 hence summable $(\lambda l. C * s ^ (l + 1))$ by (intro summable-mult) simp
 thus ?thesis
 by (rule summable-comparison-test'[of - 0]) (use bound assms in ‹auto intro!:
 mult-nonneg-nonneg›)
 qed

The weight is positive on $(0, \infty)$.

theorem weight-pos: $0 < y \implies 0 < \text{weight } y$

proof –

assume $y: 0 < y$
 have $0 < \text{suminf } (\lambda l. \text{real } (l + 1) ^ 4 * \exp (- 2 * \pi * \text{real } (l + 1) * y))$
 by (rule suminf-pos[OF weight-summable[OF y]]) simp
 thus $0 < \text{weight } y$ using y unfolding weight-def by simp
 qed

lemma weight-eq: $\text{weight } y = (2 * \pi) ^ 4 * y ^ 5 / 12 * (\sum l. \text{real } (l + 1) ^ 4$
 $* \exp (- 2 *$
 $\pi * \text{real } (l + 1) * y))$
 by (simp add: weight-def)

$\int_{-0}^{\infty} w(y)/(y^2 + a^2) dy = 1/(2a) - 1/4 + a^4 \sum_{n \geq 0} (a + n$
 $+ 1)^{-5}$.

lemma pole:

assumes $0 < a (\lambda n. 1 / (a + (\text{real } n + 1)) ^ 5) \text{ sums } Z$
 shows $\text{set-integrable lborel } \{0 < ..\} (\lambda y. \text{weight } y / (y ^ 2 + a ^ 2)) \wedge$
 $\text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda y. \text{weight } y / (y ^ 2 + a ^ 2)) =$
 $1 / (2 * a) - 1 / 4 + a ^ 4 * Z$

proof –

note $a = \text{assms}(1)$ and $Z = \text{assms}(2)$
 define c where $c \ l = 2 * \pi * (\text{real } l + 1)$ for l
 have $cp: 0 < c \ l$ for l by (simp add: c-def)
 define F where $F \ u = u ^ 4 * \exp (- (a * u)) * (1 / 4 + 1 / (2 * (\exp u -$
 $1)) - 1 / (2 * u))$
 for u
 have $UI: \text{set-integrable lborel } \{0 < ..\} F \wedge$
 $\text{set-lebesgue-integral lborel } \{0 < ..\} F = 6 / a ^ 5 - 3 / a ^ 4 + 12 * Z$
 unfolding F -def by (rule u-integral[OF a Z])
 define h where $h = (\lambda l \ u. u ^ 5 * \exp (- (a * u)) / (u ^ 2 + c \ l ^ 2))$
 have $hi: \text{set-integrable lborel } \{0 < ..\} (h \ l)$ for l unfolding h -def
 by (rule phi-integrable[OF cp a])
 have $hnn: 0 \leq h \ l \ u$ if $0 < u$ for $l \ u$ using that by (simp add: h-def)
 have $hsum: (\lambda l. h \ l \ u) \text{ sums } F \ u$ if $u: 0 < u$ for u
proof –
 have $s: (\lambda l. u ^ 4 * \exp (- (a * u)) * (u / (u ^ 2 + (2 * \pi * (\text{real } l +$
 $1)) ^ 2))) \text{ sums }$
 $(u ^ 4 * \exp (- (a * u)) * (1 / 4 + 1 / (2 * (\exp u - 1)) - 1 / (2 *$
 $u)))$
 by (rule sums-mult[OF S-series[OF u]])

```

    have e: (λl. u ^ 4 * exp (- (a * u)) * (u / (u ^ 2 + (2 * pi * (real l + 1)) ^
2))) =
      (λl. h l u)
    by (rule ext) (simp add: h-def c-def eval-nat-numeral mult-ac)
    show ?thesis using s unfolding e F-def .
  qed
  have hs: summable (λl. h l u) if 0 < u for u using hsum[OF that]
    by (rule sums-summable)
  have hS: set-integrable lborel {0<..} (λu. ∑ l. h l u)
    by (rule set-integrable-cong-Ioi[OF conjunct1[OF UI]]) (rule sums-unique[OF
hsum])
  have hV: set-lebesgue-integral lborel {0<..} (λu. ∑ l. h l u) =
    6 / a ^ 5 - 3 / a ^ 4 + 12 * Z
  proof -
    have set-lebesgue-integral lborel {0<..} (λu. ∑ l. h l u) =
      set-lebesgue-integral lborel {0<..} F
    by (rule set-integral-cong-Ioi) (rule sums-unique[OF hsum, symmetric])
    thus ?thesis using conjunct2[OF UI] by simp
  qed
  have uS: (λl. set-lebesgue-integral lborel {0<..} (h l)) sums (6 / a ^ 5 - 3 / a
^ 4 +
    12 * Z)
    using suminf-integral-nn[OF hi hnn hs hS] unfolding hV .
  define g where g = (λl y. c l ^ 4 / 12 * (y ^ 5 * exp (- (c l * y)) / (y ^ 2 +
a ^ 2)))
  have gi: set-integrable lborel {0<..} (g l) for l
    unfolding g-def by (intro set-integrable-mult-right phi-integrable[OF a cp])
  have gv: set-lebesgue-integral lborel {0<..} (g l) =
    a ^ 4 / 12 * set-lebesgue-integral lborel {0<..} (h l) for l
  proof -
    have set-lebesgue-integral lborel {0<..} (g l) = c l ^ 4 / 12 * Phi a (c l)
    unfolding g-def Phi-def set-integral-mult-right ..
    also have ... = a ^ 4 / 12 * Phi (c l) a using phi-symm[OF a cp[of l]] by
simp
    also have Phi (c l) a = set-lebesgue-integral lborel {0<..} (h l) unfolding
Phi-def h-def ..
    finally show ?thesis .
  qed
  have gsum: (λl. set-lebesgue-integral lborel {0<..} (g l)) sums (a ^ 4 / 12 * (6
/
    a ^ 5 - 3 / a ^ 4 + 12 * Z))
    unfolding gv by (rule sums-mult[OF uS])
  have gnn: 0 ≤ g l y if 0 < y for l y using that cp[of l] by (simp add: g-def)
  have c4: c l ^ 4 = (2 * pi) ^ 4 * real (l + 1) ^ 4 for l
    by (simp add: c-def power-mult-distrib add.commute)
  have wpt: weight y / (y ^ 2 + a ^ 2) = (∑ l. g l y) if y: 0 < y for y
  proof -
    have s: summable (λl. real (l + 1) ^ 4 * exp (- 2 * pi * real (l + 1) * y))
    by (rule weight-summable[OF y])

```

```

have weight  $y / (y^2 + a^2) = ((2 * \pi)^4 * y^5 / 12 / (y^2 + a^2)) * (\sum l. \text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y))$ 
unfolding weight-eq by simp
also have ... =  $(\sum l. ((2 * \pi)^4 * y^5 / 12 / (y^2 + a^2)) * (\text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y)))$ 
by (rule suminf-mult[OF s, symmetric])
also have  $(\lambda l. ((2 * \pi)^4 * y^5 / 12 / (y^2 + a^2)) * (\text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y))) = (\lambda l. g \ l \ y)$ 
unfolding g-def c4 by (rule ext) (simp add: c-def mult-ac)
finally show ?thesis .
qed
have gs: summable  $(\lambda l. g \ l \ y)$  if  $y: 0 < y$  for  $y$ 
proof -
have summable  $(\lambda l. ((2 * \pi)^4 * y^5 / 12 / (y^2 + a^2)) * (\text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y)))$ 
by (rule summable-mult[OF weight-summable[OF y]])
also have  $(\lambda l. ((2 * \pi)^4 * y^5 / 12 / (y^2 + a^2)) * (\text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y))) = (\lambda l. g \ l \ y)$ 
unfolding g-def c4 by (rule ext) (simp add: c-def mult-ac)
finally show ?thesis .
qed
have GI: set-integrable lborel  $\{0 < ..\}$   $(\lambda y. \sum l. g \ l \ y) \wedge$ 
set-lebesgue-integral lborel  $\{0 < ..\}$   $(\lambda y. \sum l. g \ l \ y) =$ 
 $(\sum l. \text{set-lebesgue-integral lborel } \{0 < ..\} (g \ l))$ 
by (rule integral-suminf-nn[OF gi gnn gs sums-summable[OF gsum]])
have val:  $a^4 / 12 * (6 / a^5 - 3 / a^4 + 12 * Z) = 1 / (2 * a) - 1 / 4 + a^4 * Z$ 
using a by (simp add: field-simps eval-nat-numeral)
show ?thesis
proof
show set-integrable lborel  $\{0 < ..\}$   $(\lambda y. \text{weight } y / (y^2 + a^2))$ 
by (rule set-integrable-cong-Ioi[OF conjunct1[OF GI]]) (rule wpt[symmetric])
have set-lebesgue-integral lborel  $\{0 < ..\}$   $(\lambda y. \text{weight } y / (y^2 + a^2)) =$ 
set-lebesgue-integral lborel  $\{0 < ..\}$   $(\lambda y. \sum l. g \ l \ y)$ 
by (rule set-integral-cong-Ioi) (rule wpt)
also have ... =  $1 / (2 * a) - 1 / 4 + a^4 * Z$ 
using conjunct2[OF GI] sums-unique[OF gsum] val by simp
finally show set-lebesgue-integral lborel  $\{0 < ..\}$   $(\lambda y. \text{weight } y / (y^2 + a^2)) =$ 
 $1 / (2 * a) - 1 / 4 + a^4 * Z$  .
qed
qed

```

23.3 Moments and the pole values

lemma *re-map-add*: $\text{map-poly } (\text{of-rat} :: \text{rat} \Rightarrow \text{real}) (P + Q) =$
 $\text{map-poly of-rat } P + \text{map-poly of-rat } Q$
by (*rule poly-eqI*) (*simp add: coeff-map-poly of-rat-add*)

lemma *re-map-diff*: $\text{map-poly } (\text{of-rat} :: \text{rat} \Rightarrow \text{real}) (P - Q) =$
 $\text{map-poly of-rat } P - \text{map-poly of-rat } Q$
by (*rule poly-eqI*) (*simp add: coeff-map-poly of-rat-diff*)

lemma *re-map-mult*: $\text{map-poly } (\text{of-rat} :: \text{rat} \Rightarrow \text{real}) (P * Q) =$
 $\text{map-poly of-rat } P * \text{map-poly of-rat } Q$
by (*rule poly-eqI*) (*simp add: coeff-map-poly coeff-mult of-rat-sum of-rat-mult*)

lemma *real-eval-add*: $\text{real-eval } (P + Q) \ x = \text{real-eval } P \ x + \text{real-eval } Q \ x$
by (*simp add: real-eval-def re-map-add*)

lemma *real-eval-diff*: $\text{real-eval } (P - Q) \ x = \text{real-eval } P \ x - \text{real-eval } Q \ x$
by (*simp add: real-eval-def re-map-diff*)

lemma *real-eval-mult*: $\text{real-eval } (P * Q) \ x = \text{real-eval } P \ x * \text{real-eval } Q \ x$
by (*simp add: real-eval-def re-map-mult*)

lemma *real-eval-const*: $\text{real-eval } [:c:] \ x = \text{of-rat } c$
by (*simp add: real-eval-def map-poly-pCons*)

lemma *real-eval-zero*: $\text{real-eval } 0 \ x = 0$
by (*simp add: real-eval-def*)

lemma *real-eval-one*: $\text{real-eval } 1 \ x = 1$
by (*simp add: real-eval-def*)

lemma *real-eval-pX*: $\text{real-eval } pX \ x = x$
by (*simp add: real-eval-def map-poly-monom poly-monom*)

lemma *real-eval-monom*: $\text{real-eval } (\text{Polynomial.monom } c \ n) \ x = \text{of-rat } c * x ^ n$
by (*simp add: real-eval-def map-poly-monom poly-monom*)

lemma *real-eval-smult*: $\text{real-eval } (\text{Polynomial.smult } c \ P) \ x = \text{of-rat } c * \text{real-eval } P \ x$

proof –

have $e: \text{Polynomial.smult } c \ P = [:c:] * P$ **by** *simp*
show *?thesis unfolding e real-eval-mult real-eval-const ..*

qed

lemma *real-eval-sum*: $\text{real-eval } (\sum i \in I. f \ i) \ x = (\sum i \in I. \text{real-eval } (f \ i) \ x)$
by (*induction I rule: infinite-finite-induct*) (*simp-all add: real-eval-zero real-eval-add*)

lemma *real-eval-prod*: $\text{real-eval } (\prod i \in I. f \ i) \ x = (\prod i \in I. \text{real-eval } (f \ i) \ x)$

by (*induction I rule: infinite-finite-induct*) (*simp-all add: real-eval-one real-eval-mult*)

lemma *real-eval-power*: $\text{real-eval } (P \wedge n) x = \text{real-eval } P x \wedge n$

by (*induction n*) (*simp-all add: real-eval-one real-eval-mult*)

lemmas *real-eval-simps* = *real-eval-add real-eval-diff real-eval-mult real-eval-const*
real-eval-pX

real-eval-monom

real-eval-smult real-eval-sum real-eval-prod real-eval-power real-eval-zero real-eval-one

lemma *val2-aux*:

fixes $W Q P F a s B :: \text{real}$

assumes $W \neq 0 \ Q \neq 0 \ P \neq 0 \ F \neq 0$

shows $W / 12 * (a * F) / (W * (2 * Q * P)) * (s * Q * P * B / F) = s * B$
 $* a / 24$

using *assms* **by** (*simp add: field-simps*)

lemma *moment-term*:

$(2 * \pi)^4 / 12 * \text{real } (l + 1)^4 * (y^{(2 * e + 5)} * \exp(-((2 * \pi * \text{real } (l + 1))^4 * y))) =$

$(y^{(2 * e)} * ((2 * \pi)^4 * y^5 / 12)) * (\text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y))$

by (*simp add: power-add mult-ac*)

Moments: $\int_{-\infty}^{\infty} y^{2e} w(y) dy = \mu(t^e)$ (Euler's formula for $\zeta(2e+2)$:
Zeta-Function.zeta-even-nat; Gamma integrals: *Gamma-integral-real*).

lemma *moment*:

set-integrable lborel $\{0 < ..\} (\lambda y. y^{(2 * e)} * \text{weight } y) \wedge$

set-lebesgue-integral lborel $\{0 < ..\} (\lambda y. y^{(2 * e)} * \text{weight } y) = \text{of-rat } (\mu \text{Mono } e)$

proof –

define *g* **where** $g = (\lambda l y. (2 * \pi)^4 / 12 * \text{real } (l + 1)^4 * (y^{(2 * e + 5)} * \exp(-((2 * \pi * \text{real } (l + 1))^4 * y)))) :: \text{real}$

have *pt*: $\bigwedge y. 0 < y \implies y^{(2 * e)} * \text{weight } y = (\sum l. g l y)$

proof –

fix $y :: \text{real}$ **assume** $y > 0$

have *s*: *summable* $(\lambda l. \text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y))$

by (*rule weight-summable[OF y]*)

have $y^{(2 * e)} * \text{weight } y = (y^{(2 * e)} * ((2 * \pi)^4 * y^5 / 12)) * (\sum l. \text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y))$

unfolding *weight-eq* **by** (*simp only: mult.assoc*)

also have $\dots = (\sum l. (y^{(2 * e)} * ((2 * \pi)^4 * y^5 / 12)) * (\text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y)))$

by (*rule suminf-mult[OF s, symmetric]*)

also have $\dots = (\sum l. g l y)$ **unfolding** *g-def moment-term ..*

finally show $y^{(2 * e)} * \text{weight } y = (\sum l. g l y)$.


```

qed
have int:  $\bigwedge l. \text{set-integrable lborel } \{0 < ..\} (g \ l)$ 
  unfolding g-def by (intro set-integrable-mult-right gam-integrable) simp
have val:  $\bigwedge l. \text{set-lebesgue-integral lborel } \{0 < ..\} (g \ l) =$ 
   $((2 * \pi)^4 / 12 * \text{fact } (2 * e + 5) / (2 * \pi)^{(2 * e + 6)}) * (1 / \text{real}$ 
 $(l + 1)^{(2 * (e + 1))})$ 
proof -
  fix l :: nat
  define X where X = real (l + 1)
  have X0:  $0 < X$  by (simp add: X-def)
  have c:  $0 < 2 * \pi * X$  using X0 by simp
  have set-lebesgue-integral lborel  $\{0 < ..\} (g \ l) =$ 
 $(2 * \pi)^4 / 12 * X^4 * (\text{fact } (2 * e + 5) / (2 * \pi * X)^{(2 * e + 5 + 1)})$ 
  unfolding g-def X-def[symmetric] set-integral-mult-right gam[OF c] ..
  also have  $(2 * \pi * X)^{(2 * e + 5 + 1)} =$ 
 $(2 * \pi)^{(2 * e + 6)} * (X^4 * X^{(2 * (e + 1))})$ 
  proof -
    have e6:  $2 * e + 5 + 1 = 2 * e + 6$  by simp
    have  $2 * e + 5 + 1 = 4 + 2 * (e + 1)$  by simp
    hence  $X^{(2 * e + 5 + 1)} = X^4 * X^{(2 * (e + 1))}$  by (simp only:
power-add)
    moreover have  $(2 * \pi)^{(2 * e + 5 + 1)} = (2 * \pi)^{(2 * e + 6)}$ 
  unfolding e6 ..
  ultimately show ?thesis by (simp only: power-mult-distrib)
qed
also have  $(2 * \pi)^4 / 12 * X^4 * (\text{fact } (2 * e + 5) / ((2 * \pi)^{(2 * e + 6)} * (X^4 * X^{(2 * (e + 1))}))) =$ 
 $((2 * \pi)^4 / 12 * \text{fact } (2 * e + 5) / (2 * \pi)^{(2 * e + 6)}) * (1 / X^{(2 * (e + 1))})$ 
  using X0 by (simp add: field-simps)
  finally show set-lebesgue-integral lborel  $\{0 < ..\} (g \ l) =$ 
 $((2 * \pi)^4 / 12 * \text{fact } (2 * e + 5) / (2 * \pi)^{(2 * e + 6)}) * (1 / \text{real}$ 
 $(l + 1)^{(2 * (e + 1))})$ 
  unfolding X-def .
qed
have zeta:  $(\lambda l. 1 / \text{real } (l + 1)^{(2 * (e + 1))}) \text{ sums}$ 
 $((- 1)^{(e + 2)} * 2^{(2 * e + 1)} * \pi^{(2 * (e + 1))} * \text{of-rat } (\text{bernoulli}$ 
 $(2 * (e + 1))) /$ 
 $\text{fact } (2 * (e + 1)))$ 
proof -
  have n:  $1 < 2 * (e + 1)$  by simp
  have s0:  $(\lambda k. 1 / \text{of-nat } (\text{Suc } k)^{(2 * (e + 1))} :: \text{complex}) \text{ sums zeta } (\text{of-nat}$ 
 $(2 * (e + 1)))$ 
  by (rule sums-zeta-nat[OF n])

```

```

have z: zeta (of-nat (2 * (e + 1))) =
  complex-of-real ((-1) ^ Suc (e + 1) * bernoulli (2 * (e + 1)) * (2 * pi) ^
(2 * (e + 1)) /
  (2 * fact (2 * (e + 1))))
  using zeta-even-nat[of e + 1] by (simp only: of-nat-mult of-nat-numeral)
have f: (λk. 1 / of-nat (Suc k) ^ (2 * (e + 1)) :: complex) =
  (λk. complex-of-real (1 / real (k + 1) ^ (2 * (e + 1))))
  by (simp add: fun-eq-iff)
have s: (λl. 1 / real (l + 1) ^ (2 * (e + 1))) sums ((-1) ^ Suc (e + 1) *
  bernoulli (2 * (e + 1)) * (2 * pi) ^ (2 * (e + 1)) / (2 * fact (2 * (e + 1))))
  using s0 unfolding f z sums-of-real-iff .
have t2: (2::real) ^ (2 * (e + 1)) = 2 * 2 ^ (2 * e + 1)
proof -
  have 2 * (e + 1) = Suc (2 * e + 1) by simp
  thus ?thesis by (simp only: power-Suc)
qed
have m1: (-1::real) ^ Suc (e + 1) = (-1) ^ e by simp
have m2: (-1::real) ^ (e + 2) = (-1) ^ e by simp
have v: (-1) ^ Suc (e + 1) * bernoulli (2 * (e + 1)) * (2 * pi) ^ (2 * (e +
1)) / (2 *
  fact (2 * (e + 1))) =
  (-1) ^ (e + 2) * 2 ^ (2 * e + 1) * pi ^ (2 * (e + 1)) * of-rat (bernoulli
(2 * (e + 1))) /
  fact (2 * (e + 1)))
  unfolding of-rat-bernoulli power-mult-distrib[of 2 pi] t2 m1 m2 using field-simps
by argo
  show ?thesis using s unfolding v .
qed
have sums: (λl. set-lebesgue-integral lborel {0<..} (g l)) sums
  (((2 * pi) ^ 4 / 12 * fact (2 * e + 5) / (2 * pi) ^ (2 * e + 6)) *
  ((-1) ^ (e + 2) * 2 ^ (2 * e + 1) * pi ^ (2 * (e + 1)) * of-rat (bernoulli
(2 * (e + 1))) /
  fact (2 * (e + 1))))
  unfolding val by (rule sums-mult[OF zeta])
have ps: ∧y. 0 < y ⇒ summable (λl. g l y)
proof -
  fix y :: real assume y: 0 < y
  have summable (λl. (y ^ (2 * e) * ((2 * pi) ^ 4 * y ^ 5 / 12)) * (real (l + 1)
^ 4 *
  exp (- 2 * pi * real (l + 1) * y)))
  by (rule summable-mult[OF weight-summable[OF y]])
  thus summable (λl. g l y) unfolding g-def moment-term .
qed
have nn: ∧l y. 0 < y ⇒ 0 ≤ g l y by (simp add: g-def)
obtain I V where IV: set-integrable lborel {0<..} (λy. ∑ l. g l y)
  set-lebesgue-integral lborel {0<..} (λy. ∑ l. g l y) =
  (∑ l. set-lebesgue-integral lborel {0<..} (g l))
  using integral-suminf-nn[OF int nn ps sums-summable[OF sums]] by blast
have eq: set-lebesgue-integral lborel {0<..} (λy. y ^ (2 * e) * weight y) =

```

```

    set-lebesgue-integral lborel {0<..} (λy. ∑ l. g l y)
  by (rule set-integral-cong-Ioi) (rule pt)
have int2: set-integrable lborel {0<..} (λy. y ^ (2 * e) * weight y)
  by (rule set-integrable-cong-Ioi[OF IV(1)]) (rule pt[symmetric])
have val2: (∑ l. set-lebesgue-integral lborel {0<..} (g l)) = of-rat (muMono e)
proof -
  have F: fact (2 * e + 5) = (2 * real e + 5) * (2 * real e + 4) * (2 * real e
+ 3) * (fact (2 *
    (e + 1)) :: real)
  proof -
    have 2 * e + 5 = Suc (Suc (Suc (2 * (e + 1)))) by simp
    hence fact (2 * e + 5) = (real (Suc (Suc (Suc (2 * (e + 1)))))) * real (Suc
(Suc (2 * (e +
      1)))) * real (Suc (2 * (e + 1))) * (fact (2 * (e + 1)) :: real)
    by (simp only: fact-Suc of-nat-mult mult.assoc)
    thus ?thesis by (simp add: algebra-simps)
  qed
  have T: (2 * pi) ^ (2 * e + 6) = (2 * pi) ^ 4 * (2 * 2 ^ (2 * e + 1) * pi ^
(2 * (e + 1)))
  proof -
    have 2 * e + 6 = 4 + 2 * (e + 1) by simp
    hence (2 * pi) ^ (2 * e + 6) = (2 * pi) ^ 4 * (2 * pi) ^ (2 * (e + 1))
      by (simp only: power-add)
    moreover have (2::real) ^ (2 * (e + 1)) = 2 * 2 ^ (2 * e + 1)
    proof -
      have 2 * (e + 1) = Suc (2 * e + 1) by simp
      thus ?thesis by (simp only: power-Suc)
    qed
    ultimately show ?thesis by (simp only: power-mult-distrib mult.assoc)
  qed
  have m: (-1 :: real) ^ (e + 2) = (-1) ^ e by simp
  have nz1: (2 * pi) ^ 4 ≠ (0::real) by simp
  have nz2: (2::real) ^ (2 * e + 1) ≠ 0 by simp
  have nz3: pi ^ (2 * (e + 1)) ≠ (0::real) by simp
  have nz4: fact (2 * (e + 1)) ≠ (0::real) by simp
  have (∑ l. set-lebesgue-integral lborel {0<..} (g l)) =
    ((2 * pi) ^ 4 / 12 * fact (2 * e + 5) / (2 * pi) ^ (2 * e + 6)) *
    ((-1) ^ (e + 2) * 2 ^ (2 * e + 1) * pi ^ (2 * (e + 1)) * of-rat (bernoulli
(2 * (e + 1)))) /
    fact (2 * (e + 1)))
  by (rule sums-unique[OF sums, symmetric])
  also have ... = (-1) ^ e * of-rat (bernoulli (2 * (e + 1))) * ((2 * real e +
5) * (2 *
    real e + 4) * (2 * real e + 3)) / 24
  unfolding F T m by (rule val2-aux[OF nz1 nz2 nz3 nz4])
  also have ... = of-rat (muMono e)
  by (simp add: muMono-def of-rat-mult of-rat-divide of-rat-power of-rat-add
of-rat-minus
    mult-ac)

```

```

    finally show ?thesis .
qed
show ?thesis using int2 eq IV(2) val2 by simp
qed

Polynomial part:  $\int -0^\infty A(y^2) w(y) dy = \mu(A)$ .

lemma poly-moment:
  set-integrable lborel {0<..} ( $\lambda y. \text{real-eval } A (y^2) * \text{weight } y$ )  $\wedge$ 
  set-lebesgue-integral lborel {0<..} ( $\lambda y. \text{real-eval } A (y^2) * \text{weight } y$ ) =
    of-rat ( $\mu\text{Poly } A$ )
proof -
  define D where D = degree A
  have ev:  $\text{real-eval } A x = (\sum e \leq D. \text{of-rat } (\text{Polynomial.coeff } A e) * x^e)$  for x
  proof -
    have  $\text{real-eval } A x = \text{real-eval } (\sum e \leq D. \text{Polynomial.monom } (\text{Polynomial.coeff } A e) e) x$ 
    unfolding D-def by (simp only: poly-as-sum-of-monoms)
    thus ?thesis by (simp only: real-eval-sum real-eval-monom)
  qed
  have e: ( $\lambda y. \text{real-eval } A (y^2) * \text{weight } y$ ) = ( $\lambda y. \sum e \leq D. \text{of-rat } (\text{Polynomial.coeff } A e) * (y^{2 * e}) * \text{weight } y$ )
  proof (rule ext)
    fix y :: real
    show  $\text{real-eval } A (y^2) * \text{weight } y = (\sum e \leq D. \text{of-rat } (\text{Polynomial.coeff } A e) * (y^{2 * e}) * \text{weight } y)$ 
    unfolding ev sum-distrib-right power-mult[symmetric] mult.assoc ..
  qed
  have ij:  $\text{set-integrable lborel } \{0<..\} (\lambda y. \text{of-rat } (\text{Polynomial.coeff } A e) * (y^{2 * e}) * \text{weight } y)$  if  $e \in \{..D\}$  for e
    by (rule set-integrable-mult-right) (rule conjunct1[OF moment])
  have  $\text{set-lebesgue-integral lborel } \{0<..\} (\lambda y. \text{real-eval } A (y^2) * \text{weight } y) = \text{set-lebesgue-integral lborel } \{0<..\} (\lambda y. \sum e \leq D. \text{of-rat } (\text{Polynomial.coeff } A e) * (y^{2 * e}) * \text{weight } y)$ 
    by (simp only: e)
  also have ... =  $(\sum e \leq D. \text{set-lebesgue-integral lborel } \{0<..\} (\lambda y. \text{of-rat } (\text{Polynomial.coeff } A e) * (y^{2 * e}) * \text{weight } y)))$ 
    by (rule conjunct2[OF set-integral-sum-fin[OF ij]])
  also have ... =  $(\sum e \leq D. \text{of-rat } (\text{Polynomial.coeff } A e) * \text{of-rat } (\mu\text{Mono } e))$ 
    by (simp only: set-integral-mult-right conjunct2[OF moment])
  also have ... =  $\text{of-rat } (\mu\text{Poly } A)$  by (simp add: muPoly-def D-def of-rat-sum of-rat-mult)
  finally have v:  $\text{set-lebesgue-integral lborel } \{0<..\} (\lambda y. \text{real-eval } A (y^2) * \text{weight } y) = \text{of-rat } (\mu\text{Poly } A)$  .
  have i:  $\text{set-integrable lborel } \{0<..\} (\lambda y. \text{real-eval } A (y^2) * \text{weight } y)$ 
    unfolding e by (rule conjunct1[OF set-integral-sum-fin[OF ij]])
  show ?thesis using i v by (rule conjI)
qed

```

Pole part: $\int -0^\infty w(y)/(y^2 + j^2) dy = j^4(\zeta(5) - H-j) - 1/4 + 1/(2j),$

via the symmetry $c^4 \Phi(a, c) = a^4 \Phi(c, a)$ of $\Phi(a, c) = \int y^5 e^{-cy} / (y^2 + a^2) dy$ (Fubini) and *coth-series*.

$\zeta(5) = \sum_{n \geq 1} \frac{1}{n^5}$ as a real series, and its tails $\zeta(5) - H_j = \sum_{n \geq 0} \frac{1}{(j+n+1)^5}$.

lemma *zeta5-sums*: $(\lambda n. 1 / (\text{real } n + 1)^5) \text{ sums } \text{zeta5}$

proof –

have *s0*: $(\lambda k. 1 / \text{of-nat } (\text{Suc } k)^5 :: \text{complex}) \text{ sums } \text{zeta} (\text{of-nat } 5)$
by (*rule sums-zeta-nat*) *simp*
have *f*: $(\lambda k. 1 / \text{of-nat } (\text{Suc } k)^5 :: \text{complex}) =$
 $(\lambda k. \text{complex-of-real } (1 / (\text{real } k + 1)^5))$
by (*simp add: fun-eq-iff add.commute*)
have *zeta 5* $\in \mathbb{R}$ **using** *zeta-real[of 5]* **by** *simp*
hence *z*: $\text{zeta} (\text{of-nat } 5) = \text{complex-of-real } \text{zeta5}$ **by** (*simp add: zeta5-def*)
show *?thesis* **using** *s0 unfolding f z sums-of-real-iff* .

qed

lemma *H5-eq-sum*: $\text{of-rat } (H5 \ j) = (\sum_{i < j} 1 / (\text{real } i + 1)^5)$
unfolding *H5-def image-Suc-lessThan[symmetric] sum.reindex[OF inj-Suc]* *o-def of-rat-sum*

by (*simp add: of-rat-divide of-rat-power of-rat-add add.commute*)

lemma *zeta-tail*: $(\lambda n. 1 / (\text{real } j + (\text{real } n + 1))^5) \text{ sums } (\text{zeta5} - \text{of-rat } (H5 \ j))$

proof –

have $(\lambda i. 1 / (\text{real } (i + j) + 1)^5) \text{ sums } (\text{zeta5} - (\sum_{i < j} 1 / (\text{real } i + 1)^5))$
using *sums-split-initial-segment[OF zeta5-sums, of j]* **by** *simp*
thus *?thesis* **unfolding** *H5-eq-sum* **by** (*simp add: algebra-simps*)

qed

lemma *pole-val*:

$j \neq 0 \implies$
 $\text{set-integrable lborel } \{0 < ..\} (\lambda y. \text{weight } y / (y^2 + \text{real } j^2)) \wedge$
 $\text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda y. \text{weight } y / (y^2 + \text{real } j^2)) =$
 $\text{real-eval } (\text{muPole } j) \ \text{zeta5}$

proof –

assume *j*: $j \neq 0$
have *jpos*: $0 < \text{real } j$ **using** *j* **by** *simp*
have $\text{set-integrable lborel } \{0 < ..\} (\lambda y. \text{weight } y / (y^2 + \text{real } j^2)) \wedge$
 $\text{set-lebesgue-integral lborel } \{0 < ..\} (\lambda y. \text{weight } y / (y^2 + \text{real } j^2)) =$
 $1 / (2 * \text{real } j) - 1 / 4 + \text{real } j^4 * (\text{zeta5} - \text{of-rat } (H5 \ j))$
by (*rule pole[OF jpos zeta-tail]*)
moreover **have** $\text{real-eval } (\text{muPole } j) \ \text{zeta5} =$
 $\text{real } j^4 * (\text{zeta5} - \text{of-rat } (H5 \ j)) - 1 / 4 + 1 / (2 * \text{real } j)$
unfolding *muPole-def*
by (*simp add: real-eval-simps of-rat-power of-rat-divide of-rat-mult of-rat-diff algebra-simps*)
ultimately show *?thesis* **by** (*simp add: algebra-simps*)

qed

lemma *real-eval-sqPoleProd-pos*:

assumes *finite S* $0 \notin S$ $0 \leq t$

shows $0 < \text{real-eval } (\text{sqPoleProd } S) \ t$

proof –

have $0 < (\prod j \in S. \text{real-eval } (pX + [: \text{of-nat } j \wedge 2:])) \ t$

proof (*rule prod-pos*)

fix *j* **assume** *j*: $j \in S$

hence $j \neq 0$ **using** *assms(2)* **by** *metis*

thus $0 < \text{real-eval } (pX + [: \text{of-nat } j \wedge 2:])) \ t$

using *assms(3)* **by** (*simp add: real-eval-simps of-rat-power add-nonneg-pos*)

qed

thus *?thesis* **unfolding** *sqPoleProd-def real-eval-prod* .

qed

Partial fractions for $P(t)/\prod_{j \in S} (t + j^2)$ at real $t \geq 0$.

lemma *partial-fractions-real*:

assumes *finite S* $0 \notin S$ $0 \leq t$

shows $\text{real-eval } P \ t / \text{real-eval } (\text{sqPoleProd } S) \ t =$

$\text{real-eval } (P \text{ div } \text{sqPoleProd } S) \ t + (\sum j \in S. \text{of-rat } (\text{sqResidue } S \ P \ j) / (t + \text{real } j^2))$

proof –

note $S = \text{assms}(1)$

define *E* **where** $E \ T = \text{real-eval } (\text{sqPoleProd } T) \ t$ **for** *T*

define *Q* **where** $Q = P \text{ div } \text{sqPoleProd } S$

define *r* **where** $r \ j = \text{sqResidue } S \ P \ j$ **for** *j*

have *Epos*: $0 < E \ T$ **if** $T \subseteq S$ **for** *T*

unfolding *E-def* **using** *that* **by** (*intro real-eval-sqPoleProd-pos*) (*use assms finite-subset in auto*)

have *Erem*: $E \ S = (t + \text{real } j^2) * E \ (S - \{j\})$ **if** *j*: $j \in S$ **for** *j*

unfolding *E-def sqPoleProd-def real-eval-prod prod.remove[OF S j]*

by (*simp add: real-eval-simps of-rat-power*)

have $\text{real-eval } P \ t = \text{real-eval } (\text{sqPoleProd } S * (P \text{ div } \text{sqPoleProd } S) + (\sum j \in S. [: \text{sqResidue } S \ P \ j:] * \text{sqPoleProd } (S - \{j\}))) \ t$

by (*rule arg-cong[where f = $\lambda R. \text{real-eval } R \ t$, OF sq-lagrange[OF S]]*)

hence *L*: $\text{real-eval } P \ t = E \ S * \text{real-eval } Q \ t + (\sum j \in S. \text{of-rat } (r \ j) * E \ (S - \{j\}))$

by (*simp add: E-def Q-def r-def real-eval-simps*)

have $\text{real-eval } P \ t / E \ S = \text{real-eval } Q \ t + (\sum j \in S. \text{of-rat } (r \ j) * E \ (S - \{j\})) / E \ S$

unfolding *L* **using** *Epos[of S]* **by** (*simp add: add-divide-distrib sum-divide-distrib*)

also have $(\sum j \in S. \text{of-rat } (r \ j) * E \ (S - \{j\})) / E \ S = (\sum j \in S. \text{of-rat } (r \ j) / (t + \text{real } j^2))$

proof (*rule sum.cong[OF refl]*)

fix *j* **assume** *j*: $j \in S$

have $0 < E \ (S - \{j\})$ **by** (*rule Epos*) *auto*

hence *E0*: $E \ (S - \{j\}) \neq 0$ **by** *simp*

have $\text{of-rat } (r \ j) * E \ (S - \{j\}) / ((t + \text{real } j^2) * E \ (S - \{j\})) =$

$of-rat (r j) / (t + real j \wedge 2)$
by (rule nonzero-mult-divide-mult-cancel-right[OF E0])
thus $of-rat (r j) * E (S - \{j\}) / E S = of-rat (r j) / (t + real j \wedge 2)$
unfolding Erem[OF j] .
qed
finally show ?thesis **unfolding** E-def Q-def r-def .
qed

lemma rat-decomp:

assumes finite S 0 $\notin$ S
shows $real-eval P (y \wedge 2) / real-eval (sqPoleProd S) (y \wedge 2) * weight y =$
 $real-eval (P div sqPoleProd S) (y \wedge 2) * weight y + (\sum j \in$
 $S. of-rat (sqResidue S P j) * (weight y / (y \wedge 2 + real j \wedge 2)))$
unfolding partial-fractions-real[OF assms zero-le-power2]
by (simp add: distrib-right sum-distrib-right)

lemma rat-integrable:

assumes finite S 0 $\notin$ S
shows set-integrable lborel {0<..} $(\lambda y. real-eval P (y \wedge 2) /$
 $real-eval (sqPoleProd S) (y \wedge 2) * weight y)$
proof –
have e: $(\lambda y. real-eval P (y \wedge 2) / real-eval (sqPoleProd S) (y \wedge 2) * weight y) =$
 $(\lambda y. real-eval (P div sqPoleProd S) (y \wedge 2) * weight y + (\sum j \in$
 $S. of-rat (sqResidue S P j) * (weight y / (y \wedge 2 + real j \wedge 2))))$
by (rule ext) (rule rat-decomp[OF assms])
have j0: $j \neq 0$ **if** $j \in S$ **for** j **using** assms(2) **that by** metis
have ij: set-integrable lborel {0<..} $(\lambda y. of-rat (sqResidue S P j) * (weight y /$
 $(y \wedge 2 + real j \wedge 2)))$ **if** $j \in S$ **for** j
by (rule set-integrable-mult-right) (rule conjunct1[OF pole-val[OF j0[OF that]]])
show ?thesis **unfolding** e
by (rule set-integral-add(1)[OF conjunct1[OF poly-moment] conjunct1[OF
set-integral-sum-fin[OF
ij]]])
qed

Proposition 2.2, (2.10): $\mu_{\zeta(5)}(R) = \int_{-\infty}^{\infty} R(y^2) w(y) dy$ for every R in the domain.

theorem muX-zeta5-integral:

assumes S: finite S **and** S0: 0 $\notin$ S
shows $real-eval (muX S P) zeta5 =$
 $set-lebesgue-integral lborel \{0<..\}$
 $(\lambda y. real-eval P (y \wedge 2) / real-eval (sqPoleProd S) (y \wedge 2) * weight y)$
proof –
have j0: $\bigwedge j. j \in S \implies j \neq 0$ **using** S0 **by** metis
have set-lebesgue-integral lborel {0<..} $(\lambda y. real-eval P (y \wedge 2) /$
 $real-eval (sqPoleProd S) (y \wedge 2) * weight y) =$
 $set-lebesgue-integral lborel \{0<..\} (\lambda y. real-eval (P div sqPoleProd S) (y \wedge$
 $2) * weight y)$

```

+ (∑ j ∈ S. of-rat (sqResidue S P j) *
  set-lebesgue-integral lborel {0<..} (λy. weight y / (y ^ 2 + real j ^ 2)))
proof -
  have e: (λy. real-eval P (y ^ 2) / real-eval (sqPoleProd S) (y ^ 2) * weight y)
=
  (λy. real-eval (P div sqPoleProd S) (y ^ 2) * weight y + (∑ j ∈
    S. of-rat (sqResidue S P j) * (weight y / (y ^ 2 + real j ^ 2))))
  by (rule ext) (rule rat-decomp[OF S S0])
  have ij: set-integrable lborel {0<..} (λy. of-rat (sqResidue S P j) * (weight y /
    (y ^ 2 + real j ^ 2))) if j ∈ S for j
    by (rule set-integrable-mult-right) (rule conjunct1[OF pole-val[OF j0[OF
that]]])
  have set-lebesgue-integral lborel {0<..} (λy. real-eval P (y ^ 2) /
    real-eval (sqPoleProd S) (y ^ 2) * weight y) =
    set-lebesgue-integral lborel {0<..} (λy. real-eval (P div sqPoleProd S) (y ^
2) *
      weight y + (∑ j ∈ S. of-rat (sqResidue S P j) * (weight y / (y ^ 2 +
        real j ^ 2))))
    by (simp only: e)
  also have ... = set-lebesgue-integral lborel {0<..} (λy. real-eval (P div
    sqPoleProd S) (y ^ 2) * weight y) +
    set-lebesgue-integral lborel {0<..} (λy. ∑ j ∈
      S. of-rat (sqResidue S P j) * (weight y / (y ^ 2 + real j ^ 2)))
    by (rule set-integral-add(2)[OF conjunct1[OF poly-moment] conjunct1[OF
set-integral-sum-fin[OF
ij]]])
  also have set-lebesgue-integral lborel {0<..} (λy. ∑ j ∈
    S. of-rat (sqResidue S P j) * (weight y / (y ^ 2 + real j ^ 2))) =
    (∑ j ∈ S. set-lebesgue-integral lborel {0<..} (λy. of-rat (sqResidue S P j)
      * (weight y / (y ^ 2 + real j ^ 2))))
    by (rule conjunct2[OF set-integral-sum-fin[OF ij]])
  also have ... = (∑ j ∈
    S. of-rat (sqResidue S P j) * set-lebesgue-integral lborel {0<..} (λy. weight y
/
  (y ^ 2 + real j ^ 2)))
    by (simp only: set-integral-mult-right)
  finally show ?thesis .
qed
also have ... = of-rat (muPoly (P div sqPoleProd S)) + (∑ j ∈
  S. of-rat (sqResidue S P j) * real-eval (muPole j) zeta5)
  using poly-moment pole-val[OF j0] by (simp cong: sum.cong)
also have ... = real-eval (muX S P) zeta5
  by (simp add: muX-def real-eval-simps)
finally show ?thesis ..
qed

```

Proposition 2.2: $G\text{-}K(\zeta(5))$ is positive definite (stated with coefficient vectors).

theorem $G\text{-}zeta5\text{-posdef}$:


```

assumes  $c: \exists i < \text{hof } K. c \ i \neq 0$ 
shows  $0 < (\sum i < \text{hof } K. \sum j < \text{hof } K. c \ i * \text{real-eval } (GK \ K \ \$\$ \ (i, j)) \ \text{zeta5}$ 
 $* \ c \ j)$ 
proof –
  define  $h$  where  $h = \text{hof } K$ 
  define  $f$  where  $f = (\lambda n \ y. \text{real-eval } (Dm \ (\text{Nof } K) \wedge 6 * pX \wedge n) \ (y \wedge 2) /$ 
 $\text{real-eval } (sqPoleProd \ \{1..K\}) \ (y \wedge 2) * \text{weight } y :: \text{real})$ 
  define  $q$  where  $q = (\lambda y. \sum i < h. c \ i * y \wedge (2 * i) :: \text{real})$ 
  have  $S0: 0 \notin \{1..K\}$  by simp
  have entry:  $\bigwedge i \ j. i < h \implies j < h \implies$ 
 $\text{real-eval } (GK \ K \ \$\$ \ (i, j)) \ \text{zeta5} = \text{set-lebesgue-integral lborel } \{0<..\} \ (f \ (i +$ 
 $j))$ 
  proof –
    fix  $i \ j$  assume  $i < h \ j < h$ 
    hence  $GK \ K \ \$\$ \ (i, j) = \mu X \ \{1..K\} \ (Dm \ (\text{Nof } K) \wedge 6 * pX \wedge (i + j))$  by
    (simp add: GK-def h-def)
    thus  $\text{real-eval } (GK \ K \ \$\$ \ (i, j)) \ \text{zeta5} = \text{set-lebesgue-integral lborel } \{0<..\} \ (f$ 
     $(i + j))$ 
    unfolding f-def by (simp only: muX-zeta5-integral[OF finite-atLeastAtMost
     $S0]$ )
  qed
  have fn:  $\bigwedge n \ y. f \ n \ y = (y \wedge 2) \wedge n * f \ 0 \ y$ 
  proof –
    fix  $n :: \text{nat}$  and  $y :: \text{real}$ 
    have a:  $\text{real-eval } (Dm \ (\text{Nof } K) \wedge 6 * pX \wedge n) \ (y \wedge 2) =$ 
 $\text{real-eval } (Dm \ (\text{Nof } K) \wedge 6) \ (y \wedge 2) * (y \wedge 2) \wedge n$ 
    by (simp only: real-eval-mult real-eval-power real-eval-pX)
    have b:  $\text{real-eval } (Dm \ (\text{Nof } K) \wedge 6 * pX \wedge 0) \ (y \wedge 2) = \text{real-eval } (Dm \ (\text{Nof}$ 
     $K) \wedge 6) \ (y \wedge 2)$ 
    by (simp only: power-0 mult-1-right)
    show  $f \ n \ y = (y \wedge 2) \wedge n * f \ 0 \ y$  unfolding f-def a b by (simp add: mult-ac)
  qed
  have f0pos:  $\bigwedge y. 0 < y \implies 0 < f \ 0 \ y$ 
  proof –
    fix  $y :: \text{real}$  assume  $y: 0 < y$ 
    have D:  $Dm \ (\text{Nof } K) = sqPoleProd \ \{1..\text{Nof } K\}$  by (simp add: Dm-def
     $sqPoleProd-def$ )
    have  $0 < \text{real-eval } (Dm \ (\text{Nof } K)) \ (y \wedge 2)$  unfolding D by (rule real-eval-sqPoleProd-pos)
    auto
    moreover have  $0 < \text{real-eval } (sqPoleProd \ \{1..K\}) \ (y \wedge 2)$ 
    by (rule real-eval-sqPoleProd-pos) auto
    ultimately show  $0 < f \ 0 \ y$  unfolding f-def using weight-pos[OF y]
    by (simp add: real-eval-mult real-eval-power real-eval-one)
  qed
  have int:  $\bigwedge n. \text{set-integrable lborel } \{0<..\} \ (f \ n)$  unfolding f-def
  by (rule rat-integrable[OF finite-atLeastAtMost S0])
  have pt:  $\bigwedge y. (\sum i < h. \sum j < h. c \ i * c \ j * f \ (i + j) \ y) = q \ y \wedge 2 * f \ 0 \ y$ 
  proof –
    fix  $y :: \text{real}$ 

```

have $s: c\ i * c\ j * f\ (i + j)\ y = (c\ i * y^{\wedge}(2 * i)) * (c\ j * y^{\wedge}(2 * j)) * f\ 0\ y$
for $i\ j$
proof –
have $p: (y^{\wedge} 2)^{\wedge}(i + j) = y^{\wedge}(2 * i) * y^{\wedge}(2 * j)$ **by** (*simp only: power-add power-mult*)
show ?thesis **unfolding** $fn[of\ i + j\ y]\ p$ **by** (*simp add: mult-ac*)
qed
have $q\ y^{\wedge} 2 = (\sum\ i < h. \sum\ j < h. (c\ i * y^{\wedge}(2 * i)) * (c\ j * y^{\wedge}(2 * j)))$
unfolding $q\text{-def}\ power2\text{-eq-square}$ **by** (*rule sum-product*)
hence $q\ y^{\wedge} 2 * f\ 0\ y = (\sum\ i < h. \sum\ j < h. (c\ i * y^{\wedge}(2 * i)) * (c\ j * y^{\wedge}(2 * j)) * f\ 0\ y)$
by (*simp only: sum-distrib-right*)
also have $\dots = (\sum\ i < h. \sum\ j < h. c\ i * c\ j * f\ (i + j)\ y)$ **by** (*simp only: s*)
finally show $(\sum\ i < h. \sum\ j < h. c\ i * c\ j * f\ (i + j)\ y) = q\ y^{\wedge} 2 * f\ 0\ y$..
qed
have $quad: (\sum\ i < h. \sum\ j < h. c\ i * real\text{-eval}\ (GK\ K\ \$\$ (i, j))\ zeta5 * c\ j) = set\text{-lebesgue-integral}\ lborel\ \{0 < ..\} (\lambda y. q\ y^{\wedge} 2 * f\ 0\ y)$
proof –
have $ii: set\text{-integrable}\ lborel\ \{0 < ..\} (\lambda y. c\ i * c\ j * f\ (i + j)\ y)$ **for** $i\ j$
by (*rule set-integrable-mult-right*) (*rule int*)
have $(\sum\ i < h. \sum\ j < h. c\ i * real\text{-eval}\ (GK\ K\ \$\$ (i, j))\ zeta5 * c\ j) = (\sum\ i < h. \sum\ j < h. set\text{-lebesgue-integral}\ lborel\ \{0 < ..\} (\lambda y. c\ i * c\ j * f\ (i + j)\ y))$
by (*intro sum.cong refl*) (*simp add: entry set-integral-mult-right mult-ac*)
also have $\dots = (\sum\ i < h. set\text{-lebesgue-integral}\ lborel\ \{0 < ..\} (\lambda y. \sum\ j < h. c\ i * c\ j * f\ (i + j)\ y))$
by (*intro sum.cong refl*) (*rule conjunct2[OF set-integral-sum-fin[OF ii], symmetric]*)
also have $\dots = set\text{-lebesgue-integral}\ lborel\ \{0 < ..\} (\lambda y. \sum\ i < h. \sum\ j < h. c\ i * c\ j * f\ (i + j)\ y)$
by (*rule conjunct2[OF set-integral-sum-fin, symmetric]*) (*rule conjunct1[OF set-integral-sum-fin[OF ii]]*)
also have $\dots = set\text{-lebesgue-integral}\ lborel\ \{0 < ..\} (\lambda y. q\ y^{\wedge} 2 * f\ 0\ y)$
by (*simp only: pt*)
finally show ?thesis .
qed
have $Fint: set\text{-integrable}\ lborel\ \{0 < ..\} (\lambda y. q\ y^{\wedge} 2 * f\ 0\ y)$
proof –
have $ii: set\text{-integrable}\ lborel\ \{0 < ..\} (\lambda y. c\ i * c\ j * f\ (i + j)\ y)$ **for** $i\ j$
by (*rule set-integrable-mult-right*) (*rule int*)
have $set\text{-integrable}\ lborel\ \{0 < ..\} (\lambda y. \sum\ i < h. \sum\ j < h. c\ i * c\ j * f\ (i + j)\ y)$
by (*rule conjunct1[OF set-integral-sum-fin]*) (*rule conjunct1[OF set-integral-sum-fin[OF ii]]*)
thus ?thesis **by** (*simp only: pt*)
qed
have $qne: finite\ \{y. q\ y = 0\}$
proof –
define Q **where** $Q = (\sum\ i < h. Polynomial.monom\ (c\ i)\ (2 * i))$

```

have pq: poly Q y = q y for y by (simp add: Q-def q-def poly-sum poly-monom)
obtain i0 where i0: i0 < h c i0 ≠ 0 using c unfolding h-def by blast
have Polynomial.coeff Q (2 * i0) = c i0 using i0(1) by (simp add: Q-def
coeff-sum coeff-monom)
hence Q ≠ 0 using i0(2) by auto
hence finite {y. poly Q y = 0} by (rule poly-roots-finite)
thus ?thesis unfolding pq .
qed
have 0 < set-lebesgue-integral lborel {0<..} (λy. q y ^ 2 * f 0 y)
proof -
  define G where G y = indicator {0<..} y *R (q y ^ 2 * f 0 y) for y :: real
  have Gi: integrable lborel G using Fint unfolding set-integrable-def G-def .
  have Gnn: 0 ≤ G y for y using f0pos[of y] by (cases 0 < y) (simp-all add:
G-def)
  have Gpos: 0 < G y if 0 < y q y ≠ 0 for y using that f0pos[of y]
  by (simp add: G-def)
  have 0 ≤ integralL lborel G by (rule integral-nonneg-AE) (simp add: Gnn)
  moreover have integralL lborel G ≠ 0
  proof
    assume integralL lborel G = 0
    hence AE x in lborel. G x = 0 using integral-nonneg-eq-0-iff-AE[OF Gi] Gnn
  by blast
  then obtain N where N: {x ∈ space lborel. G x ≠ 0} ⊆
    N emeasure lborel N = 0 N ∈ sets lborel by (rule AE-E)
  have Nn: N ∈ null-sets lborel using N(2,3) by blast
  have Zn: {y. q y = 0} ∈ null-sets lborel by (rule finite-imp-null-set-lborel[OF
qne])
  have sub: {1..2::real} ⊆ N ∪ {y. q y = 0}
  proof
    fix y :: real assume y: y ∈ {1..2}
    show y ∈ N ∪ {y. q y = 0}
    proof (cases q y = 0)
      case False
      hence G y ≠ 0 using Gpos[of y] y by force
      thus ?thesis using N(1) by auto
    qed simp
  qed
  have {1..2::real} ∈ null-sets lborel
  by (rule null-sets-subset[OF null-sets.Un[OF Nn Zn] - sub]) simp
  thus False by (simp add: null-sets-def)
qed
ultimately show ?thesis unfolding set-lebesgue-integral-def G-def by linarith
qed
thus ?thesis unfolding quad[symmetric] h-def .
qed

```

Consequence: $\Delta\text{-}K(\zeta(5)) > 0$ (positive definite $\Rightarrow$ positive determinant; alternatively from Andreief with a positive integrand).

lemma *real-eval-hom: comm-ring-hom* ($\lambda P. \text{real-eval } P \ x$)

by *unfold-locales* (*simp-all add: real-eval-def re-map-add re-map-mult*)

theorem *Delta-zeta5-pos-aux*: $0 < \text{real-eval } (\text{DeltaK } K) \text{ zeta5}$

proof –

define *G* **where** $G = \text{map-mat } (\lambda P. \text{real-eval } P \text{ zeta5}) (GK \ K)$

have *Gc*: $G \in \text{carrier-mat } (\text{hof } K) (\text{hof } K)$ **by** (*simp add: G-def GK-def*)

have $\text{real-eval } (\text{DeltaK } K) \text{ zeta5} = \text{Determinant.det } G$

unfolding *DeltaK-def G-def* **by** (*rule comm-ring-hom.hom-det[OF real-eval-hom, symmetric]*)

also have $0 < \text{Determinant.det } G$

proof (*rule posdef-det-pos[OF Gc]*)

fix $c :: \text{nat} \Rightarrow \text{real}$ **assume** $c: \exists i < \text{hof } K. c \ i \neq 0$

have $(\sum i < \text{hof } K. \sum j < \text{hof } K. c \ i * G \ \$\$ (i, j) * c \ j)$

$= (\sum i < \text{hof } K. \sum j < \text{hof } K. c \ i * \text{real-eval } (GK \ K \ \$\$ (i, j)) \text{ zeta5} * c \ j)$

by (*intro sum.cong refl*) (*simp add: G-def GK-def*)

also have $0 < \dots$ **by** (*rule G-zeta5-posdef[OF c]*)

finally show $0 < (\sum i < \text{hof } K. \sum j < \text{hof } K. c \ i * G \ \$\$ (i, j) * c \ j)$.

qed

finally show *?thesis* .

qed

theorem *Delta-zeta5-pos*: $0 < \text{real-eval } (\text{DeltaK } K) \text{ zeta5}$

by (*rule Delta-zeta5-pos-aux*)

end

theory *Zeta5-Exact-Constants*

imports *Zeta5-Definitions*

begin

24 Exact rational constants

Appendix B of the paper: the exact rational values of A^* , A_M and the margins in (7.2), by rational arithmetic (Lean theory *AppendixB*).

lemma *Hc-eq*: $Hc-q = 23 / 20$ **by** (*simp add: Hc-q-def alpha-q-def*)

The cancellation of the quadratic coefficient of R (Appendix B.1).

lemma *quadratic-cancellation*:

$2 * \text{lam-q} - 12 * \text{lam-q} * \text{alpha-q} + 2 * (Hc-q \wedge 2 - Hc-q - \text{lam-q} \wedge 2) - 18$

$* \text{alpha-q} \wedge 2 + 6 *$

$\text{alpha-q} = 0$ **by** (*simp add: lam-q-def alpha-q-def Hc-q-def power2-eq-square*)

lemma *lam-M0*: $\text{lam-q} * M0c = -49173 / 8000$ **by** (*simp add: lam-q-def M0c-def*)

(5.19): $A^* = I\text{-out} + \int_3 \{20\} R/x^3 - 2689/48000$.

lemma *Astar-eq*: $Astar = Iout + 322437603634266857629 / 7535670527041937280000$

– $2689 / 48000$

```

    by (simp add: Astar-def Iout-def)

lemma A200-eq: AM 200 = 127125602969131786927559 / 94195881588024216000000
  by (simp add: AM-def Astar-def lam-q-def)

lemma A100000-eq: AM 100000 = 7756864096839411316755964319057 / 5887242599251513500000000000000
  by (simp add: AM-def Astar-def lam-q-def)

(7.2),  $M = 200$ :  $-1600(A_2-0-0 + Ubar) > 139/5$ , with the exact margin.

lemma margin-200:
  - 1600 * (AM 200 + Ubar) - 139 / 5 = 3089837638249482469 / 58872425992515135000
  ^
  139 / 5 < - 1600 * (AM 200 + Ubar) by (simp add: AM-def Astar-def lam-q-def
  Ubar-def)

(7.2),  $M = 100000$ :  $-1600(A-1-0-0-0-0 + Ubar) > 7907/100$ .

lemma margin-100000:
  - 1600 * (AM 100000 + Ubar) - 7907 / 100 =
    29873543950273155160680943 / 3679526624532195937500000000 ^
    7907 / 100 < - 1600 * (AM 100000 + Ubar) by (simp add: AM-def Astar-def
    lam-q-def Ubar-def)

(6.4):  $Ubar < -136699/100000$ .

lemma U-lt: Ubar < - 136699 / 100000 by (simp add: Ubar-def)

end

theory Zeta5-Uniform-Asymptotics
  imports Zeta5-Inner-Range Piecewise-Interval-Integrals
begin

```

25 The uniform estimate (5.7) in the inner range

For fixed $M \geq 40$, $\gamma_p^{in} = p \Gamma(K/p) + O_M(1)$ and $v_p(S_K) = p N(K/p) + O_M(1)$ uniformly over the range (4.1), with the explicit constant $1000 M^2$ (*gammaIn-uniform-estimate*). Floors and $v_p(S_K)$, a Riemann-sum identity for the z -integral, exact class counts and the discrete decomposition of γ_p^{in} ; everything else is integer arithmetic. Lean theory *Uniform57*.

25.1 Floors and $v_p(S-K)$ in the inner range

```

lemma floor-nat-div: floor (real n / real d) = int (n div d)
  by (rule floor-divide-of-nat-eq)

```

Legendre with a single term: $v_p(n!) = \lfloor n/p \rfloor$ for $n < p^2$.

```

lemma multiplicity-fact-small:
  assumes prime p n < p ^ 2

```

shows $\text{multiplicity } p \text{ (fact } n) = n \text{ div } p$
proof –
have L : $\text{multiplicity } p \text{ (fact } n) = (\sum a = 1..n + 1. n \text{ div } p \wedge a)$
by (rule *legendre-sum*[*OF* *assms*(1)]) *simp*
have $n \text{ div } p \wedge a = 0$ **if** $a \in \{1..n + 1\}$ $a \neq 1$ **for** a
proof –
from *that* **have** $2 \leq a$ **by** *auto*
hence $p \wedge 2 \leq p \wedge a$ **by** (rule *power-increasing*) (use *prime-ge-1-nat*[*OF* *assms*(1)]) *in simp*
thus ?thesis **using** *assms*(2) **by** (intro *div-less*) *linarith*
qed
hence $(\sum a = 1..n + 1. n \text{ div } p \wedge a) = n \text{ div } p$ **by** (*subst sum.remove*[*of* - 1])
auto
thus ?thesis **using** L **by** *simp*
qed

Under (4.1) all factorial arguments are $< p^2$ (*innerHyp-bounds*: $5K < p^2$).

lemma *vpS-inner*:

assumes p : *prime* p **and** h : *InnerHyp* K M p
shows $\text{vpS } K \text{ } p = 2 * \text{int } (\text{hof } K) * \text{int } (K \text{ div } p) - 12 * \text{int } (\text{hof } K) * \text{int } (\text{Nof } K \text{ div } p)$

$$- 2 * \text{int } (\sum i = 1..\text{hof } K - 1. 2 * i \text{ div } p)$$

proof –
have b : $5 * K < p^2$ **using** *innerHyp-bounds*[*OF* h] **by** *simp*
have $p7$: $7 \leq p$ **using** *hyp-facts*[*OF* p h] **by** *simp*
have $m4$: $\text{multiplicity } p \text{ (} 4::\text{nat) } = 0$
proof –
have $\neg p \text{ dvd } 4$
proof
assume $p \text{ dvd } 4$
hence $p \leq 4$ **by** (rule *dvd-imp-le*) *simp*
thus *False* **using** $p7$ **by** *linarith*
qed
thus ?thesis **by** (rule *not-dvd-imp-multiplicity-0*)
qed
define $A :: \text{nat}$ **where** $A = \text{fact } K \wedge (2 * \text{hof } K) * 4 \wedge (\text{hof } K - 1)$
define $D :: \text{nat}$ **where** $D = \text{fact } (\text{Nof } K) \wedge (12 * \text{hof } K) * (\prod i = 1..\text{hof } K - 1. \text{fact } (2 * i) \wedge 2)$
have SK : $SK \text{ } K = \text{of-nat } A / \text{of-nat } D$ **by** (*simp add*: *SK-def* *A-def* *D-def*)
have mA : $\text{multiplicity } p \text{ } A = 2 * \text{hof } K * (K \text{ div } p)$
proof –
have pe : *prime-elem* p **using** p **by** *simp*
have Kp : $K < p^2$ **using** b **by** *linarith*
have $\text{multiplicity } p \text{ } A = 2 * \text{hof } K * \text{multiplicity } p \text{ (fact } K) + (\text{hof } K - 1) * \text{multiplicity } p \text{ (} 4::\text{nat) }$
unfolding *A-def* **using** pe **by** (*simp add*: *prime-elem-multiplicity-mult-distrib* *prime-elem-multiplicity-power-distrib*)
thus ?thesis **unfolding** *multiplicity-fact-small*[*OF* p Kp] $m4$ **by** *presburger*

```

qed
have mD: multiplicity p D = 12 * hof K * (Nof K div p) + 2 * (∑ i =
  1..hof K - 1. 2 * i div p)
proof -
  have pe: prime-elem p using p by simp
  have hK: hof K ≤ K unfolding hof-def by linarith
  have NK: Nof K < p ^ 2 using b unfolding Nof-def by linarith
  have fi: multiplicity p (fact (2 * i) ^ 2) = 2 * (2 * i div p) if i ∈ {1..hof K
- 1}
  for i
  proof -
    have i: 1 ≤ i i ≤ hof K - 1 using that by auto
    have ip: 2 * i < p ^ 2 using hK i b by arith
    show ?thesis using pe by (force simp: prime-elem-multiplicity-power-distrib
multiplicity-fact-small[OF p ip])
  qed
have P: multiplicity p (∏ i = 1..hof K - 1. fact (2 * i) ^ 2) = (∑ i =
  1..hof K - 1. 2 * (2 * i div p))
proof -
  have multiplicity p (∏ i = 1..hof K - 1. fact (2 * i) ^ 2) = (∑ i =
  1..hof K - 1. multiplicity p (fact (2 * i) ^ 2))
  by (rule prime-elem-multiplicity-prod-distrib[OF pe]) auto
  also have ... = (∑ i = 1..hof K - 1. 2 * (2 * i div p))
  by (rule sum.cong[OF refl]) (rule fi)
  finally show ?thesis .
qed
have multiplicity p D = 12 * hof K * multiplicity p (fact (Nof K)) +
  multiplicity p (∏ i = 1..hof K - 1. fact (2 * i) ^ 2)
  unfolding D-def using pe by (subst prime-elem-multiplicity-mult-distrib)
(simp-all add:
  prime-elem-multiplicity-power-distrib)
thus ?thesis unfolding P multiplicity-fact-small[OF p NK] by (metis sum-distrib-left)
qed
have pei: prime-elem (int p) using p by (simp add: prime-nat-int-transfer)
have A0: A ≠ 0 and D0: D ≠ 0 by (simp-all add: A-def D-def)
have v: vpS K p = int (multiplicity p A) - int (multiplicity p D)
  unfolding vpS-def vp-def SK using A0 D0 pei
  by (simp add: qmultiplicity-divide qmultiplicity-of-nat multiplicity-int)
show ?thesis unfolding v mA mD by auto
qed

```

$$G_4(n) = 4 \, t \, n - p \, t \, (t + 1), \, t = \lfloor 2n/p \rfloor; \, G_4(n)/4 = p \, J(n/p).$$

definition $G_4 :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{int}$ **where**

$$G_4 \, p \, n = 4 * \text{int} \, (2 * n \, \text{div} \, p) * \text{int} \, n - \text{int} \, p * \text{int} \, (2 * n \, \text{div} \, p) * (\text{int} \, (2 * n \, \text{div} \, p) + 1)$$

lemma $G_4\text{-step}$:

assumes $2 \leq p$

shows $0 \leq G_4 \, p \, (n + 1) - G_4 \, p \, n - 4 * \text{int} \, (2 * n \, \text{div} \, p) \wedge$

$$G4\ p\ (n + 1) - G4\ p\ n - 4 * \text{int}\ (2 * n \text{ div } p) \leq 4 * (\text{int}\ (2 * (n + 1) \text{ div } p) - \text{int}\ (2 * n \text{ div } p))$$

```

proof -
  define t where t = 2 * n div p
  define t' where t' = 2 * (n + 1) div p
  have tt: t ≤ t' t' ≤ t + 1
  proof -
    show t ≤ t' unfolding t-def t'-def by (rule div-le-mono) simp
    have 2 * (n + 1) ≤ 2 * n + p using assms by simp
    hence t' ≤ (2 * n + p) div p unfolding t'-def by (rule div-le-mono)
    also have (2 * n + p) div p = t + 1 unfolding t-def using assms by force
    finally show t' ≤ t + 1 .
  qed
  have bounds: t * p ≤ 2 * n 2 * n < (t + 1) * p t' * p ≤ 2 * (n + 1) 2 * (n + 1) <
  1) <
    (t' + 1) * p
  proof -
    have p0: 0 < p using assms by simp
    have g: x div p * p ≤ x x < (x div p + 1) * p for x
    proof -
      show x div p * p ≤ x by fastforce
      show x < (x div p + 1) * p
    proof -
      have e: (x div p + 1) * p = x div p * p + p by (simp add: algebra-simps)
      show ?thesis unfolding e using mod-less-divisor[OF p0, of x] div-mult-mod-eq[of
x p]
        by linarith
    qed
  qed
  show t * p ≤ 2 * n 2 * n < (t + 1) * p t' * p ≤ 2 * (n + 1) 2 * (n + 1) <
  (t' + 1) * p
  unfolding t-def t'-def by (rule g)+
qed
show ?thesis
proof (cases t' = t)
  case True
    have e: G4 p (n + 1) - G4 p n - 4 * int (2 * n div p) = 0
    unfolding G4-def t-def[symmetric] t'-def[symmetric] True by (fastforce simp:
algebra-simps)
    show ?thesis using True unfolding e unfolding t-def[symmetric] t'-def[symmetric]
by simp
  next
  case False
    hence t1: t' = t + 1 using tt by linarith
    have e: G4 p (n + 1) - G4 p n - 4 * int (2 * n div p) = 2 * (2 * int n + 2
- int p * int t')
    unfolding G4-def t-def[symmetric] t'-def[symmetric] t1 by (simp add: alge-
bra-simps)
    have b1: int t' * int p ≤ 2 * int n + 2

```



```

proof -
  have  $\text{int } (t' * p) \leq \text{int } (2 * (n + 1))$  using  $\text{bounds}(3)$  by (simp only:
of-nat-le-iff)
  thus ?thesis by auto
qed
have  $b2: 2 * \text{int } n < (\text{int } t + 1) * \text{int } p$ 
proof -
  have  $\text{int } (2 * n) < \text{int } ((t + 1) * p)$  using  $\text{bounds}(2)$  by (simp only:
of-nat-less-iff)
  thus ?thesis by (auto simp: algebra-simps)
qed
have  $b3: \text{int } t' = \text{int } t + 1$  using  $t1$  by simp
have  $e1: \text{int } t' * \text{int } p = \text{int } p * \text{int } t + \text{int } p (\text{int } t + 1) * \text{int } p = \text{int } p * \text{int } t + \text{int } p$ 
   $\text{int } p * \text{int } t' = \text{int } p * \text{int } t + \text{int } p$  unfolding  $b3$  by (simp-all add:
algebra-simps)
  have  $0 \leq 2 * (2 * \text{int } n + 2 - \text{int } p * \text{int } t')$  using  $b1\ e1$  by presburger
  moreover have  $2 * (2 * \text{int } n + 2 - \text{int } p * \text{int } t') \leq 4 * (\text{int } t' - \text{int } t)$ 
using  $b2\ b3\ e1$ 
  by presburger
  ultimately show ?thesis unfolding  $e$  unfolding  $t\text{-def}[\text{symmetric}]\ t'\text{-def}[\text{symmetric}]$ 
by (rule conjI)
qed
qed

```

```

lemma  $G4\text{-tele}$ :
  assumes  $2 \leq p$ 
  shows  $0 \leq G4\ p\ n - 4 * (\sum i < n. \text{int } (2 * i \text{ div } p)) \wedge$ 
 $G4\ p\ n - 4 * (\sum i < n. \text{int } (2 * i \text{ div } p)) \leq 4 * \text{int } (2 * n \text{ div } p)$ 
proof (induction n)
  case 0
  show ?case by (simp add: G4-def)
next
  case (Suc n)
  show ?case using  $\text{Suc } G4\text{-step}[\text{OF } \text{assms}, \text{of } n]$  by (simp add: algebra-simps)
qed

```

```

lemma  $\text{sum-Icc-eq-range}$ :  $f\ 0 = (0 :: \text{int}) \implies 1 \leq (h :: \text{nat}) \implies$ 
 $(\sum i = 1..h - 1. f\ i) = (\sum i < h. f\ i)$ 
proof -
  assume  $f0: f\ 0 = (0 :: \text{int})$  and  $h: 1 \leq h$ 
  obtain  $k$  where  $k: h = \text{Suc } k$  using  $h$  by (cases h) auto
  have  $s: \{..<h\} = \text{insert } 0\ \{1..h - 1\}$  unfolding  $k$  by force
  show ?thesis unfolding  $s$  using  $f0$  by (auto simp: sum.insert)
qed

```

$v\text{-}p(S\text{-}K) - p\ N(K/p) = (G\text{-}4(h) - 4 \sum \{i < h\} \lfloor 2i/p \rfloor) / 2$ ($K = 40n$:
 $\lambda K/p = h/p$, $\alpha K/p = N/p$ exactly).

lemma $vpS\text{-sub}$:

assumes p : prime p **and** h : InnerHyp K M p
shows $\text{of-int } (vpS\ K\ p) - \text{real } p * Nfun\text{-}r\ (\text{real } K / \text{real } p)$
 $= \text{of-int } (G4\ p\ (\text{hof } K) - 4 * (\sum i < \text{hof } K. \text{int } (2 * i \text{ div } p))) / 2$
proof –
obtain n **where** K : $K = 40 * n$ **using** h **unfolding** InnerHyp-def **by** (auto
elim: dvdE)
have hh : $\text{hof } K = 37 * n$ **and** hN : $\text{Nof } K = 3 * n$ **unfolding** K hof-def Nof-def
by simp-all
have pr : $0 < \text{real } p$ **using** p **by** (simp add: prime-gt-0-nat)
have v : $vpS\ K\ p = 2 * \text{int } (\text{hof } K) * \text{int } (K \text{ div } p) - 12 * \text{int } (\text{hof } K) * \text{int } (\text{Nof } K \text{ div } p) - 2 * (\sum i < \text{hof } K. \text{int } (2 * i \text{ div } p))$
unfolding vpS -inner[OF p h] of-nat-sum
proof –
have $0 < n$ **using** h **unfolding** InnerHyp-def K **by** simp
hence $h1$: $1 \leq \text{hof } K$ **unfolding** hh **by** fastforce
have s : $(\sum i = 1.. \text{hof } K - 1. \text{int } (2 * i \text{ div } p)) = (\sum i < \text{hof } K. \text{int } (2 * i \text{ div } p))$
by (rule sum-Icc-eq-range) (use $h1$ in simp-all)
show $2 * \text{int } (\text{hof } K) * \text{int } (K \text{ div } p) - 12 * \text{int } (\text{hof } K) * \text{int } (\text{Nof } K \text{ div } p) - 2 * (\sum i = 1.. \text{hof } K - 1. \text{int } (2 * i \text{ div } p))$
 $= 2 * \text{int } (\text{hof } K) * \text{int } (K \text{ div } p) - 12 * \text{int } (\text{hof } K) * \text{int } (\text{Nof } K \text{ div } p) - 2 * (\sum i < \text{hof } K. \text{int } (2 * i \text{ div } p))$
unfolding s ..
qed
have $e1$: $\text{lam-r} * \text{real } K / \text{real } p = \text{real } (\text{hof } K) / \text{real } p$ **and** $e2$: $\text{alpha-r} * (\text{real } K / \text{real } p) = \text{real } (\text{Nof } K) / \text{real } p$
and $e3$: $2 * (\text{real } (\text{hof } K) / \text{real } p) = \text{real } (2 * \text{hof } K) / \text{real } p$
unfolding hh hN **unfolding** K lam-r-def alpha-r-def **by** (simp-all add: field-simps)
have $f1$: $\text{floor } (\text{real } K / \text{real } p) = \text{int } (K \text{ div } p)$ **by** (rule floor-nat-div)
have $f2$: $\text{floor } (\text{alpha-r} * (\text{real } K / \text{real } p)) = \text{int } (\text{Nof } K \text{ div } p)$ **unfolding** $e2$ **by** (rule floor-nat-div)
have l : $\text{lam-r} * (\text{real } K / \text{real } p) = \text{real } (\text{hof } K) / \text{real } p$ **using** $e1$ **by** argo
have $f3$: $\text{floor } (2 * (\text{lam-r} * (\text{real } K / \text{real } p))) = \text{int } (2 * \text{hof } K \text{ div } p)$ **unfolding** l $e3$ **by** (rule floor-nat-div)
have pN : $\text{real } p * Nfun\text{-}r\ (\text{real } K / \text{real } p) = 2 * \text{real } (\text{hof } K) * \text{real } (K \text{ div } p) - 12 * \text{real } (\text{hof } K) * \text{real } (\text{Nof } K \text{ div } p) - 2 * \text{real } (2 * \text{hof } K \text{ div } p) * \text{real } (\text{hof } K) + \text{real } p * \text{real } (2 * \text{hof } K \text{ div } p) * (\text{real } (2 * \text{hof } K \text{ div } p) + 1) / 2$
proof –
have hh' : $\text{real } (\text{hof } K) = \text{lam-r} * \text{real } K$ **unfolding** hh **unfolding** K lam-r-def **by** simp
show ?thesis **unfolding** $Nfun\text{-}r$ -def J -r-def $f1$ $f2$ $f3$ **using** pr **by** (fastforce simp: field-simps hh')

```

qed
show ?thesis unfolding v pN G4-def by (force simp: of-int-sum)
qed

lemma vpS-bound:
  assumes p: prime p and h: InnerHyp K M p
  shows |of-int (vpS K p) - real p * Nfun-r (real K / real p)| ≤ 4 * real M
proof -
  have p2: 2 ≤ p using p by (rule prime-ge-2-nat)
  have t: 0 ≤ G4 p (hof K) - 4 * (∑ i < hof K. int (2 * i div p)) ∧
    G4 p (hof K) - 4 * (∑ i < hof K. int (2 * i div p)) ≤ 4 * int (2 * hof
K div p)
    by (rule G4-tele[OF p2])
  have q: 2 * hof K div p ≤ 2 * M
  proof -
    have hof K ≤ K unfolding hof-def by linarith
    moreover have K < p * M using h unfolding InnerHyp-def by simp
    ultimately have 2 * hof K < 2 * (p * M) by linarith
    hence 2 * hof K < (2 * M) * p by (force simp: algebra-simps)
    hence 2 * hof K div p < 2 * M using p2 by (force simp: div-less-iff-less-mult)
    thus ?thesis by simp
  qed
  define X where X = G4 p (hof K) - 4 * (∑ i < hof K. int (2 * i div p))
  have 0 ≤ (of-int X :: real) using t unfolding X-def[symmetric] by linarith
  moreover have (of-int X :: real) ≤ 8 * real M
  proof -
    have X ≤ 4 * int (2 * hof K div p) using t unfolding X-def by simp
    also have ... ≤ 4 * int (2 * M) using q by simp
    finally have X ≤ 8 * int M by simp
    thus ?thesis by simp
  qed
  ultimately show ?thesis unfolding vpS-sub[OF p h] X-def[symmetric] by (force
simp: abs-le-iff)
qed

```

25.2 The Riemann-sum identity for the z -integral

```

lemma floor-between:
  assumes 0 < p of-int j / real p < y y < (of-int j + 1) / real p
  shows floor y = j div int p
proof -
  have pr: 0 < real p using assms(1) by simp
  have r0: 0 ≤ j mod int p j mod int p < int p using assms(1) by simp-all
  have d: j div int p * int p + j mod int p = j by (rule div-mult-mod-eq)
  have e: (j div int p + 1) * int p = j div int p * int p + int p by (simp add:
algebra-simps)
  have i1: j div int p * int p ≤ j using r0 d by linarith
  have i2: j + 1 ≤ (j div int p + 1) * int p unfolding e using r0 d by linarith
  have l1: of-int (j div int p) * real p ≤ of-int j

```

```

proof –
  have real-of-int (j div int p * int p) ≤ real-of-int j using i1
  by (simp only: of-int-le-iff)
  thus ?thesis by simp
qed
have l2: of-int j + 1 ≤ (of-int (j div int p) + 1) * real p
proof –
  have real-of-int (j + 1) ≤ real-of-int ((j div int p + 1) * int p) using i2
  by (simp only: of-int-le-iff)
  thus ?thesis by simp
qed
have y1: of-int j < y * real p using assms(2) pr pos-divide-less-eq by blast
have y2: y * real p < of-int j + 1 using assms(3) pr by (fastforce simp:
pos-less-divide-eq)
have of-int (j div int p) ≤ y
proof –
  have of-int (j div int p) * real p < y * real p using l1 y1 by linarith
  thus ?thesis using pr by force
qed
moreover have y < of-int (j div int p) + 1
proof –
  have y * real p < (of-int (j div int p) + 1) * real p using l2 y2 by linarith
  thus ?thesis using pr by fastforce
qed
ultimately show ?thesis by (simp add: floor-eq-iff)
qed

```

l on the a -th grid cell $(a/p, (a+1)/p)$.

definition *ellZ* :: *nat* ⇒ *nat* ⇒ *nat* ⇒ *int* **where**
ellZ p A a = (*int A - int a - 1*) *div int p* + (*int A + int a*) *div int p* + 1

definition *gg* :: *int* ⇒ *int* ⇒ *int* ⇒ *int* **where**
gg T v l = (*T - 3 * v*) * (*T + 3 * v - l - 5*)

lemma *ell-on-cell*:

```

assumes  $0 < p$  real a / real p < z < (real a + 1) / real p
shows ell-r (real A / real p) z = of-int (ellZ p A a)
proof –
  have f1: floor (real A / real p - z) = (int A - int a - 1) div int p
  proof (rule floor-between[OF assms(1)])
    show real-of-int (int A - int a - 1) / real p < real A / real p - z using
assms(3)
    by (auto simp: diff-divide-distrib add-divide-distrib)
    show real A / real p - z < (real-of-int (int A - int a - 1) + 1) / real p using
assms(2)
    by (fastforce simp: diff-divide-distrib add-divide-distrib)
  qed
have f2: floor (real A / real p + z) = (int A + int a) div int p
proof (rule floor-between[OF assms(1)])

```

show $\text{real-of-int } (\text{int } A + \text{int } a) / \text{real } p < \text{real } A / \text{real } p + z$ **using** $\text{assms}(2)$
by $(\text{force simp: diff-divide-distrib add-divide-distrib})$
show $\text{real } A / \text{real } p + z < (\text{real-of-int } (\text{int } A + \text{int } a) + 1) / \text{real } p$ **using**
 $\text{assms}(3)$
by $(\text{fastforce simp: diff-divide-distrib add-divide-distrib})$
qed
show $?thesis$ **unfolding** $\text{ell-r-def ellZ-def f1 f2}$ **by** simp
qed

definition $\text{phiZ} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{int} \Rightarrow \text{real}$

$\Rightarrow \text{real}$ **where**
 $\text{phiZ } p \ N \ K \ T \ z = (\text{of-int } T - 3 * \text{ell-r } (\text{real } N / \text{real } p) \ z) * (\text{of-int } T + 3 * \text{ell-r } (\text{real } N / \text{real } p) \ z - \text{ell-r } (\text{real } K / \text{real } p) \ z - 5)$

lemma phi-on-cell :

assumes $0 < p$ $\text{real } a / \text{real } p < z$ $z < (\text{real } a + 1) / \text{real } p$
shows $\text{phiZ } p \ N \ K \ T \ z = \text{of-int } (gg \ T \ (\text{ellZ } p \ N \ a) \ (\text{ellZ } p \ K \ a))$
unfolding $\text{phiZ-def gg-def ell-on-cell}[OF \ \text{assms}, \text{ of } N] \ \text{ell-on-cell}[OF \ \text{assms}, \text{ of } K]$
by $(\text{simp add: algebra-simps})$

lemma integral-cell :

assumes $0 < p$ $\text{real } a / \text{real } p \leq c$ $c \leq (\text{real } a + 1) / \text{real } p$
shows $\text{interval-lebesgue-integrable lborel } (\text{ereal } (\text{real } a / \text{real } p)) \ (\text{ereal } c) \ (\text{phiZ } p \ N \ K \ T) \wedge$
 $\text{interval-lebesgue-integral lborel } (\text{ereal } (\text{real } a / \text{real } p)) \ (\text{ereal } c) \ (\text{phiZ } p \ N \ K \ T)$
 $= (c - \text{real } a / \text{real } p) * \text{of-int } (gg \ T \ (\text{ellZ } p \ N \ a) \ (\text{ellZ } p \ K \ a))$
by $(\text{rule const-piece}[OF \ \text{assms}(2)]) \ (\text{use assms}(1,3) \text{ in } \langle \text{auto simp: phi-on-cell} \rangle)$

$p \int -0^{1/2} \varphi = \sum -\{a < m\} \ gg(a) + gg(m)/2$ for $p = 2m + 1$.

lemma integral-sum :

assumes $\text{pm}: p = 2 * m + 1$
shows $\text{real } p * \text{interval-lebesgue-integral lborel } (\text{ereal } 0) \ (\text{ereal } (1 / 2)) \ (\text{phiZ } p \ N \ K \ T)$
 $= (\sum a < m. \text{of-int } (gg \ T \ (\text{ellZ } p \ N \ a) \ (\text{ellZ } p \ K \ a))) + \text{of-int } (gg \ T \ (\text{ellZ } p \ N \ m) \ (\text{ellZ } p \ K \ m)) / 2$

proof –

have $p0: 0 < p$ **using** pm **by** simp
have $\text{pr}: \text{real } p = 2 * \text{real } m + 1$ **using** pm **by** simp
have $\text{pre}: 0 < \text{real } p$ **using** $p0$ **by** simp
have $\text{cells}: \text{interval-lebesgue-integrable lborel } (\text{ereal } 0) \ (\text{ereal } (\text{real } k / \text{real } p)) \ (\text{phiZ } p \ N \ K \ T) \wedge$
 $\text{interval-lebesgue-integral lborel } (\text{ereal } 0) \ (\text{ereal } (\text{real } k / \text{real } p)) \ (\text{phiZ } p \ N \ K \ T)$
 $= (\sum a < k. (1 / \text{real } p) * \text{of-int } (gg \ T \ (\text{ellZ } p \ N \ a) \ (\text{ellZ } p \ K \ a)))$ **if** $k \leq m$
for k
using that

```

proof (induction k)
  case 0
  show ?case
  using const-piece[of 0 0 phiZ p N K T 0] by force
next
  case (Suc k)
  have c: interval-lebesgue-integrable lborel (ereal (real k / real p)) (ereal (real
(Suc k) /
    real p)) (phiZ p N K T) ∧
    interval-lebesgue-integral lborel (ereal (real k / real p)) (ereal (real (Suc k) /
    real p)) (phiZ p N K T)
    = (real (Suc k) / real p - real k / real p) * of-int (gg T (ellZ p N k) (ellZ
p K k))
  by (rule integral-cell[OF p0]) (simp-all add: divide-right-mono)
  have ih: interval-lebesgue-integrable lborel (ereal 0) (ereal (real k /
    real p)) (phiZ p N K T) ∧
    interval-lebesgue-integral lborel (ereal 0) (ereal (real k / real p)) (phiZ p N K
T)
    = (∑ a < k. (1 / real p) * of-int (gg T (ellZ p N a) (ellZ p K a)))
  using Suc.IH Suc.prem by simp
  have ch: interval-lebesgue-integrable lborel (ereal 0) (ereal (real (Suc k) /
    real p)) (phiZ p N K T) ∧
    interval-lebesgue-integral lborel (ereal 0) (ereal (real (Suc k) / real p)) (phiZ
p N K T)
    = (∑ a < k. (1 / real p) * of-int (gg T (ellZ p N a) (ellZ p K a)))
    + (real (Suc k) / real p - real k / real p) * of-int (gg T (ellZ p N k) (ellZ
p K k))
  by (rule chain[OF - ih c]) (simp-all add: divide-right-mono)
  have e: real (Suc k) / real p - real k / real p = 1 / real p
  by (force simp: diff-divide-distrib[symmetric])
  show ?case using ch unfolding e by simp
qed
have last: interval-lebesgue-integrable lborel (ereal (real m / real p)) (ereal (1 /
  2)) (phiZ p N K T) ∧
    interval-lebesgue-integral lborel (ereal (real m / real p)) (ereal (1 / 2)) (phiZ
p N K T)
    = (1 / 2 - real m / real p) * of-int (gg T (ellZ p N m) (ellZ p K m))
  by (rule integral-cell[OF p0]) (use pr pre in ⟨simp-all add: divide-le-eq
le-divide-eq⟩)
have half: 1 / 2 - real m / real p = 1 / (2 * real p)
proof -
  have 2 * real m + 1 ≠ 0 by simp
  hence d: 1 / 2 - real m / (2 * real m + 1) =
    (1 * (2 * real m + 1) - real m * 2) / (2 * (2 * real m + 1))
  by (simp add: diff-frac-eq)
  have n: 1 * (2 * real m + 1) - real m * 2 = (1::real) by simp
  show ?thesis unfolding pr d n by metis
qed
from chain[OF - - cells[OF le-refl] last]

```

have *interval-lebesgue-integral lborel* (*ereal 0*) (*ereal (1 / 2)*) (*phiZ p N K T*)
 = ($\sum a < m. (1 / \text{real } p) * \text{of-int } (gg \ T \ (\text{ellZ } p \ N \ a) \ (\text{ellZ } p \ K \ a))) + (1$
 $/ \ 2 -$
 $\text{real } m / \text{real } p) * \text{of-int } (gg \ T \ (\text{ellZ } p \ N \ m) \ (\text{ellZ } p \ K \ m))$
using *pr pre* **by** (*simp add: divide-le-eq*)
thus *?thesis unfolding half using pre* **by** (*simp add: sum-distrib-left field-simps*)
qed

25.3 Exact class counts

lemma *count-mod-eq*:

assumes $0 < p \ 1 \leq r \ r < p$
shows $\text{card } \{j \in \{1..A\}. j \bmod p = r\} = (A + p - r) \text{ div } p$
proof –
have $e: \{j \in \{1..A\}. j \bmod p = r\} = (\lambda t. t * p + r) \text{ ‘ } \{.. < (A + p - r) \text{ div } p\}$
proof (*intro equalityI subsetI*)
fix j **assume** $j \in \{j \in \{1..A\}. j \bmod p = r\}$
hence $j: 1 \leq j \ j \leq A \ j \bmod p = r$ **by** *auto*
have $jd: j = j \text{ div } p * p + r$ **using** $j(3)$ *div-mult-mod-eq[of j p]* **by** *simp*
have $(j \text{ div } p + 1) * p \leq A + p - r$ **using** $jd \ j(2)$ **by** (*simp add: algebra-simps*)
hence $j \text{ div } p + 1 \leq (A + p - r) \text{ div } p$ **using** *assms(1)* **using** *less-eq-div-iff-mult-less-eq*
by *blast*
hence $j \text{ div } p < (A + p - r) \text{ div } p$ **by** *linarith*
thus $j \in (\lambda t. t * p + r) \text{ ‘ } \{.. < (A + p - r) \text{ div } p\}$ **using** jd
by (*intro rev-image-eqI[of j div p]*) *simp-all*
next
fix j **assume** $j \in (\lambda t. t * p + r) \text{ ‘ } \{.. < (A + p - r) \text{ div } p\}$
then obtain t **where** $t < (A + p - r) \text{ div } p \ j = t * p + r$ **by** *auto*
have $t + 1 \leq (A + p - r) \text{ div } p$ **using** $t(1)$ **by** *linarith*
hence $(t + 1) * p \leq A + p - r$ **using** *assms(1)* **by** (*metis less-eq-div-iff-mult-less-eq*)
hence $j \leq A$ **using** $t(2)$ *assms* **by** (*simp add: algebra-simps*)
moreover **have** $j \bmod p = r$ **using** $t(2)$ *assms(3)* **by** *simp*
ultimately show $j \in \{j \in \{1..A\}. j \bmod p = r\}$ **using** $t(2)$ *assms(2)* **by** *simp*
qed
have *inj*: *inj-on* $(\lambda t. t * p + r) \text{ ‘ } \{.. < (A + p - r) \text{ div } p\}$ **using** *assms(1)*
by (*auto simp: inj-on-def*)
show *?thesis unfolding e card-image[OF inj]* **by** *simp*
qed

lemma *ell-exact*:

assumes $\text{odd } p \ 1 \leq a \ a \leq m\text{Half } p$
shows $\text{ell } p \ A \ a = (A + p - a) \text{ div } p + (A + a) \text{ div } p$
proof –
have $ap: a < p \ p - a \neq a \ 1 \leq p - a \ p - a < p$ **using** *assms* **unfolding**
mHalf-def
by (*auto elim!: oddE*)
have *split*: $\{j \in \{1..A\}. j \bmod p = a \bmod p \vee$
 $(j + a) \bmod p = 0\} = \{j \in \{1..A\}. j \bmod p = a\} \cup \{j \in \{1..A\}. j \bmod p = p$
 $- a\}$

```

proof -
  have p0:  $0 < p$  using ap(1) by simp
  have m:  $(j + a) \bmod p = 0 \iff j \bmod p = p - a$  for j
  proof -
    have r:  $j \bmod p < p$  using p0 by simp
    have  $(j + a) \bmod p = 0 \iff j \bmod p + a = 0 \vee j \bmod p + a = p$ 
      using mod-add-zero-iff-aux[OF r ap(1)] by (simp add: mod-add-left-eq)
    thus ?thesis using assms(2) ap(1) r by auto
  qed
  show ?thesis using m ap by auto
qed
have disj:  $\{j \in \{1..A\}. j \bmod p = a\} \cap \{j \in \{1..A\}. j \bmod p = p - a\} = \{\}$ 
  using ap by auto
have p0:  $0 < p$  using ap(1) by simp
have c:  $\text{card } (\{j \in \{1..A\}. j \bmod p = a\} \cup \{j \in \{1..A\}. j \bmod p = p - a\})$ 
   $= \text{card } \{j \in \{1..A\}. j \bmod p = a\} + \text{card } \{j \in \{1..A\}. j \bmod p = p - a\}$ 
  by (rule card-Un-disjoint) (use disj in auto)
have c1:  $\text{card } \{j \in \{1..A\}. j \bmod p = a\} = (A + p - a) \text{ div } p$ 
  by (rule count-mod-eq) (use p0 ap assms(2) in auto)
have c2:  $\text{card } \{j \in \{1..A\}. j \bmod p = p - a\} = (A + a) \text{ div } p$ 
  using count-mod-eq[OF p0 ap(3) ap(4), of A] ap(1) by simp
show ?thesis unfolding ell-def split c c1 c2 ..
qed

lemma ell-succ-eq:
  assumes odd p  $a + 1 \leq mHalf\ p$ 
  shows  $\text{int } (\text{ell } p\ A\ (a + 1)) = \text{ellZ } p\ A\ a + ((\text{int } A + \text{int } a + 1) \text{ div } \text{int } p -$ 
 $(\text{int } A +$ 
 $\text{int } a) \text{ div } \text{int } p)$ 
  proof -
    have ap:  $a + 1 < p$  using assms unfolding mHalf-def by linarith
    have ex:  $\text{ell } p\ A\ (a + 1) = (A + p - (a + 1)) \text{ div } p + (A + (a + 1)) \text{ div } p$ 
      by (rule ell-exact[OF assms(1) - assms(2)]) simp
    have e1:  $\text{int } ((A + p - (a + 1)) \text{ div } p) = (\text{int } A - \text{int } a - 1) \text{ div } \text{int } p + 1$ 
    proof -
      have  $\text{int } ((A + p - (a + 1)) \text{ div } p) = \text{int } (A + p - (a + 1)) \text{ div } \text{int } p$  by
        (rule zdiv-int)
      also have  $\text{int } (A + p - (a + 1)) = (\text{int } A - \text{int } a - 1) + \text{int } p$  using ap by
        simp
      also have  $((\text{int } A - \text{int } a - 1) + \text{int } p) \text{ div } \text{int } p = (\text{int } A - \text{int } a - 1) \text{ div } \text{int } p + 1$ 
        using ap by simp
      finally show ?thesis .
    qed
    have e2:  $\text{int } ((A + (a + 1)) \text{ div } p) = (\text{int } A + \text{int } a + 1) \text{ div } \text{int } p$  unfolding
      zdiv-int
      by (simp add: algebra-simps)
    show ?thesis unfolding ex ellZ-def of-nat-add e1 e2 by linarith
  qed

```



```

lemma sum-disc:
  assumes  $0 < p$   $m < p$ 
  shows  $(\sum a < m. ((\text{int } A + \text{int } a + 1) \text{ div int } p - (\text{int } A + \text{int } a) \text{ div int } p))$ 
 $\leq 1$ 
    and  $0 \leq (\text{int } A + \text{int } a + 1) \text{ div int } p - (\text{int } A + \text{int } a) \text{ div int } p$ 
proof -
  have tele:  $(\sum a < k. ((\text{int } A + \text{int } a + 1) \text{ div int } p - (\text{int } A + \text{int } a) \text{ div int } p)) =$ 
 $(\text{int } A + \text{int } k) \text{ div int } p - \text{int } A \text{ div int } p$  for  $k$ 
    by (induction  $k$ ) (simp-all add: algebra-simps)
  show  $(\sum a < m. ((\text{int } A + \text{int } a + 1) \text{ div int } p - (\text{int } A + \text{int } a) \text{ div int } p))$ 
 $\leq 1$ 
    unfolding tele
    using zdiv-mono1[of int  $A + \text{int } m$  int  $A + \text{int } p$  int  $p$ ] assms by (simp add:
div-add-self2)
  show  $0 \leq (\text{int } A + \text{int } a + 1) \text{ div int } p - (\text{int } A + \text{int } a) \text{ div int } p$ 
    using zdiv-mono1[of int  $A + \text{int } a$  int  $A + \text{int } a + 1$  int  $p$ ] assms(1) by simp
qed

lemma ell-cases:
  assumes odd  $p$   $1 \leq a$   $a \leq m\text{Half } p$ 
  shows  $\text{ell } p \ A \ a = 2 * A \text{ div } p \vee \text{ell } p \ A \ a = 2 * A \text{ div } p + 1$ 
proof -
  have  $p0$ :  $0 < p$  using assms(1) by (rule odd-pos)
  have  $ap$ :  $a < p$  using assms(3)  $p0$  unfolding mHalf-def by linarith
  define  $u$  where  $u = (A + p - a) \text{ div } p$ 
  define  $v$  where  $v = (A + a) \text{ div } p$ 
  define  $q$  where  $q = 2 * A \text{ div } p$ 
  have  $g$ :  $x \text{ div } p * p \leq x \ x < (x \text{ div } p + 1) * p$  for  $x$ 
proof -
  show  $x \text{ div } p * p \leq x$  by fastforce
  have  $e$ :  $(x \text{ div } p + 1) * p = x \text{ div } p * p + p$  by (simp add: algebra-simps)
  show  $x < (x \text{ div } p + 1) * p$  unfolding  $e$ 
    using mod-less-divisor[OF  $p0$ , of  $x$ ] div-mult-mod-eq[of  $x$   $p$ ] by linarith
qed
  have  $bu$ :  $u * p \leq A + p - a$   $A + p - a < (u + 1) * p$  unfolding  $u$ -def by
(rule  $g$ )+
  have  $bv$ :  $v * p \leq A + a$   $A + a < (v + 1) * p$  unfolding  $v$ -def by (rule  $g$ )+
  have  $bq$ :  $q * p \leq 2 * A$   $2 * A < (q + 1) * p$  unfolding  $q$ -def by (rule  $g$ )+
  have  $s1$ :  $(u + v) * p < (q + 2) * p$  using  $bu$   $bv$   $bq$   $ap$  by (simp add: algebra-simps)
  have  $s2$ :  $(q + 1) * p < (u + v + 2) * p$  using  $bu$   $bv$   $bq$   $ap$  by (simp add:
algebra-simps)
  have  $u + v < q + 2$  using  $s1$  using mult-less-cancel2 by blast
  moreover have  $q + 1 < u + v + 2$  using  $s2$  using mult-less-cancel2 by blast
  ultimately show ?thesis unfolding ell-exact[OF assms]  $u$ -def[symmetric]  $v$ -def[symmetric]
 $q$ -def[symmetric] by linarith
qed

```

lemma *card-inter-of-chain*: $P \subseteq Q \vee Q \subseteq P \implies$
 $\text{finite } P \implies \text{finite } Q \implies \text{card } (P \cap Q) = \min (\text{card } P) (\text{card } Q)$
proof –
 assume c : $P \subseteq Q \vee Q \subseteq P$ and fP : *finite* P and fQ : *finite* Q
 show ?thesis
 proof (cases $P \subseteq Q$)
 case True
 hence $P \cap Q = P$ card $P \leq \text{card } Q$ using *card-mono*[*OF fQ True*] by auto
 thus ?thesis by simp
 next
 case False
 hence Q : $Q \subseteq P$ using c by simp
 hence $P \cap Q = Q$ card $Q \leq \text{card } P$ using *card-mono*[*OF fP Q*] by auto
 thus ?thesis by (simp add: *min.absorb2*)
 qed
 qed

lemma *sum-two-lin*: $(\sum i < L. 2 * (\text{of-nat } i + c)) =$
 $\text{of-nat } L * (\text{of-nat } L - 1) + 2 * \text{of-nat } L * (c :: \text{rat})$
 by (induction L) (simp-all add: *algebra-simps*)

25.4 The discrete decomposition of $\gamma\text{-}p\hat{\text{in}}$

The number of classes with $l\text{-}K(a) = q + 1$, $q = \lfloor 2K/p \rfloor$.

definition $c1 :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat}$ **where**
 $c1 \ K \ p = \text{card } \{a \in \{1..m\text{Half } p\}. \text{ell } p \ K \ a = 2 * K \text{ div } p + 1\}$

lemma *sum-ellK*:
 assumes p : *prime* p and h : *InnerHyp* $K \ M \ p$
 shows $(\sum a = 1..m\text{Half } p. \text{ell } p \ K \ a) = m\text{Half } p * (2 * K \text{ div } p) + c1 \ K \ p$
proof –
 have *odd*: *odd* p using *hyp-facts*[*OF p h*] by simp
 have e : $\bigwedge a. a \in \{1..m\text{Half } p\} \implies$
 $\text{ell } p \ K \ a = 2 * K \text{ div } p + (\text{if } \text{ell } p \ K \ a = 2 * K \text{ div } p + 1 \text{ then } 1 \text{ else } 0)$
 using *ell-cases*[*OF odd*] by fastforce
 have $(\sum a = 1..m\text{Half } p. \text{ell } p \ K \ a) = (\sum a = 1..m\text{Half } p. 2 * K \text{ div } p + (\text{if } \text{ell } p \ K \ a =$
 $2 * K \text{ div } p + 1 \text{ then } 1 \text{ else } 0))$
 by (rule *sum.cong*[*OF refl*]) (rule e)
 also have $\dots = m\text{Half } p * (2 * K \text{ div } p) + \text{card } \{a \in \{1..m\text{Half } p\}. \text{ell } p \ K \ a =$
 $2 * K \text{ div } p + 1\}$
 by (simp add: *sum.distrib sum.inter-filter*[*symmetric*])
 finally show ?thesis unfolding *c1-def* .
 qed

lemma *sum-ellK'*:
 assumes p : *prime* p and h : *InnerHyp* $K \ M \ p$
 shows $(\sum a = 1..m\text{Half } p. \text{ell } p \ K \ a) = K - K \text{ div } p$
 using *sum-ell*[*OF p*] *hyp-facts*[*OF p h*] by simp

lemma *c1-le*: $c1\ K\ p \leq mHalf\ p$
unfolding *c1-def* **by** (*rule order-trans*[*OF card-mono*[*of* $\{1..mHalf\ p\}$]]) *auto*

$\varepsilon\text{-}a \in \{0, 1\}$, and $\varepsilon\text{-}a = 1$ exactly when the rank of a is $< E$.

lemma *eps-cases*: $epsIn\ K\ M\ p\ a = 0 \vee epsIn\ K\ M\ p\ a = 1$
by (*simp add: epsIn-def*)

lemma *eps-iff*:

assumes p : *prime* p **and** h : *InnerHyp* $K\ M\ p$ **and** a : $a \in \{1..mHalf\ p\}$
shows $epsIn\ K\ M\ p\ a = 1 \iff int\ (defaultPos\ K\ p\ a) < Ein\ K\ M\ p$

proof –

define m **where** $m = mHalf\ p$

define A **where** $A = \{1..m\}$

define r **where** $r\ c\ a \iff ell\ p\ K\ a < ell\ p\ K\ c \vee$
 $(ell\ p\ K\ c = ell\ p\ K\ a \wedge c < a)$ **for** $c\ a$

have rk : $defaultPos\ K\ p\ a = card\ \{c \in A. r\ c\ a\}$ **for** a
by (*simp add: defaultPos-def A-def m-def r-def*)

have irr : $\bigwedge x. \neg r\ x\ x$ **and** tr : $\bigwedge x\ y\ z. r\ x\ y \implies$
 $r\ y\ z \implies r\ x\ z$

and tot : $\bigwedge x\ y. x \in A \implies y \in A \implies$
 $x \neq y \implies r\ x\ y \vee r\ y\ x$

unfolding r -def **by** *auto*

note $RP = rank\ props[of\ A\ r, OF - irr\ tr\ tot, folded\ rk]$

have iff : $x \in A \implies y \in A \implies$

$defaultPos\ K\ p\ x < defaultPos\ K\ p\ y \iff r\ x\ y$ **for** $x\ y$

using $RP(1)$ **by** (*simp add: A-def*)

have aA : $a \in A$ **using** a **by** (*simp add: A-def m-def*)

have $\{c \in \{1..mHalf\ p\}. defaultPos\ K\ p\ c < defaultPos\ K\ p\ a\} = \{c \in A. r\ c\ a\}$
using $iff[OF - aA]$ **by** (*auto simp: A-def m-def*)

hence $card\ \{c \in \{1..mHalf\ p\}. defaultPos\ K\ p\ c < defaultPos\ K\ p\ a\} = defaultPos\ K\ p\ a$

using rk **by** *simp*

thus *?thesis* **unfolding** *epsIn-def* **by** *simp*

qed

lemma *defaultPos-lt*:

assumes $a \in \{1..mHalf\ p\}$ $c \in \{1..mHalf\ p\}$ $ell\ p\ K\ c < ell\ p\ K\ a$
shows $defaultPos\ K\ p\ a < defaultPos\ K\ p\ c$

proof –

define m **where** $m = mHalf\ p$

define A **where** $A = \{1..m\}$

define r **where** $r\ c\ a \iff ell\ p\ K\ a < ell\ p\ K\ c \vee$
 $(ell\ p\ K\ c = ell\ p\ K\ a \wedge c < a)$ **for** $c\ a$

have rk : $defaultPos\ K\ p\ a = card\ \{c \in A. r\ c\ a\}$ **for** a
by (*simp add: defaultPos-def A-def m-def r-def*)

have irr : $\bigwedge x. \neg r\ x\ x$ **and** tr : $\bigwedge x\ y\ z. r\ x\ y \implies$
 $r\ y\ z \implies r\ x\ z$

and tot : $\bigwedge x\ y. x \in A \implies y \in A \implies$

```

     $x \neq y \implies r\ x\ y \vee r\ y\ x$ 
    unfolding r-def by auto
    note RP = rank-props[of A r, OF - irr tr tot, folded rk]
    have iff:  $x \in A \implies y \in A \implies$ 
      defaultPos K p x < defaultPos K p y  $\longleftrightarrow r\ x\ y$  for  $x\ y$ 
    using RP(1) by (simp add: A-def)
    have aA:  $a \in A$  and cA:  $c \in A$  using assms(1,2) by (simp-all add: A-def m-def)
    show ?thesis using iff[OF aA cA] assms(3) unfolding r-def by simp
  qed

```

$\sum -a\ \varepsilon\text{-}a\ l\text{-}K(a) = q\ E + \min(E, c-1)$: the classes with $\varepsilon = 1$ and those with $l\text{-}K = q + 1$ form a chain.

```

lemma sum-if-const: finite A  $\implies B \subseteq A \implies$ 
  ( $\sum x \in A. \text{if } x \in B \text{ then } c \text{ else } 0$ ) = int (card B) * (c :: int)
  by (simp add: sum.If-cases Int-absorb1 Int-absorb2)

```

lemma sum-eps-ell:

```

  assumes p: prime p and h: InnerHyp K M p
  shows ( $\sum a = 1..m\text{Half } p. \text{epsIn } K\ M\ p\ a * \text{int } (\text{ell } p\ K\ a)$ ) =
    int (2 * K div p) * Ein K M p + min (Ein K M p) (int (c1 K p))
  proof -
    have odd: odd p using hyp-facts[OF p h] by simp
    have E:  $0 \leq \text{Ein } K\ M\ p$  and  $\text{Ein } K\ M\ p < \text{int } (m\text{Half } p)$  using TE-spec[OF p h] by
    auto
    define P where P = {a  $\in$  {1..mHalf p}. int (defaultPos K p a) < Ein K M p}
    define Q where Q = {a  $\in$  {1..mHalf p}. ell p K a = 2 * K div p + 1}
    have cardP: int (card P) = Ein K M p
    proof -
      define m where m = mHalf p
      define A where A = {1..m}
      define r where r c a  $\longleftrightarrow \text{ell } p\ K\ a < \text{ell } p\ K\ c \vee$ 
        ( $\text{ell } p\ K\ c = \text{ell } p\ K\ a \wedge c < a$ ) for c a
      have rk: defaultPos K p a = card {c  $\in$  A. r c a} for a
        by (simp add: defaultPos-def A-def m-def r-def)
      have irr:  $\bigwedge x. \neg r\ x\ x$  and tr:  $\bigwedge x\ y\ z. r\ x\ y \implies$ 
        r y z  $\implies r\ x\ z$ 
        and tot:  $\bigwedge x\ y. x \in A \implies y \in A \implies$ 
           $x \neq y \implies r\ x\ y \vee r\ y\ x$ 
      unfolding r-def by auto
      note RP = rank-props[of A r, OF - irr tr tot, folded rk]
      have cA: card A = m by (simp add: A-def)
      have bij: bij-betw (defaultPos K p) A {.. $m$ } using RP(2) cA by (simp add:
      A-def)
      define En where En = nat (Ein K M p)
      have En: Ein K M p = int En and En  $\leq m$  using E by (auto simp: En-def m-def)
      have PE: P = {a  $\in$  A. defaultPos K p a < En} unfolding P-def A-def m-def
      En(1) by simp
      have card {a  $\in$  A. defaultPos K p a < En} = card {.. $En$ }
    proof -

```

```

have inj: inj-on (defaultPos K p) {a ∈ A. defaultPos K p a < En}
  using bij by (auto simp: bij-betw-def intro: inj-on-subset)
have defaultPos K p ' {a ∈ A. defaultPos K p a < En} = {.. $En$ }
proof (intro equalityI subsetI)
  fix i assume i ∈ {.. $En$ }
  hence i ∈ {.. $m$ } using En by auto
  hence i ∈ defaultPos K p ' A using bij by (simp add: bij-betw-def)
  then obtain a where a ∈ A defaultPos K p a = i by blast
  thus i ∈ defaultPos K p ' {a ∈ A. defaultPos K p a < En}
    using {i ∈ {.. $En$ }} by auto
qed auto
thus ?thesis using card-image[OF inj] by simp
qed
thus ?thesis unfolding PE using En by simp
qed
have cardQ: card Q = c1 K p unfolding Q-def c1-def ..
have e:  $\bigwedge a. a \in \{1..mHalf\ p\} \implies$ 
   $epsIn\ K\ M\ p\ a * int\ (ell\ p\ K\ a) = (if\ a \in P\ then\ int\ (2 * K\ div\ p)\ else\ 0) +$ 
  (if a ∈
     $P \cap Q\ then\ 1\ else\ 0)$ 
proof -
  fix a assume a: a ∈ {1..mHalf p}
  have epsIn K M p a = 1  $\longleftrightarrow$  a ∈ P using eps-iff[OF p h a] a
    unfolding P-def by simp
  moreover have epsIn K M p a = 0  $\vee$  epsIn K M p a = 1 by (rule eps-cases)
  moreover have ell p K a = 2 * K div p  $\vee$  ell p K a = 2 * K div p + 1
    using ell-cases[OF odd] a by auto
  moreover have a ∈ Q  $\longleftrightarrow$  ell p K a = 2 * K div p + 1 using a
    unfolding Q-def by simp
  ultimately show epsIn K M p a * int (ell p K a) = (if a ∈
     $P\ then\ int\ (2 * K\ div\ p)\ else\ 0) + (if\ a \in P \cap Q\ then\ 1\ else\ 0)$ 
    by fastforce
qed
have chain:  $P \subseteq Q \vee Q \subseteq P$ 
proof (rule ccontr)
  assume  $\neg (P \subseteq Q \vee Q \subseteq P)$ 
  then obtain c a where c: c ∈ P c  $\notin Q$  and a: a ∈ Q a  $\notin P$  by blast
  have c1: c ∈ {1..mHalf p} and a1: a ∈ {1..mHalf p} using a c unfolding
P-def Q-def
    by auto
  have cq: ell p K c = 2 * K div p using ell-cases[OF odd, of c K] c c1 unfolding
Q-def by auto
  have aq: ell p K a = 2 * K div p + 1 using a unfolding Q-def by auto
  have defaultPos K p a < defaultPos K p c by (rule defaultPos-lt[OF a1 c1])
  (use cq aq in simp)
  hence a ∈ P using c a1 unfolding P-def by auto
  thus False using a by simp
qed
have ( $\sum a = 1..mHalf\ p. epsIn\ K\ M\ p\ a * int\ (ell\ p\ K\ a)$ ) =

```

```

    int (2 * K div p) * int (card P) + int (card (P ∩ Q))
  proof -
    have s1: (∑ a = 1..mHalf p. epsIn K M p a * int (ell p K a)) = (∑ a =
      1..mHalf p. (if a ∈ P then int (2 * K div p) else 0) + (if a ∈
      P ∩ Q then 1 else 0))
    using e by simp
    have Psub: P ⊆ {1..mHalf p} unfolding P-def by blast
    have PQsub: P ∩ Q ⊆ {1..mHalf p} using Psub by blast
    note s2 = sum-if-const[OF finite-atLeastAtMost Psub, of int (2 * K div p)]
    note s3 = sum-if-const[OF finite-atLeastAtMost PQsub, of 1]
    show ?thesis unfolding s1 sum.distrib s2 s3 by (simp add: algebra-simps)
  qed
  note s = this
  have fin: finite P finite Q by (simp-all add: P-def Q-def)
  have m: int (card (P ∩ Q)) = min (Ein K M p) (int (c1 K p))
    unfolding card-inter-of-chain[OF chain fin] using cardP cardQ by simp
  show ?thesis unfolding s m cardP ..
  qed

```

The sum over the rows of class $a \geq 1$.

lemma *class-sum*:

```

  assumes p: prime p and h: InnerHyp K M p and a: a ∈ {1..mHalf p}
  shows (∑ i < nat (Lin K M p a). 2 * wIn K M p a i)
    = of-int (gg (Tin K M p) (int (ell p (Nof K) a)) (int (ell p K a)) + epsIn
    K M p a * (2 *
      Tin K M p - int (ell p K a) - 4))
  proof -
    have a0: a ≠ 0 using a by auto
    have L: 0 ≤ Lin K M p a using Lin-nonneg[OF p h] a by auto
    have Lq: (of-nat (nat (Lin K M p a)) :: rat) =
      of-int (Tin K M p) - 3 * of-nat (ell p (Nof K) a) + of-int (epsIn K M p a)
    using L unfolding Lin-def bIn-def by (simp add: a0)
    have w: ⋀ i. wIn K M p a i = of-nat i + (of-int (bIn K p a) - (of-nat (ell p K
    a) + 4) / 2)
    using a0 by (simp add: wIn-def)
    have (∑ i < nat (Lin K M p a). 2 * wIn K M p a i) = (∑ i <
      nat (Lin K M p a). 2 * (of-nat i + (of-int (bIn K p a) - (of-nat (ell p K a) +
      4) / 2)))
    by (simp add: w)
    also have ... = of-nat (nat (Lin K M p a)) * (of-nat (nat (Lin K M p a)) -
    1) + 2 *
      of-nat (nat (Lin K M p a)) * (of-int (bIn K p a) - (of-nat (ell p K a) + 4) /
    2)
    by (rule sum-two-lin)
    finally have F: (∑ i < nat (Lin K M p a). 2 * wIn K M p a i) =
      of-nat (nat (Lin K M p a)) * (of-nat (nat (Lin K M p a)) - 1)
      + 2 * of-nat (nat (Lin K M p a)) * (of-int (bIn K p a) - (of-nat (ell p K a)
      + 4) / 2) .
    show ?thesis

```

unfolding $F\ Lq\ gg\text{-}def\ bIn\text{-}def$ **using** $eps\text{-}cases[of\ K\ M\ p\ a]$ **by** $(auto\ simp: algebra\text{-}simps)$

qed

definition $Sd :: nat \Rightarrow nat \Rightarrow int \Rightarrow int$ **where**

$Sd\ K\ p\ T = (\sum\ a = 1..mHalf\ p.\ gg\ T\ (int\ (ell\ p\ (Nof\ K)\ a))\ (int\ (ell\ p\ K\ a)))$

$\gamma\text{-}in = 2\ \sum\ \{i < 4M+10\}\ w\{0,i\} + [S\text{-}d(T) + E\ (2T - q - 4) - min(E, c-1)].$

lemma $gamma\text{-}decomp$:

assumes p : $prime\ p$ **and** h : $InnerHyp\ K\ M\ p$

shows $gammaIn\ K\ M\ p = 2 * (\sum\ i < 4 * M + 10.\ wIn\ K\ M\ p\ 0\ i) + of\text{-}int\ (Sd\ K\ p\ (Tin\ K\ M\ p) + Ein\ K\ M\ p * (2 * Tin\ K\ M\ p - int\ (2 * K\ div\ p) - 4) - min\ (Ein\ K\ M\ p)\ (int\ (c1\ K\ p)))$

proof –

have $L0$: $nat\ (Lin\ K\ M\ p\ 0) = 4 * M + 10$ **by** $(simp\ add: Lin\text{-}def\ L0\text{-}def)$

have $gammaIn\ K\ M\ p = 2 * (\sum\ i < 4 * M + 10.\ wIn\ K\ M\ p\ 0\ i) + (\sum\ a = 1..mHalf\ p.\ \sum\ i < nat\ (Lin\ K\ M\ p\ a).\ 2 * wIn\ K\ M\ p\ a\ i)$

proof –

have s : $\{..mHalf\ p\} = insert\ 0\ \{1..mHalf\ p\}$ **by** $auto$

show $?thesis$ **unfolding** $gammaIn\text{-}def\ s$ **using** $L0$ **by** $(simp\ add: sum\text{-}distrib\text{-}left)$

qed

also have $(\sum\ a = 1..mHalf\ p.\ \sum\ i < nat\ (Lin\ K\ M\ p\ a).\ 2 * wIn\ K\ M\ p\ a\ i) = of\text{-}int\ (Sd\ K\ p\ (Tin\ K\ M\ p) + Ein\ K\ M\ p * (2 * Tin\ K\ M\ p - int\ (2 * K\ div\ p) - 4) - min\ (Ein\ K\ M\ p)\ (int\ (c1\ K\ p)))$

proof –

have $es0$: $(\sum\ a = 1..mHalf\ p.\ epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - int\ (ell\ p\ K\ a) - 4)) = (\sum\ a = 1..mHalf\ p.\ epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - 4) - (\sum\ a = 1..mHalf\ p.\ epsIn\ K\ M\ p\ a * int\ (ell\ p\ K\ a)))$

proof –

have pt : $\bigwedge a.\ epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - int\ (ell\ p\ K\ a) - 4) = epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - 4) - epsIn\ K\ M\ p\ a * int\ (ell\ p\ K\ a)$ **by** $(simp\ add: algebra\text{-}simps)$

show $?thesis$ **unfolding** $pt\ sum\text{-}subtractf$ **by** $(simp\ add: sum\text{-}distrib\text{-}right)$

qed

have es : $(\sum\ a = 1..mHalf\ p.\ epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - int\ (ell\ p\ K\ a) - 4)) = Ein\ K\ M\ p * (2 * Tin\ K\ M\ p - int\ (2 * K\ div\ p) - 4) - min\ (Ein\ K\ M\ p)\ (int\ (c1\ K\ p))$

unfolding $es0\ sum\text{-}epsIn[OF\ p\ h]\ sum\text{-}eps\text{-}ell[OF\ p\ h]$ **by** $(simp\ add: algebra\text{-}simps)$

have cs : $(\sum\ a = 1..mHalf\ p.\ \sum\ i < nat\ (Lin\ K\ M\ p\ a).\ 2 * wIn\ K\ M\ p\ a\ i) = (\sum\ a = 1..mHalf\ p.\ (of\text{-}int\ (gg\ (Tin\ K\ M\ p)\ (int\ (ell\ p\ (Nof\ K)\ a))\ (int\ (ell\ p\ K\ a)) + epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - int\ (ell\ p\ K\ a) - 4)) :: rat))$

using $class\text{-}sum[OF\ p\ h]$ **by** $simp$

have $cs2$: $(\sum a = 1..mHalf\ p. (of-int\ (gg\ (Tin\ K\ M\ p)\ (int\ (ell\ p\ (Nof\ K)\ a))$
 $(int\ (ell\ p\ K$
 $a)) + epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - int\ (ell\ p\ K\ a) - 4)) :: rat))$
 $= of-int\ (Sd\ K\ p\ (Tin\ K\ M\ p) + (\sum a =$
 $1..mHalf\ p. epsIn\ K\ M\ p\ a * (2 * Tin\ K\ M\ p - int\ (ell\ p\ K\ a) - 4)))$
by $(simp\ add: Sd-def\ of-int-sum\ sum.distrib)$
show $?thesis$ **unfolding** $cs\ cs2\ es$ **by** $simp$
qed
finally show $?thesis$.
qed

lemma $Tin-ge$:

assumes p : $prime\ p$ **and** h : $InnerHyp\ K\ M\ p$
shows $-1 \leq Tin\ K\ M\ p$
proof –
have $1 \leq mHalf\ p$ **using** $hyp-facts[OF\ p\ h]$ **unfolding** $mHalf-def$ **by** $linarith$
hence $0 \leq Lin\ K\ M\ p\ 1$ **by** $(rule\ Lin-nonneg[OF\ p\ h])$
moreover have $Lin\ K\ M\ p\ 1 = Tin\ K\ M\ p - bIn\ K\ p\ 1 + epsIn\ K\ M\ p\ 1$ **by**
 $(simp\ add: Lin-def)$
moreover have $0 \leq bIn\ K\ p\ 1$ **by** $(simp\ add: bIn-def)$
moreover have $epsIn\ K\ M\ p\ 1 \leq 1$ **by** $(simp\ add: epsIn-def)$
ultimately show $?thesis$ **by** $linarith$
qed

lemma $w0-bound$:

assumes p : $prime\ p$ **and** h : $InnerHyp\ K\ M\ p$ **and** i : $i < 4 * M + 10$
shows $|wIn\ K\ M\ p\ 0\ i| \leq 15 * of-nat\ M$
proof –
have hf : $7 \leq p$ $odd\ p$ $K\ div\ p < M$ $Nof\ K\ div\ p < M$ **using** $hyp-facts[OF\ p\ h]$
by $auto$
have M : $40 \leq M$ **using** h **unfolding** $InnerHyp-def$ **by** $simp$
have Tl : $-1 \leq Tin\ K\ M\ p$ **by** $(rule\ Tin-ge[OF\ p\ h])$
have Tu : $Tin\ K\ M\ p \leq 4 * int\ M$ **by** $(rule\ Tin-le[OF\ p\ h])$
define $x0$:: rat **where** $x0 = 2 * of-nat\ i + 6 * of-nat\ (Nof\ K\ div\ p) - of-nat$
 $(K\ div\ p) + 1 / 2$
define f :: $nat \Rightarrow rat$ **where** $f\ c =$
 $of-int\ (Zin\ K\ M\ p\ c) - (of-nat\ (ell\ p\ K\ c) + 4) / 2$ **for** c
define S **where** $S = insert\ x0\ (f\ ` \{1..mHalf\ p\})$
have w : $wIn\ K\ M\ p\ 0\ i = Min\ S$ **unfolding** $wIn-def\ S-def\ x0-def\ f-def$ **by** $simp$
have fin : $finite\ S$ **and** ne : $S \neq \{\}$ **unfolding** $S-def$ **by** $simp-all$
have fb : $-15 * of-nat\ M \leq f\ c \wedge f\ c \leq 15 * of-nat\ M$ **if** c : $c \in \{1..mHalf\ p\}$
for c
proof –
have lb : $ell\ p\ K\ c \leq 2 * (K\ div\ p) + 2$ **using** $ell-bounds[OF\ hf(2)]\ c$ **by** $auto$
have ec : $epsIn\ K\ M\ p\ c = 0 \vee epsIn\ K\ M\ p\ c = 1$ **by** $(rule\ eps-cases)$
have $l2q$: $(of-nat\ (ell\ p\ K\ c) :: rat) \leq 2 * of-nat\ M$ **using** $lb\ hf(3)$ **by** $linarith$
have $Tq1$: $(-1 :: rat) \leq of-int\ (Tin\ K\ M\ p)$ **using** Tl **by** $linarith$
have $Tq2$: $(of-int\ (Tin\ K\ M\ p) :: rat) \leq 4 * of-nat\ M$ **using** Tu **by** $linarith$
have $eq1$: $(0 :: rat) \leq of-int\ (epsIn\ K\ M\ p\ c)$ **using** ec **by** $linarith$


```

have eq2: (of-int (epsIn K M p c) :: rat) ≤ 1 using ec by linarith
have Mq: (40 :: rat) ≤ of-nat M using M by linarith
have l0: (0 :: rat) ≤ of-nat (ell p K c) by simp
have - 15 * of-nat M ≤ of-int (Tin K M p) + of-int (epsIn K M p c) -
(of-nat (ell p K c) +
  4) / (2::rat)
  using Tq1 Tq2 eq1 eq2 l0 l2q Mq by (simp add: field-simps)
moreover have of-int (Tin K M p) + of-int (epsIn K M p c) - (of-nat (ell p
K c) + 4) /
  (2::rat) ≤ 15 * of-nat M
  using Tq1 Tq2 eq1 eq2 l0 l2q Mq by (simp add: field-simps)
ultimately show ?thesis unfolding f-def Zin-def of-int-add by satx
qed
have x0b: - 15 * of-nat M ≤ x0 ∧ x0 ≤ 15 * of-nat M
proof -
  have (of-nat (K div p) :: rat) ≤ of-nat M and (of-nat (Nof K div p) :: rat) ≤
of-nat M
  using hf(3,4) by simp-all
moreover have (of-nat i :: rat) ≤ 4 * of-nat M + 10 using i by linarith
moreover have (40 :: rat) ≤ of-nat M using M by simp
ultimately show ?thesis unfolding x0-def by linarith
qed
have - 15 * of-nat M ≤ Min S
  using fin ne fb x0b unfolding S-def using Min-ge-iff by blast
moreover have Min S ≤ 15 * of-nat M
proof -
  have Min S ≤ x0 using fin unfolding S-def by (simp add: Min-le)
  thus ?thesis using x0b by linarith
qed
ultimately show ?thesis unfolding w by (simp add: abs-le-iff)
qed

```

lemma Z0-bound:

```

assumes p: prime p and h: InnerHyp K M p
shows |2 * (∑ i < 4 * M + 10. wIn K M p 0 i)| ≤ 150 * of-nat M ^ 2
proof -
  have |∑ i < 4 * M + 10. wIn K M p 0 i| ≤ (∑ i <
4 * M + 10. 15 * of-nat M)
  by (rule order-trans[OF sum-abs]) (rule sum-mono, rule w0-bound[OF p h],
simp)
  note hs = this
  have M: (40 :: rat) ≤ of-nat M using h unfolding InnerHyp-def by simp
  have s: (∑ i < 4 * M + 10. (15 :: rat) * of-nat M) = (4 * of-nat M + 10) *
(15 * of-nat M)
  by simp
  have b: |∑ i < 4 * M + 10. wIn K M p 0 i| ≤
(4 * of-nat M + 10) * (15 * of-nat M) using hs s by order
  have 40 * (of-nat M :: rat) ≤ of-nat M ^ 2 using M by (simp add: power2-eq-square)
  moreover have (4 * of-nat M + 10) * (15 * of-nat M) =

```

$60 * \text{of-nat } M \wedge 2 + 150 * (\text{of-nat } M :: \text{rat})$ **by** (*simp add: power2-eq-square algebra-simps*)

moreover have $|2 * (\sum i < 4 * M + 10. \text{wIn } K \ M \ p \ 0 \ i)| = 2 * |\sum i < 4 * M + 10. \text{wIn } K \ M \ p \ 0 \ i|$ **by** (*simp add: abs-mult*)

ultimately show *?thesis* **using** $b \ M$ **by** *linarith*

qed

$p \ \Gamma(K/p)$ in terms of the z -integral of phiZ at $T\text{-f} = \lfloor 2H\text{-c } K/p \rfloor$.

lemma *Gam-eq*:

assumes p : *prime* p **and** h : *InnerHyp* $K \ M \ p$

defines $Tf \equiv \text{floor } (2 * Hc\text{-r} * (\text{real } K / \text{real } p))$

and $s \equiv Hc\text{-r} * (\text{real } K / \text{real } p) - \text{of-int } (\text{floor } (2 * Hc\text{-r} * (\text{real } K / \text{real } p))) / 2$

shows $\text{real } p * \text{Gam-r } (\text{real } K / \text{real } p)$
 $= \text{real } p * \text{interval-lebesgue-integral lborel } (\text{ereal } 0) (\text{ereal } (1 / 2)) (\text{phiZ } p \ (\text{Nof } K) \ K \ Tf)$
 $+ \text{real } p * s * (2 * \text{of-int } Tf - \text{real } (2 * K \text{ div } p) - 4)$
 $- \min (\text{real } p * s) (\text{real } K - \text{real } p * \text{real } (2 * K \text{ div } p) / 2)$

proof –

obtain n **where** K : $K = 40 * n$ **using** h **unfolding** *InnerHyp-def* **by** (*auto elim: dvdE*)

have pr : $0 < \text{real } p$ **using** p **by** (*simp add: prime-gt-0-nat*)

have N : $\text{real } (\text{Nof } K) = 3 * \text{real } n$ **unfolding** *K Nof-def* **by** *simp*

have al : $\alpha\text{-r} * (\text{real } K / \text{real } p) = \text{real } (\text{Nof } K) / \text{real } p$ **unfolding** N **unfolding** *K alpha-r-def* **by** (*simp add: field-simps*)

have q : $\text{floor } (2 * (\text{real } K / \text{real } p)) = \text{int } (2 * K \text{ div } p)$

proof –

have $2 * (\text{real } K / \text{real } p) = \text{real } (2 * K) / \text{real } p$ **by** *simp*

thus *?thesis* **by** (*simp only: floor-nat-div*)

qed

define nn **where** $nn = (2 * (\text{real } K / \text{real } p) - \text{real } (2 * K \text{ div } p)) / 2$

have ph : $(\lambda z. (\text{of-int } Tf - 3 * \text{ell-r } (\text{real } (\text{Nof } K) / \text{real } p) \ z) * (\text{of-int } Tf + 3 * \text{ell-r } (\text{real } (\text{Nof } K) / \text{real } p) \ z - 5))$

$= \text{phiZ } p \ (\text{Nof } K) \ K \ Tf$ **by** (*rule ext*) (*simp add: phiZ-def*)

have G : $\text{Gam-r } (\text{real } K / \text{real } p) = \text{interval-lebesgue-integral lborel } (\text{ereal } 0) (\text{ereal } (1 / 2)) (\text{phiZ } p \ (\text{Nof } K) \ K \ Tf)$

$+ s * (2 * \text{of-int } Tf - \text{real } (2 * K \text{ div } p) - 5) + \text{pos-r } (s - nn)$

unfolding *Gam-r-def* *Let-def* **unfolding** *s-def[symmetric]*

unfolding *Tf-def[symmetric]* $al \ q \ nn\text{-def } ph$ **by** *simp*

have mn : $\text{pos-r } (s - nn) = s - \min s \ nn$ **by** (*simp add: pos-r-def min-def max-def*)

have pm : $\text{real } p * \min s \ nn = \min (\text{real } p * s) (\text{real } p * nn)$ **using** pr **by** (*simp add: min-def mult-left-mono*)

have pn : $\text{real } p * nn = \text{real } K - \text{real } p * \text{real } (2 * K \text{ div } p) / 2$ **unfolding** *nn-def* **using** pr

by (*simp add: field-simps*)

show *?thesis* **unfolding** $G \ mn$ **using** $pm \ pn$ **using** *algebra-simps* **by** *argo*

qed

25.5 Discrete sum versus the integral

lemma *sum-Icc-shift*: $(\sum a = 1..m. f a) = (\sum a < (m :: nat). f (a + 1))$

proof (*induction m*)

case 0

show ?case by simp

next

case (Suc m)

show ?case using Suc by (simp add: sum.cl-ivl-Suc)

qed

lemma *ellZ-bounds*:

assumes $0 < p$ $a < p$

shows $0 \leq \text{ellZ } p \ A \ a \wedge \text{ellZ } p \ A \ a \leq 2 * \text{int } (A \text{ div } p) + 2$

proof –

have *ip*: $0 < \text{int } p$ using *assms*(1) by simp

have $-1 \leq (\text{int } A - \text{int } a - 1) \text{ div } \text{int } p$

proof –

have $(-\text{int } p) \text{ div } \text{int } p \leq (\text{int } A - \text{int } a - 1) \text{ div } \text{int } p$ using *assms ip*

by (*intro zdiv-mono1*) *simp-all*

moreover have $(-\text{int } p) \text{ div } \text{int } p = -1$ using *ip* by (simp add: *zdiv-zminus1-eq-if*)

ultimately show ?thesis by *linarith*

qed

moreover have $(\text{int } A - \text{int } a - 1) \text{ div } \text{int } p \leq \text{int } A \text{ div } \text{int } p$ using *ip*

by (*intro zdiv-mono1*) *simp-all*

moreover have $(\text{int } A + \text{int } a) \text{ div } \text{int } p \leq \text{int } A \text{ div } \text{int } p + 1$

proof –

have $(\text{int } A + \text{int } a) \text{ div } \text{int } p \leq (\text{int } A + \text{int } p) \text{ div } \text{int } p$ using *assms ip*

by (*intro zdiv-mono1*) *simp-all*

thus ?thesis using *ip* by simp

qed

moreover have $0 \leq (\text{int } A + \text{int } a) \text{ div } \text{int } p$ using *ip* by (simp add: *pos-imp-zdiv-nonneg-iff*)

moreover have $\text{int } A \text{ div } \text{int } p = \text{int } (A \text{ div } p)$ by (simp add: *zdiv-int*)

ultimately show ?thesis unfolding *ellZ-def* by *linarith*

qed

lemma *disc-01*:

assumes $0 < p$

shows $(\text{int } A + \text{int } a + 1) \text{ div } \text{int } p - (\text{int } A + \text{int } a) \text{ div } \text{int } p = 0 \vee$

$(\text{int } A + \text{int } a + 1) \text{ div } \text{int } p - (\text{int } A + \text{int } a) \text{ div } \text{int } p = 1$

proof –

have *ip*: $0 < \text{int } p$ using *assms*(1) by simp

have $(\text{int } A + \text{int } a) \text{ div } \text{int } p \leq (\text{int } A + \text{int } a + 1) \text{ div } \text{int } p$ using *ip*

by (*intro zdiv-mono1*) *simp-all*

moreover have $(\text{int } A + \text{int } a + 1) \text{ div } \text{int } p \leq (\text{int } A + \text{int } a) \text{ div } \text{int } p + 1$

proof –

have $(\text{int } A + \text{int } a + 1) \text{ div } \text{int } p \leq (\text{int } A + \text{int } a + \text{int } p) \text{ div } \text{int } p$ using

```

assms ip
  by (intro zdiv-mono1) simp-all
  thus ?thesis using ip by simp
qed
ultimately show ?thesis by linarith
qed

lemma gg-lip:
  assumes  $40 \leq M - 2 \leq T$   $T \leq 4 * \text{int } M + 1$   $0 \leq v$   $v \leq 2 * \text{int } M$ 
     $0 \leq l$   $l \leq 2 * \text{int } M$ 
     $dv = 0 \vee dv = 1$   $dl = 0 \vee dl = 1$ 
  shows  $|gg \ T \ (v + dv) \ (l + dl) - gg \ T \ v \ l| \leq 100 * \text{int } M * (dv + dl)$ 
proof -
  have d1:  $gg \ T \ v \ (l + 1) - gg \ T \ v \ l = 3 * v - T$  by (simp add: gg-def algebra-simps)
  have d2:  $gg \ T \ (v + 1) \ l - gg \ T \ v \ l = 3 * l - 18 * v + 6$  by (simp add: gg-def algebra-simps)
  have d3:  $gg \ T \ (v + 1) \ (l + 1) - gg \ T \ v \ l = 3 * l - 15 * v - T + 9$ 
    by (simp add: gg-def algebra-simps)
  consider  $dv = 0 \ dl = 0 \mid dv = 0 \ dl = 1 \mid dv = 1 \ dl = 0 \mid dv = 1 \ dl = 1$ 
    using assms(8,9) by blast
  then show ?thesis
  proof cases
    case 1 thus ?thesis by simp
  next
    case 2 thus ?thesis using d1 assms(1-7) by (simp add: abs-le-iff)
  next
    case 3 thus ?thesis using d2 assms(1-7) by (simp add: abs-le-iff)
  next
    case 4 thus ?thesis using d3 assms(1-7) by (simp add: abs-le-iff)
  qed
qed

```

```

lemma gg-bound:
  assumes  $40 \leq M - 2 \leq T$   $T \leq 4 * \text{int } M + 1$   $0 \leq v$   $v \leq 2 * \text{int } M$ 
     $0 \leq l$   $l \leq 2 * \text{int } M$ 
  shows  $|gg \ T \ v \ l| \leq 150 * \text{int } M^2$ 
proof -
  have  $|T - 3 * v| \leq 11 * \text{int } M$  and  $|T + 3 * v - l - 5| \leq 13 * \text{int } M$ 
    using assms by (auto simp: abs-le-iff)
  hence  $|gg \ T \ v \ l| \leq (11 * \text{int } M) * (13 * \text{int } M)$  unfolding gg-def abs-mult
    by (rule mult-mono) simp-all
  thus ?thesis by (simp add: power2-eq-square)
qed

```

$|S-d(T) - p \int -0^{1/2} \varphi-T| \leq 100 M^2$: per-cell comparison plus the last half cell.

```

lemma S-vs-I:
  assumes  $p$ :  $p$  prime  $p$  and  $h$ :  $\text{InnerHyp } K \ M \ p$  and  $T$ :  $-2 \leq T$   $T \leq 4 * \text{int } M$ 

```

```

+ 1
  shows |of-int (Sd K p T) - real p * interval-lebesgue-integral lborel (ereal 0)
(ereal (1 /
  2)) (phiZ p (Nof K) K T)| ≤ 100 * real M ^ 2
proof -
  have facts: 7 ≤ p odd p 2 * mHalf p + 1 = p K div p < M Nof K div p < M
  using hyp-facts[OF p h] by auto
  have M: 40 ≤ M using h unfolding InnerHyp-def by simp
  define m where m = mHalf p
  have p0: 0 < p using facts by simp
  have mp: m < p unfolding m-def using facts by simp
  have I: real p * interval-lebesgue-integral lborel (ereal 0) (ereal (1 / 2)) (phiZ p
(Nof K) K T)
    = (∑ a < m. of-int (gg T (ellZ p (Nof K) a) (ellZ p K a))) +
      of-int (gg T (ellZ p (Nof K) m) (ellZ p K m)) / 2
  unfolding m-def by (rule integral-sum) (use facts in simp)
  have S: Sd K p T = (∑ a < m. gg T (int (ell p (Nof K) (a + 1))) (int (ell p
K (a + 1))))
  unfolding Sd-def m-def by (rule sum-Icc-shift)
  have cell: |gg T (int (ell p (Nof K) (a + 1))) (int (ell p K (a + 1))) -
gg T (ellZ p (Nof K) a) (ellZ p K a)|
    ≤ 100 * int M * (((int (Nof K) + int a + 1) div int p - (int (Nof
K) +
      int a) div int p) + ((int K + int a + 1) div int p - (int K + int a)
div int p))
  if a: a < m for a
proof -
  have a1: a + 1 ≤ mHalf p using a unfolding m-def by simp
  have ap: a < p using a mp by simp
  have eN: int (ell p (Nof K) (a + 1)) =
    ellZ p (Nof K) a + ((int (Nof K) + int a + 1) div int p - (int (Nof K) +
int a) div int p)
  by (rule ell-succ-eq[OF facts(2) a1])
  have eK: int (ell p K (a + 1)) = ellZ p K a + ((int K + int a + 1) div int p
- (int K +
    int a) div int p)
  by (rule ell-succ-eq[OF facts(2) a1])
  have bN: 0 ≤ ellZ p (Nof K) a ∧ ellZ p (Nof K) a ≤ 2 * int (Nof K div p)
+ 2
  by (rule ellZ-bounds[OF p0 ap])
  have bK: 0 ≤ ellZ p K a ∧ ellZ p K a ≤ 2 * int (K div p) + 2
  by (rule ellZ-bounds[OF p0 ap])
  have nN: 2 * int (Nof K div p) + 2 ≤ 2 * int M using facts(5) by simp
  have nK: 2 * int (K div p) + 2 ≤ 2 * int M using facts(4) by simp
  show ?thesis unfolding eN eK
proof (rule gg-lip[OF M T])
  show 0 ≤ ellZ p (Nof K) a using bN by simp
  show ellZ p (Nof K) a ≤ 2 * int M using bN nN by linarith
  show 0 ≤ ellZ p K a using bK by simp

```

```

show  $\text{ellZ } p \ K \ a \leq 2 * \text{int } M$  using  $bK \ nK$  by linarith
show  $(\text{int } (\text{Nof } K) + \text{int } a + 1) \text{ div int } p - (\text{int } (\text{Nof } K) + \text{int } a) \text{ div int } p$ 
 $= 0 \vee$ 
 $(\text{int } (\text{Nof } K) + \text{int } a + 1) \text{ div int } p - (\text{int } (\text{Nof } K) + \text{int } a) \text{ div int } p =$ 
 $1$ 
by (rule disc-01[OF p0])
show  $(\text{int } K + \text{int } a + 1) \text{ div int } p - (\text{int } K + \text{int } a) \text{ div int } p = 0 \vee$ 
 $(\text{int } K + \text{int } a + 1) \text{ div int } p - (\text{int } K + \text{int } a) \text{ div int } p = 1$  by (rule
disc-01[OF p0])
qed
qed
have diff:  $|\text{Sd } K \ p \ T - (\sum a < m. \text{gg } T (\text{ellZ } p (\text{Nof } K) \ a) (\text{ellZ } p \ K \ a))| \leq$ 
 $200 * \text{int } M$ 
proof -
define dN where  $dN \ a = (\text{int } (\text{Nof } K) + \text{int } a + 1) \text{ div int } p - (\text{int } (\text{Nof } K)$ 
 $+ \text{int } a) \text{ div int } p$ 
for a
define dK where  $dK \ a = (\text{int } K + \text{int } a + 1) \text{ div int } p - (\text{int } K + \text{int } a) \text{ div}$ 
 $\text{int } p$  for a
have  $|\text{Sd } K \ p \ T - (\sum a < m. \text{gg } T (\text{ellZ } p (\text{Nof } K) \ a) (\text{ellZ } p \ K \ a))|$ 
 $= |\sum a < m. (\text{gg } T (\text{int } (\text{ell } p (\text{Nof } K) (a + 1))) (\text{int } (\text{ell } p \ K (a + 1))) -$ 
 $\text{gg } T (\text{ellZ } p (\text{Nof } K) \ a) (\text{ellZ } p \ K \ a))|$ 
unfolding S sum-subtractf ..
also have  $\dots \leq (\sum a <$ 
 $m. |\text{gg } T (\text{int } (\text{ell } p (\text{Nof } K) (a + 1))) (\text{int } (\text{ell } p \ K (a + 1))) -$ 
 $\text{gg } T (\text{ellZ } p (\text{Nof } K) \ a) (\text{ellZ } p \ K \ a)|)$ 
by (rule sum-abs)
also have  $\dots \leq (\sum a < m. 100 * \text{int } M * (dN \ a + dK \ a))$ 
unfolding dN-def dK-def by (rule sum-mono) (rule cell, simp)
also have  $\dots = 100 * \text{int } M * ((\sum a < m. dN \ a) + (\sum a < m. dK \ a))$ 
by (simp add: distrib-left sum.distrib sum-distrib-left)
also have  $\dots \leq 100 * \text{int } M * (1 + 1)$ 
proof -
have  $(\sum a < m. dN \ a) \leq 1$  unfolding dN-def by (rule sum-disc(1)[OF p0
mp])
moreover have  $(\sum a < m. dK \ a) \leq 1$  unfolding dK-def by (rule
sum-disc(1)[OF p0 mp])
ultimately show ?thesis by (intro mult-left-mono) simp-all
qed
finally show ?thesis by simp
qed
have last:  $|\text{gg } T (\text{ellZ } p (\text{Nof } K) \ m) (\text{ellZ } p \ K \ m)| \leq 150 * \text{int } M \wedge 2$ 
proof -
have bN:  $0 \leq \text{ellZ } p (\text{Nof } K) \ m \wedge \text{ellZ } p (\text{Nof } K) \ m \leq 2 * \text{int } (\text{Nof } K \text{ div } p)$ 
 $+ 2$ 
by (rule ellZ-bounds[OF p0 mp])
have bK:  $0 \leq \text{ellZ } p \ K \ m \wedge \text{ellZ } p \ K \ m \leq 2 * \text{int } (K \text{ div } p) + 2$ 
by (rule ellZ-bounds[OF p0 mp])
have nN:  $2 * \text{int } (\text{Nof } K \text{ div } p) + 2 \leq 2 * \text{int } M$  using facts(5) by simp

```

have $nK: 2 * \text{int } (K \text{ div } p) + 2 \leq 2 * \text{int } M$ **using** *facts(4)* **by** *simp*
show *?thesis* **by** (*rule gg-bound[OF M T]*) (*use bN bK nN nK in linarith*) +
qed
have $|\text{of-int } (Sd K p T) - (\sum a <$
 $m. (\text{of-int } (gg T (\text{ellZ } p (\text{Nof } K) a) (\text{ellZ } p K a)) :: \text{real})| \leq 200 * \text{real } M$
proof –
have $\text{real-of-int } |Sd K p T - (\sum a <$
 $m. gg T (\text{ellZ } p (\text{Nof } K) a) (\text{ellZ } p K a))| \leq \text{real-of-int } (200 * \text{int } M)$
using *diff* **by** (*simp only: of-int-le-iff*)
thus *?thesis* **by** (*simp add: of-int-sum*)
qed
moreover **have** $|\text{of-int } (gg T (\text{ellZ } p (\text{Nof } K) m) (\text{ellZ } p K m)) :: \text{real}| \leq$
 $150 * \text{real } M ^ 2$
proof –
have $\text{real-of-int } |gg T (\text{ellZ } p (\text{Nof } K) m) (\text{ellZ } p K m)| \leq$
 $\text{real-of-int } (150 * \text{int } M ^ 2)$
using *last* **by** (*simp only: of-int-le-iff*)
thus *?thesis* **by** *simp*
qed
moreover **have** $40 \leq \text{real } M$ **using** *M* **by** *simp*
hence $200 * \text{real } M \leq 25 * \text{real } M ^ 2$ **by** (*simp add: power2-eq-square*)
ultimately show *?thesis* **unfolding** *I* **using** *abs-le-iff* **by** *linarith*
qed

25.6 The comparison functions

definition $Dd :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{int} \Rightarrow \text{real} \Rightarrow \text{real}$

where

$Dd K p T E = \text{of-int } (Sd K p T) + E * (2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) -$
 $4) -$
 $\min E (\text{real } (c1 K p))$

definition $Cc :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{int} \Rightarrow \text{real} \Rightarrow \text{real}$

where

$Cc K p T \sigma = \text{real } p * \text{interval-lebesgue-integral lborel } (\text{ereal } 0) (\text{ereal } (1 /$
 $2)) (\text{phiZ } p (\text{Nof } K) K T)$
 $+ \sigma * (2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) - 4) - \min \sigma (\text{real } K -$
 $\text{real } p * \text{real } (2 * K \text{ div } p) / 2)$

lemma *Dd-trans*:

assumes *p*: *prime p* **and** *h*: *InnerHyp K M p*

shows $Dd K p T (\text{real } (mHalf p)) = Dd K p (T + 1) 0$

proof –

have $s: (\sum a = 1..mHalf p. \text{ell } p K a) = mHalf p * (2 * K \text{ div } p) + c1 K p$
by (*rule sum-ellK[OF p h]*)

have $c: c1 K p \leq mHalf p$ **by** (*rule c1-le*)

have $e: Sd K p (T + 1) = Sd K p T + \text{int } (mHalf p) * (2 * T - 4) - \text{int}$
 $(mHalf p * (2 * K \text{ div } p) +$
 $c1 K p)$

```

proof –
  have  $g: gg\ (T + 1)\ v\ l = gg\ T\ v\ l + (2 * T - 4) - l$  for  $v\ l$ 
    by (simp add: gg-def algebra-simps)
    show ?thesis unfolding Sd-def s[symmetric]  $g$  by (simp add: sum.distrib
sum-subtractf of-nat-sum)
  qed
  show ?thesis
    unfolding Dd-def e using  $c$  by (simp add: min-def algebra-simps)
qed

```

```

lemma q-le:
  assumes  $p$ : prime p and  $h$ : InnerHyp K M p
  shows  $2 * K \text{ div } p \leq 2 * M$ 
proof –
  have  $KpM$ :  $K < p * M$  using  $h$  unfolding InnerHyp-def by simp
  have  $p0$ :  $0 < p$  using prime-gt-0-nat[OF p] .
  have  $2 * K < (2 * M) * p$  using  $KpM$  by (simp add: algebra-simps)
  hence  $2 * K \text{ div } p < 2 * M$  using  $p0$  by (simp add: div-less-iff-less-mult)
  thus ?thesis by simp
qed

```

```

lemma slope-bound:
  assumes  $p$ : prime p and  $h$ : InnerHyp K M p and  $T$ :  $-2 \leq T \leq 4 * \text{int } M + 1$ 
  shows  $|2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) - 4| \leq 11 * \text{real } M$ 
  using q-le[OF p h]  $T\ h$  unfolding InnerHyp-def by (auto simp: abs-le-iff)

```

```

lemma Dd-lip:
  assumes  $p$ : prime p and  $h$ : InnerHyp K M p and  $T$ :  $-2 \leq T \leq 4 * \text{int } M + 1$ 
  shows  $|Dd\ K\ p\ T\ E1 - Dd\ K\ p\ T\ E2| \leq |E1 - E2| * (11 * \text{real } M + 1)$ 
proof –
  have  $s$ :  $|2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) - 4| \leq 11 * \text{real } M$ 
    by (rule slope-bound[OF p h T])
  have  $m$ :  $|\min E1\ (\text{real } (c1\ K\ p)) - \min E2\ (\text{real } (c1\ K\ p))| \leq |E1 - E2|$ 
    using min-def abs-le-iff by linarith
  have  $e$ :  $Dd\ K\ p\ T\ E1 - Dd\ K\ p\ T\ E2 = (E1 - E2) * (2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) - 4) -$ 
     $(\min E1\ (\text{real } (c1\ K\ p)) - \min E2\ (\text{real } (c1\ K\ p)))$ 
    unfolding Dd-def by (simp add: algebra-simps)
  have  $| (E1 - E2) * (2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) - 4) | \leq$ 
     $|E1 - E2| * (11 * \text{real } M)$ 
    unfolding abs-mult by (rule mult-left-mono[OF s]) simp
  thus ?thesis unfolding  $e$  using  $m$  abs-triangle-ineq4  $[of\ (E1 - E2) * (2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) - 4)]$ 
    using algebra-simps by
argo
qed

```


lemma *c1-close*:

assumes *p*: prime *p* **and** *h*: InnerHyp *K M p*
shows $|\text{real } (c1 \ K \ p) - (\text{real } K - \text{real } p * \text{real } (2 * K \text{ div } p) / 2)| \leq 1$
proof –
have *facts*: $2 * mHalf \ p + 1 = p \wedge \leq p$ **using** *hyp-facts*[*OF p h*] **by** *auto*
have *s*: $mHalf \ p * (2 * K \text{ div } p) + c1 \ K \ p = K - K \text{ div } p$
using *sum-ellK*[*OF p h*] *sum-ellK'*[*OF p h*] **by** *simp*
have *q*: $2 * (K \text{ div } p) \leq 2 * K \text{ div } p \leq 2 * K \text{ div } p \leq 2 * (K \text{ div } p) + 1$
proof –
have *p0*: $0 < p$ **using** *facts* **by** *simp*
have *g1*: $K \text{ div } p * p \leq K$ **by** *fastforce*
have *g2*: $K < (K \text{ div } p + 1) * p$
proof –
have *e*: $(K \text{ div } p + 1) * p = K \text{ div } p * p + p$ **by** (*simp add: algebra-simps*)
show ?thesis **unfolding** *e* **using** *mod-less-divisor*[*OF p0, of K*] *div-mult-mod-eq*[*of K p*]
by *linarith*
qed
show $2 * (K \text{ div } p) \leq 2 * K \text{ div } p$ **using** *g1 p0* **by** (*simp add: less-eq-div-iff-mult-less-eq*)
have $2 * K < (2 * (K \text{ div } p) + 2) * p$ **using** *g2* **by** (*simp add: algebra-simps*)
hence $2 * K \text{ div } p < 2 * (K \text{ div } p) + 2$ **using** *p0* **by** (*simp add: div-less-iff-less-mult*)
thus $2 * K \text{ div } p \leq 2 * (K \text{ div } p) + 1$ **by** *simp*
qed
have *kk*: $K \text{ div } p \leq K$ **by** (*rule div-le-dividend*)
have *c*: $\text{real } (c1 \ K \ p) = \text{real } K - \text{real } (K \text{ div } p) - \text{real } (mHalf \ p) * \text{real } (2 * K \text{ div } p)$
proof –
have *c0*: $c1 \ K \ p = (K - K \text{ div } p) - mHalf \ p * (2 * K \text{ div } p)$ **using** *s* **by** *linarith*
have *le*: $mHalf \ p * (2 * K \text{ div } p) \leq K - K \text{ div } p$ **using** *s* **by** *linarith*
show ?thesis **unfolding** *c0* *of-nat-diff*[*OF le*] *of-nat-diff*[*OF kk*] **by** *simp*
qed
have *pr*: $\text{real } p = 2 * \text{real } (mHalf \ p) + 1$ **using** *facts*(1)
by (*metis of-nat-1 of-nat-add of-nat-mult of-nat-numeral*)
have $2 * \text{real } (K \text{ div } p) \leq \text{real } (2 * K \text{ div } p) \leq \text{real } (2 * K \text{ div } p) \leq 2 * \text{real } (K \text{ div } p) + 1$ **using** *q* **by** *simp-all*
thus ?thesis **unfolding** *c pr* **by** (*simp add: abs-le-iff algebra-simps*)
qed

lemma *Dd-Cc*:

assumes *p*: prime *p* **and** *h*: InnerHyp *K M p* **and** *T*: $-2 \leq T \leq 4 * \text{int } M + 1$
shows $|Dd \ K \ p \ T \ E - Cc \ K \ p \ T \ \sigma| \leq 100 * \text{real } M \wedge 2 + |E - \sigma| * (11 * \text{real } M + 1) + 1$
proof –
have *s*: $|2 * \text{of-int } T - \text{real } (2 * K \text{ div } p) - 4| \leq 11 * \text{real } M$
by (*rule slope-bound*[*OF p h T*])
have *SI*: $|\text{of-int } (Sd \ K \ p \ T) - \text{real } p * \text{real } (2 * K \text{ div } p) - 4| \leq 11 * \text{real } M$

$$\begin{aligned} & \text{interval-lebesgue-integral lborel (ereal 0) (ereal (1 / 2)) (phiZ p (Nof K) K T)} \\ \leq & 100 * \text{real } M \wedge 2 \\ & \text{by (rule S-vs-I[OF p h T])} \\ & \text{have } |\text{real (c1 K p)} - (\text{real K} - \text{real p} * \text{real (2 * K div p)} / 2)| \leq 1 \\ & \text{by (rule c1-close[OF p h])} \\ & \text{hence m: } |\min E (\text{real (c1 K p)}) - \min \sigma (\text{real K} - \text{real p} * \text{real (2 * K div p)} \\ & / \\ & 2)| \leq |E - \sigma| + 1 \\ & \text{using min-def abs-le-iff by argo} \\ & \text{have e: Dd K p T E - Cc K p T } \sigma \\ & = (\text{of-int (Sd K p T)} - \text{real p} * \text{interval-lebesgue-integral lborel (ereal 0) (ereal} \\ & (1 / \\ & 2)) (\text{phiZ p (Nof K) K T})) \\ & + (E - \sigma) * (2 * \text{of-int T} - \text{real (2 * K div p)} - 4) \\ & - (\min E (\text{real (c1 K p)}) - \min \sigma (\text{real K} - \text{real p} * \text{real (2 * K div p)} / \\ & 2)) \\ & \text{unfolding Dd-def Cc-def by (simp add: algebra-simps)} \\ & \text{have a: } |(E - \sigma) * (2 * \text{of-int T} - \text{real (2 * K div p)} - 4)| \leq \\ & |E - \sigma| * (11 * \text{real M}) \\ & \text{unfolding abs-mult by (rule mult-left-mono[OF s]) simp} \\ & \text{have M0: } 0 \leq |E - \sigma| \text{ by simp} \\ & \text{show ?thesis unfolding e using a m SI M0 by (simp add: abs-le-iff alge-} \\ & \text{bra-simps)} \\ & \text{qed} \end{aligned}$$

25.7 Assembly

lemma *gamma-real*:

assumes p : prime p **and** h : InnerHyp K M p
shows $\text{of-rat (gammaIn K M p)} = \text{of-rat (2 * (\sum i <$
 $4 * M + 10. \text{wIn K M p 0 i}) + \text{Dd K p (Tin K M p) (of-int (Ein K M p))}$
unfolding *gamma-decomp*[OF p h] *Dd-def of-rat-add of-rat-of-int-eq*
by (*simp add: of-int-min*)

The fill relation: $mT + E = m \cdot 2H\text{-c } K/p + \Delta$ with $|\Delta| \leq 8M$.

lemma *fill-rel*:

assumes p : prime p **and** h : InnerHyp K M p
shows $|(\text{real (mHalf p)} * \text{of-int (Tin K M p)} + \text{of-int (Ein K M p)}) - \text{real (mHalf}$
 $p) * (2 * \text{Hc-r} * (\text{real K} / \text{real p}))| \leq 8 * \text{real M}$
proof –
have *facts*: $7 \leq p \cdot 2 * \text{mHalf p} + 1 = p \text{ Nof K div p} < M$ **using** *hyp-facts*[OF
 p h] **by** *auto*
have *TE*: $\text{int (mHalf p)} * \text{Tin K M p} + \text{Ein K M p} = \text{rhs44 K M p}$ **using**
 TE-spec [OF p h] **by** *simp*
obtain n **where** K : $K = 40 * n$ **using** h **unfolding** *InnerHyp-def* **by** (*auto*
 elim: dvdE)
have M : $40 \leq M$ **and** KpM : $K < p * M$ **using** h **unfolding** *InnerHyp-def* **by**

auto
have *hh*: *hof* *K* = $37 * n$ **and** *hN*: *Nof* *K* = $3 * n$ **unfolding** *K* *hof-def* *Nof-def*
by *simp-all*
have *pr*: $\text{real } p = 2 * \text{real } (mHalf\ p) + 1$ **using** *facts*(2)
by (*metis of-nat-1 of-nat-add of-nat-mult of-nat-numeral*)
have *p0*: $0 < \text{real } p$ **using** *facts*(1) **by** *simp*
have *lhs*: $\text{real } (mHalf\ p) * \text{of-int } (Tin\ K\ M\ p) + \text{of-int } (Ein\ K\ M\ p) =$
 $37 * \text{real } n - (4 * \text{real } M + 10) + 3 * (3 * \text{real } n - \text{real } (3 * n\ \text{div } p))$
proof –
have $\text{real-of-int } (int\ (mHalf\ p) * Tin\ K\ M\ p + Ein\ K\ M\ p) = \text{real-of-int } (rhs44\ K\ M\ p)$ **using** *TE*
by *simp*
thus *?thesis* **unfolding** *rhs44-def* *L0-def* *hh* *hN* **by** *simp*
qed
have *mv*: $\text{real } (mHalf\ p) * (2 * Hc-r * (\text{real } K / \text{real } p)) = 46 * \text{real } n - 46 * \text{real } n / \text{real } p$
unfolding *K* *Hc-r-def* **using** *pr* *p0* **by** (*simp add: field-simps*)
have $46 * \text{real } n / \text{real } p \leq 46 * \text{real } M / 40$
proof –
have $\text{real } (40 * n) < \text{real } (p * M)$ **using** *KpM* **unfolding** *K* **by** (*simp only: of-nat-less-iff*)
hence $40 * \text{real } n < \text{real } p * \text{real } M$ **by** *simp*
thus *?thesis* **using** *p0* **by** (*simp add: field-simps*)
qed
moreover **have** $0 \leq 46 * \text{real } n / \text{real } p$ **using** *p0* **by** *simp*
moreover **have** $\text{real } (3 * n\ \text{div } p) \leq \text{real } M$ **and** $0 \leq \text{real } (3 * n\ \text{div } p)$
using *facts*(3) *hN* **by** *simp-all*
moreover **have** $(40 :: \text{real}) \leq \text{real } M$ **using** *M* **by** *linarith*
ultimately show *?thesis*
unfolding *lhs* *mv* **using** *p0*
using *abs-le-iff* **by** *argo*
qed

lemma *m-large*:

assumes *p*: *prime* *p* **and** *h*: *InnerHyp* *K* *M* *p*
shows $100 * \text{real } M \leq \text{real } (mHalf\ p)$
proof –
have $200 * M < p$ **using** *innerHyp-bounds[OF h]* **by** *simp*
moreover **have** $2 * mHalf\ p + 1 = p$ **using** *hyp-facts[OF p h]* **by** *simp*
ultimately have $100 * M \leq mHalf\ p$ **by** *linarith*
thus *?thesis* **by** (*simp only: of-nat-le-iff[symmetric] of-nat-mult of-nat-numeral*)
qed

$|\gamma\text{-in} - p\ \Gamma(K/p)| \leq 1000\ M^2$: *T-f* $\in \{T - 1, T, T + 1\}$ and three cases.

theorem *gamma-bound*:

assumes *p*: *prime* *p* **and** *h*: *InnerHyp* *K* *M* *p*
shows $|\text{of-rat } (gammaIn\ K\ M\ p) - \text{real } p * Gam-r\ (\text{real } K / \text{real } p)| \leq$
 $1000 * \text{real } M^2$
proof –

```

have facts:  $7 \leq p^2 * mHalf\ p + 1 = p$  using hyp-facts[OF p h] by auto
have M:  $40 \leq M$  using h unfolding InnerHyp-def by simp
have Mr:  $40 \leq real\ M$  using M by simp
have M2:  $40 * real\ M \leq real\ M^2$  using Mr by (simp add: power2-eq-square)
have Tl:  $-1 \leq Tin\ K\ M\ p$  by (rule Tin-ge[OF p h])
have Tu:  $Tin\ K\ M\ p \leq 4 * int\ M$  by (rule Tin-le[OF p h])
have E:  $0 \leq Ein\ K\ M\ p$   $Ein\ K\ M\ p < int\ (mHalf\ p)$  using TE-spec[OF p h] by
auto
have Z:  $|of-rat\ (2 * (\sum i < 4 * M + 10. wIn\ K\ M\ p\ 0\ i))| \leq 150 * real\ M^2$ 
proof -
  have of-rat  $|2 * (\sum i < 4 * M + 10. wIn\ K\ M\ p\ 0\ i)| \leq$ 
    (of-rat  $(150 * of-nat\ M^2) :: real$ )
    using Z0-bound[OF p h] by (simp only: of-rat-less-eq)
  note hz = this
  have e:  $(of-rat\ (150 * of-nat\ M^2) :: real) = 150 * real\ M^2$ 
    by (simp add: of-rat-mult of-rat-power)
  have e2:  $|(of-rat\ (2 * (\sum i < 4 * M + 10. wIn\ K\ M\ p\ 0\ i)) :: real)| =$ 
    of-rat  $|2 * (\sum i < 4 * M + 10. wIn\ K\ M\ p\ 0\ i)|$  by (metis abs-of-rat)
  show ?thesis using hz unfolding e e2 by satx
qed
define m where  $m = real\ (mHalf\ p)$ 
define v where  $v = 2 * Hc-r * (real\ K / real\ p)$ 
define T where  $T = Tin\ K\ M\ p$ 
define E' where  $E' = (of-int\ (Ein\ K\ M\ p) :: real)$ 
define Tf where  $Tf = floor\ v$ 
define  $\sigma$  where  $\sigma = (2 * m + 1) * (v - of-int\ Tf) / 2$ 
define Zr :: real where  $Zr = of-rat\ (2 * (\sum i < 4 * M + 10. wIn\ K\ M\ p\ 0\ i))$ 
have pr:  $real\ p = 2 * m + 1$  unfolding m-def using facts(2)
  by (metis of-nat-1 of-nat-add of-nat-mult of-nat-numeral)
have G:  $real\ p * Gam-r\ (real\ K / real\ p) = Cc\ K\ p\ Tf\ \sigma$ 
proof -
  have g:  $real\ p * Gam-r\ (real\ K / real\ p)$ 
    =  $real\ p * interval-lebesgue-integral\ lborel\ (ereal\ 0)\ (ereal\ (1 /$ 
       $2))\ (phiZ\ p\ (Nof\ K)\ K\ Tf)$ 
    +  $real\ p * (Hc-r * (real\ K / real\ p) - of-int\ Tf / 2) * (2 * of-int\ Tf -$ 
 $real\ (2 *$ 
       $K\ div\ p) - 4)$ 
    -  $min\ (real\ p * (Hc-r * (real\ K / real\ p) - of-int\ Tf / 2))\ (real\ K -$ 
 $real\ p *$ 
       $real\ (2 * K\ div\ p) / 2)$ 
    using Gam-eq[OF p h] unfolding Tf-def v-def .
  have ps:  $real\ p * (Hc-r * (real\ K / real\ p) - of-int\ Tf / 2) = \sigma$ 
    unfolding  $\sigma$ -def v-def pr[symmetric] by (simp add: algebra-simps)
  show ?thesis unfolding g ps Cc-def ..
qed
have decomp:  $of-rat\ (gammaIn\ K\ M\ p) - real\ p * Gam-r\ (real\ K / real\ p) =$ 
  Zr +  $(Dd\ K\ p\ T\ E' - Cc\ K\ p\ Tf\ \sigma)$ 
  unfolding G Zr-def T-def E'-def gamma-real[OF p h] by simp
have fill:  $|(m * of-int\ T + E') - m * v| \leq 8 * real\ M$ 

```

```

  unfolding m-def T-def E'-def v-def by (rule fill-rel[OF p h])
  have ml:  $100 * \text{real } M \leq m$  unfolding m-def by (rule m-large[OF p h])
  have E0:  $0 \leq E'$  and Em:  $E' < m$  unfolding E'-def m-def using E by simp-all
  have f1:  $\text{of-int } Tf \leq v$  and f2:  $v < \text{of-int } Tf + 1$  unfolding Tf-def by simp-all
  have Tf-range:  $Tf = T \vee Tf = T + 1 \vee Tf = T - 1$ 
  proof -
    have fl1:  $m * (\text{of-int } T - v) + E' \leq 8 * \text{real } M$ 
      using fill by (simp add: abs-le-iff algebra-simps)
    have fl2:  $-(8 * \text{real } M) \leq m * (\text{of-int } T - v) + E'$ 
      using fill by (simp add: abs-le-iff algebra-simps)
    have m0:  $0 \leq m$  using ml Mr by linarith
    have Tf  $\leq T + 1$ 
    proof (rule ccontr)
      assume  $\neg Tf \leq T + 1$ 
      hence  $T + 2 \leq Tf$  by simp
      hence  $\text{real-of-int } (T + 2) \leq \text{of-int } Tf$  by (simp only: of-int-le-iff)
      hence  $\text{real-of-int } T + 2 \leq \text{of-int } Tf$  by simp
      hence  $2 \leq v - \text{of-int } T$  using f1 by linarith
      hence  $m * 2 \leq m * (v - \text{of-int } T)$  by (rule mult-left-mono[OF - m0])
      thus False using fl2 Em ml Mr by (simp add: algebra-simps)
    qed
    moreover have  $T - 1 \leq Tf$ 
    proof (rule ccontr)
      assume  $\neg T - 1 \leq Tf$ 
      hence  $Tf \leq T - 2$  by simp
      hence  $\text{real-of-int } Tf \leq \text{of-int } (T - 2)$  by (simp only: of-int-le-iff)
      hence  $\text{real-of-int } Tf \leq \text{of-int } T - 2$  by simp
      hence  $1 \leq \text{of-int } T - v$  using f2 by linarith
      hence  $m * 1 \leq m * (\text{of-int } T - v)$  by (rule mult-left-mono[OF - m0])
      thus False using fl1 E0 ml Mr by (simp add: algebra-simps)
    qed
    ultimately show ?thesis by linarith
  qed
  have main:  $|Dd \ K \ p \ T \ E' - Cc \ K \ p \ Tf \ \sigma| \leq 850 * \text{real } M \wedge 2$ 
    using Tf-range
  proof (elim disjE)
    assume e:  $Tf = T$ 
    have c:  $|Dd \ K \ p \ T \ E' - Cc \ K \ p \ T \ \sigma| \leq$ 
       $100 * \text{real } M \wedge 2 + |E' - \sigma| * (11 * \text{real } M + 1) + 1$ 
      by (rule Dd-Cc[OF p h]) (use Tl Tu T-def in linarith)+
    have Es:  $|E' - \sigma| \leq 8 * \text{real } M + 1$ 
    proof -
      have id:  $E' - \sigma = ((m * \text{of-int } T + E') - m * v) - (v - \text{of-int } T) / 2$ 
        unfolding  $\sigma$ -def e by (simp add: algebra-simps field-simps)
      have r:  $0 \leq v - \text{of-int } T \vee v - \text{of-int } T < 1$  using f1 f2 unfolding e by
        simp-all
      show ?thesis unfolding id using fill r abs-le-iff by argo
    qed
    have  $|E' - \sigma| * (11 * \text{real } M + 1) \leq (8 * \text{real } M + 1) * (11 * \text{real } M + 1)$ 

```

```

    by (rule mult-right-mono[OF Es]) (use Mr in simp)
  moreover have  $(8 * \text{real } M + 1) * (11 * \text{real } M + 1) = 88 * \text{real } M^{\wedge} 2 + 19 * \text{real } M + 1$ 
    by (simp add: algebra-simps power2-eq-square)
  ultimately show ?thesis unfolding e using c M2 Mr by linarith
next
  assume e:  $Tf = T + 1$ 
  have lip:  $|Dd\ K\ p\ T\ E' - Dd\ K\ p\ T\ m| \leq |E' - m| * (11 * \text{real } M + 1)$ 
    by (rule Dd-lip[OF p h]) (use Tl Tu T-def in linarith)+
  have tr:  $Dd\ K\ p\ T\ m = Dd\ K\ p\ (T + 1)\ 0$  unfolding m-def by (rule Dd-trans[OF p h])
  have c:  $|Dd\ K\ p\ (T + 1)\ 0 - Cc\ K\ p\ (T + 1)\ \sigma| \leq 100 * \text{real } M^{\wedge} 2 + |0 - \sigma| * (11 * \text{real } M + 1) + 1$ 
    by (rule Dd-Cc[OF p h]) (use Tl Tu T-def in linarith)+
  have mE:  $m - E' \leq 8 * \text{real } M$ 
proof -
  have vt:  $\text{of-int } T + 1 \leq v$  using f1 unfolding e by simp
  have m0:  $0 \leq m$  using ml Mr by linarith
  have m * ( $\text{of-int } T + 1$ )  $\leq m * v$  by (rule mult-left-mono[OF vt m0])
  thus ?thesis using fill by (simp add: abs-le-iff algebra-simps)
qed
  have s0:  $0 \leq \sigma$  and s1:  $\sigma \leq 12 * \text{real } M$ 
proof -
  have r:  $0 \leq v - \text{of-int } Tf\ v - \text{of-int } Tf < 1$  using f1 f2 by simp-all
  have m0:  $0 \leq m$  using ml Mr by linarith
  show  $0 \leq \sigma$  unfolding  $\sigma$ -def using r m0 by simp
  have mm:  $m * (v - \text{of-int } Tf) \leq 8 * \text{real } M$ 
proof -
  have  $m * (v - \text{of-int } Tf) = m * v - m * (\text{of-int } T + 1)$  unfolding e
    by (simp add: algebra-simps)
  thus ?thesis using fill Em by (simp add: abs-le-iff algebra-simps)
qed
  have  $\sigma = m * (v - \text{of-int } Tf) + (v - \text{of-int } Tf) / 2$  unfolding  $\sigma$ -def
    by (simp add: field-simps)
  thus  $\sigma \leq 12 * \text{real } M$  using mm r Mr by argo
qed
  have eq:  $Dd\ K\ p\ T\ E' - Cc\ K\ p\ (T + 1)\ \sigma = (Dd\ K\ p\ T\ E' - Dd\ K\ p\ T\ m) + (Dd\ K\ p\ (T + 1)\ 0 - Cc\ K\ p\ (T + 1)\ \sigma)$ 
    unfolding tr by simp
  have em:  $|E' - m| \leq 8 * \text{real } M$  using mE Em by (simp add: abs-le-iff)
  have a:  $|E' - m| * (11 * \text{real } M + 1) \leq 8 * \text{real } M * (11 * \text{real } M + 1)$ 
    by (rule mult-right-mono[OF em]) (use Mr in simp)
  have s':  $|0 - \sigma| \leq 12 * \text{real } M$  using s0 s1 by simp
  have b:  $|0 - \sigma| * (11 * \text{real } M + 1) \leq 12 * \text{real } M * (11 * \text{real } M + 1)$ 
    by (rule mult-right-mono[OF s']) (use Mr in simp)
  have c2:  $8 * \text{real } M * (11 * \text{real } M + 1) + 12 * \text{real } M * (11 * \text{real } M + 1)$ 
=
   $220 * \text{real } M^{\wedge} 2 + 20 * \text{real } M$ 
  by (simp add: algebra-simps power2-eq-square)

```

```

    have t: |(Dd K p T E' - Dd K p T m) + (Dd K p (T + 1) 0 - Cc K p (T
+ 1) σ)|
      ≤ |Dd K p T E' - Dd K p T m| + |Dd K p (T + 1) 0 - Cc K p (T +
1) σ| by (rule abs-triangle-ineq)
    show ?thesis unfolding e eq using t lip c a b c2 M2 Mr by linarith
next
  assume e: Tf = T - 1
  have lip: |Dd K p T E' - Dd K p T 0| ≤ |E' - 0| * (11 * real M + 1)
    by (rule Dd-lip[OF p h]) (use Tl Tu T-def in linarith)+
  have tr: Dd K p (T - 1) m = Dd K p T 0 using Dd-trans[OF p h, of T -
1] unfolding m-def
    by simp
  have c: |Dd K p (T - 1) m - Cc K p (T - 1) σ| ≤
    100 * real M ^ 2 + |m - σ| * (11 * real M + 1) + 1
    by (rule Dd-Cc[OF p h]) (use Tl Tu T-def in linarith)+
  have E8: E' ≤ 8 * real M
  proof -
    have vt: v ≤ of-int T using f2 unfolding e by simp
    have m0: 0 ≤ m using ml Mr by linarith
    have m * v ≤ m * of-int T by (rule mult-left-mono[OF vt m0])
    thus ?thesis using fill by (simp add: abs-le-iff algebra-simps)
  qed
  have ms: |m - σ| ≤ 12 * real M
  proof -
    have r: 0 < of-int T - v of-int T - v ≤ 1 using f1 f2 unfolding e by
simp-all
    have m0: 0 ≤ m using ml Mr by linarith
    have id: m - σ = m * (of-int T - v) - (1 - (of-int T - v)) / 2
      unfolding σ-def e by (simp add: algebra-simps field-simps)
    have m * (of-int T - v) ≤ 8 * real M using fill E0
      by (simp add: abs-le-iff algebra-simps)
    moreover have 0 ≤ m * (of-int T - v) using r m0 by simp
    ultimately show ?thesis unfolding id using r Mr abs-le-iff by argo
  qed
  have eq: Dd K p T E' - Cc K p (T - 1) σ =
    (Dd K p T E' - Dd K p T 0) + (Dd K p (T - 1) m - Cc K p (T - 1) σ)
    unfolding tr by simp
  have em: |E' - 0| ≤ 8 * real M using E0 E8 by simp
  have a: |E' - 0| * (11 * real M + 1) ≤ 8 * real M * (11 * real M + 1)
    by (rule mult-right-mono[OF em]) (use Mr in simp)
  have b: |m - σ| * (11 * real M + 1) ≤ 12 * real M * (11 * real M + 1)
    by (rule mult-right-mono[OF ms]) (use Mr in simp)
  have c2: 8 * real M * (11 * real M + 1) + 12 * real M * (11 * real M + 1)
=
    220 * real M ^ 2 + 20 * real M
    by (simp add: algebra-simps power2-eq-square)
  have t: |(Dd K p T E' - Dd K p T 0) + (Dd K p (T - 1) m - Cc K p (T
- 1) σ)|
    ≤ |Dd K p T E' - Dd K p T 0| + |Dd K p (T - 1) m - Cc K p (T -

```

```

1)  $\sigma$  | by (rule abs-triangle-ineq)
  show ?thesis unfolding e eq using t lip c a b c2 M2 Mr by linarith
qed
show ?thesis
  unfolding decomp
  using abs-triangle-ineq [of Zr Dd K p T E' - Cc K p Tf  $\sigma$ ] Z[folded Zr-def]
main
  by linarith
qed

```

Lemma 5.7 with $C(M) = 1000 M^2$.

```

theorem gammaIn-uniform-estimate:
  assumes  $40 \leq M$ 
  shows  $\exists Cst. \forall K p. \text{prime } p \longrightarrow \text{InnerHyp } K M p \longrightarrow$ 
     $|of\text{-rat } (\text{gammaIn } K M p) - \text{real } p * \text{Gam-r } (\text{real } K / \text{real } p)| \leq Cst \wedge$ 
     $|of\text{-int } (vpS K p) - \text{real } p * \text{Nfun-r } (\text{real } K / \text{real } p)| \leq Cst$ 
proof (intro exI[ $of - 1000 * \text{real } M^2$ ] allI impI conjI)
  fix K p assume p: prime p and h: InnerHyp K M p
  show  $|of\text{-rat } (\text{gammaIn } K M p) - \text{real } p * \text{Gam-r } (\text{real } K / \text{real } p)| \leq$ 
     $1000 * \text{real } M^2$  by (rule gamma-bound[OF p h])
  have  $4 * \text{real } M \leq 1000 * \text{real } M^2$ 
  proof -
    have Mr: 40 ≤ real M using assms by simp
    have  $40 * \text{real } M \leq \text{real } M^2$  using Mr by (simp add: power2-eq-square)
    thus ?thesis using Mr by linarith
  qed
  thus  $|of\text{-int } (vpS K p) - \text{real } p * \text{Nfun-r } (\text{real } K / \text{real } p)| \leq 1000 * \text{real } M^2$ 
    using vpS-bound[OF p h] by linarith
qed

```

end

```

theory Zeta5-Explicit-Integrals
  imports Zeta5-Definitions Piecewise-Interval-Integrals
begin

```

26 The explicit integrals of Section 5

The closed form $R(x) = x F(x) + Q(x)$ with $-1/2 \leq Q \leq 13/8$ ((B.1), (5.13)), the exact integrals (5.10) and (5.18) as sums of affine pieces (the piece lemmas are generated from exact rational tables, mirroring the generated Lean files), the tails (5.14)–(5.17), and continuity of the limiting functions off a countable set. All integrals are *interval-lebesgue-integral* over *lborel*. Lean theories *Integrals5Base*, *Integrals5Gam*, *Integrals5*, *Integrals5Outer*, *Integrals5Inner*, *Integrals5Tail*, *RCont*.

26.1 The closed form $R(x) = x F(x) + Q(x)$ (B.1), (5.13)

definition $ind :: real \Rightarrow real \Rightarrow real$ **where** $ind\ p\ z = (if\ p < z\ then\ 1\ else\ 0)$

definition $d0 :: real \Rightarrow real$ **where** $d0\ u = min\ u\ (1 - u)$

definition $es :: real \Rightarrow real$ **where** $es\ u = (if\ u \leq 1 / 2\ then\ 1\ else\ -1)$

lemma $ind-of-lt: p < z \implies ind\ p\ z = 1$ **and** $ind-of-le: z \leq p \implies ind\ p\ z = 0$
by ($simp-all\ add: ind-def$)

lemma $es-cases: (es\ u = 1 \wedge d0\ u = u) \vee (es\ u = -1 \wedge d0\ u = 1 - u)$
unfolding $es-def\ d0-def$ **by** ($cases\ u \leq 1 / 2$) ($auto\ simp: min-def$)

lemma $d0-mem: 0 \leq u \implies u \leq 1 \implies 0 \leq d0\ u \wedge d0\ u \leq 1 / 2$
unfolding $d0-def$ **by** ($auto\ simp: min-def$)

lemma $es-pm: es\ u = 1 \vee es\ u = -1$
by ($simp\ add: es-def$)

Integral over $(0, 1/2)$ of a product of two step functions with steps at p and q .

lemma $step3$:

fixes $h :: real \Rightarrow real$

assumes $p: 0 \leq p \leq 1 / 2$ **and** $q: 0 \leq q \leq 1 / 2$

and $H: \bigwedge z. z \in \{0 < .. < 1 / 2\} \implies z \neq p \implies$

$z \neq q \implies$

$h\ z = (a0 + a1 * ind\ q\ z) * (b0 + b1 * ind\ q\ z + b2 * ind\ p\ z)$

shows $interval-lebesgue-integral\ lborel\ (ereal\ 0)\ (ereal\ (1 / 2))\ h$

$= a0 * b0 / 2 + (a0 * b1 + a1 * b0 + a1 * b1) * (1 / 2 - q) + a0 * b2 * (1 / 2 - p) +$

$a1 * b2 * (1 / 2 - max\ p\ q)$

proof ($cases\ p \leq q$)

case $True$

have $e1: interval-lebesgue-integrable\ lborel\ (ereal\ 0)\ (ereal\ p)\ h \wedge$

$interval-lebesgue-integral\ lborel\ (ereal\ 0)\ (ereal\ p)\ h = (p - 0) * (a0 * b0)$

by ($rule\ const-piece[OF\ p(1)]$) ($use\ True\ q\ H\ in\ \langle auto\ simp: ind-def \rangle$)

have $e2: interval-lebesgue-integrable\ lborel\ (ereal\ p)\ (ereal\ q)\ h \wedge$

$interval-lebesgue-integral\ lborel\ (ereal\ p)\ (ereal\ q)\ h = (q - p) * (a0 * (b0 + b2))$

by ($rule\ const-piece[OF\ True]$) ($use\ p\ q\ H\ in\ \langle auto\ simp: ind-def \rangle$)

have $e3: interval-lebesgue-integrable\ lborel\ (ereal\ q)\ (ereal\ (1 / 2))\ h \wedge$

$interval-lebesgue-integral\ lborel\ (ereal\ q)\ (ereal\ (1 / 2))\ h =$

$(1 / 2 - q) * ((a0 + a1) * (b0 + b1 + b2))$

by ($rule\ const-piece[OF\ q(2)]$) ($use\ p\ q\ True\ H\ in\ \langle auto\ simp: ind-def \rangle$)

from $chain[OF\ p(1)\ True\ e1\ e2]$ **have** $e12: interval-lebesgue-integrable\ lborel\ (ereal\ 0)\ (ereal\ q)\ h \wedge$

$h \wedge$

$interval-lebesgue-integral\ lborel\ (ereal\ 0)\ (ereal\ q)\ h =$

$(p - 0) * (a0 * b0) + (q - p) * (a0 * (b0 + b2))$.
from chain[OF q(1) q(2) e12 e3] **show** ?thesis **using** True **by** (simp add: max-def algebra-simps)
next
case False
hence qp: $q \leq p$ **by** simp
have e1: interval-lebesgue-integrable lborel (ereal 0) (ereal q) h $\wedge$
interval-lebesgue-integral lborel (ereal 0) (ereal q) h = $(q - 0) * (a0 * b0)$
by (rule const-piece[OF q(1)]) (use qp p H **in** ‹auto simp: ind-def›)
have e2: interval-lebesgue-integrable lborel (ereal q) (ereal p) h $\wedge$
interval-lebesgue-integral lborel (ereal q) (ereal p) h = $(p - q) * ((a0 + a1) * (b0 + b1))$
by (rule const-piece[OF qp]) (use p q H **in** ‹auto simp: ind-def›)
have e3: interval-lebesgue-integrable lborel (ereal p) (ereal (1 / 2)) h $\wedge$
interval-lebesgue-integral lborel (ereal p) (ereal (1 / 2)) h =
 $(1 / 2 - p) * ((a0 + a1) * (b0 + b1 + b2))$
by (rule const-piece[OF p(2)]) (use p q qp H **in** ‹auto simp: ind-def›)
from chain[OF q(1) qp e1 e2] **have** e12: interval-lebesgue-integrable lborel (ereal 0) (ereal p) h
 $\wedge$
interval-lebesgue-integral lborel (ereal 0) (ereal p) h =
 $(q - 0) * (a0 * b0) + (p - q) * ((a0 + a1) * (b0 + b1))$.
from chain[OF p(1) p(2) e12 e3] **show** ?thesis **using** False **by** (simp add: max-def algebra-simps)
qed

$l(y, z) = 2 \lfloor y \rfloor + 1 - es\{y\} \cdot ind(d0\{y\}, z)$ for $z \in (0, 1/2)$, $z \neq d0\{y\}$.

lemma ell-step:

assumes z: $z \in \{0 < .. < 1 / 2\}$ **and** zp: $z \neq d0 \text{ (frac } y\text{)}$
shows ell-r $y \ z = 2 * of\text{-int} \text{ (floor } y\text{)} + 1 - es \text{ (frac } y\text{)} * ind \text{ (} d0 \text{ (frac } y\text{)) } z$
proof –
have y: $y = of\text{-int} \text{ (floor } y\text{)} + \text{frac } y$ **by** (simp add: frac-def)
have f: $0 \leq \text{frac } y \text{ frac } y < 1$ **by** (simp-all add: frac-lt-1)
show ?thesis
proof (cases $\text{frac } y \leq 1 / 2$)
case True
have d: $d0 \text{ (frac } y\text{)} = \text{frac } y$ **and** e: $es \text{ (frac } y\text{)} = 1$ **using** True **unfolding**
d0-def es-def
by (auto simp: min-def)
have h2: $\text{floor } (y + z) = \text{floor } y$ **using** z True f **unfolding** frac-def greaterThanLessThan-iff
by (intro floor-eq-iff[THEN iffD2] conjI) (elim conjE, linarith)+
show ?thesis
proof (cases $z < \text{frac } y$)
case True
have h1: $\text{floor } (y - z) = \text{floor } y$ **using** z True f **unfolding** frac-def
greaterThanLessThan-iff
by (intro floor-eq-iff[THEN iffD2] conjI) (elim conjE, linarith)+
show ?thesis **unfolding** ell-r-def h1 h2 d e **using** True **by** (simp add: ind-def)
next

```

    case False
    hence lt: frac y < z using zp d by auto
    have h1: floor (y - z) = floor y - 1 using z lt f unfolding frac-def
    greaterThanLessThan-iff
    by (intro floor-eq-iff[THEN iffD2] conjI) (elim conjE, linarith)+
    show ?thesis unfolding ell-r-def h1 h2 d e using lt by (simp add: ind-def)
  qed
next
  case False
  have d: d0 (frac y) = 1 - frac y and e: es (frac y) = - 1 using False
  unfolding d0-def es-def by (auto simp: min-def)
  have h1: floor (y - z) = floor y using z False f unfolding frac-def greaterThanLessThan-iff
  by (intro floor-eq-iff[THEN iffD2] conjI) (elim conjE, linarith)+
  show ?thesis
  proof (cases z < 1 - frac y)
    case True
    have h2: floor (y + z) = floor y using z True f unfolding frac-def
    greaterThanLessThan-iff
    by (intro floor-eq-iff[THEN iffD2] conjI) (elim conjE, linarith)+
    show ?thesis unfolding ell-r-def h1 h2 d e using True by (simp add: ind-def)
  next
    case False
    hence lt: 1 - frac y < z using zp d by auto
    have h2: floor (y + z) = floor y + 1 using z lt f unfolding frac-def
    greaterThanLessThan-iff
    by (intro floor-eq-iff[THEN iffD2] conjI) (elim conjE, linarith)+
    show ?thesis unfolding ell-r-def h1 h2 d e using lt by (simp add: ind-def)
  qed
  qed
  qed

```

The z -integral in Γ .

lemma *gam-int*:

```

interval-lebesgue-integral lborel (ereal 0) (ereal (1 / 2))
(λz. (T - 3 * ell-r (alpha-r * x) z) * (T + 3 * ell-r (alpha-r * x) z - ell-r
x z - 5))
= (let p = d0 (frac x); q = d0 (frac (alpha-r * x));
a0 = T - 3 * (2 * of-int (floor (alpha-r * x)) + 1); a1 = 3 * es (frac
(alpha-r * x));
b0 = T + 3 * (2 * of-int (floor (alpha-r * x)) + 1) - (2 * of-int (floor
x) + 1) - 5;
b1 = - 3 * es (frac (alpha-r * x)); b2 = es (frac x) in
a0 * b0 / 2 + (a0 * b1 + a1 * b0 + a1 * b1) * (1 / 2 - q) + a0 * b2 *
(1 / 2 - p) + a1 * b2 *
(1 / 2 - max p q))

```

proof -

```

  have hp: 0 ≤ d0 (frac x) ∧ d0 (frac x) ≤ 1 / 2
  by (rule d0-mem) (simp-all add: frac-lt-1 less-imp-le)
  have hq: 0 ≤ d0 (frac (alpha-r * x)) ∧ d0 (frac (alpha-r * x)) ≤ 1 / 2

```

by (*rule d0-mem*) (*simp-all add: frac-lt-1 less-imp-le*)
show *?thesis*
unfolding *Let-def*
by (*rule step3*[*OF conjunct1*[*OF hp*] *conjunct2*[*OF hp*] *conjunct1*[*OF hq*] *conjunct2*[*OF hq*]])
(simp add: ell-step algebra-simps)
qed

$Q = R - x F$, eq. (5.13) with (B.1); arguments $\{x\}$, $\{\alpha x\}$, $\{2H-c x\}$, $\{2x\}$, $\{2\lambda x\}$.

definition $Qc :: real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real$
 $\Rightarrow real$ **where**

$$\begin{aligned}
 Qc \ f \ g \ \tau \ \sigma \ \eta = & 9 * (d0 \ g * (1 - 2 * d0 \ g)) \\
 & - 3 * (es \ f * es \ g * (d0 \ f + d0 \ g - \max(d0 \ f) (d0 \ g) - 2 * d0 \ f * d0 \ g)) \\
 & + ((\tau \wedge 2 - \sigma * \tau) / 2 - pos-r ((\tau - \sigma) / 2)) + (\eta - \\
 & \eta \wedge 2) / 2
 \end{aligned}$$

definition $Fx :: real \Rightarrow real$ **where**

$$Fx \ x = 4 * lam-r + 2 * lam-r * frac \ x - 12 * lam-r * frac (\alpha x)$$

The polynomial identities behind the three remaining sign cases of R -eq, with the fractional parts, the $\max$ and the $pos-r$ term as variables.

lemma R -eq-*alg2*:

fixes $x \ fx \ fa \ fh \ f2 \ fl \ mx \ pp :: real$
shows $- ((2 * Hc-r * x - fh - 3 * (2 * (\alpha x - fa) + 1)) * (2 * Hc-r * x - fh + 3 * (2 * (\alpha x - fa) + 1) - (2 * (x - fx) + 1) - 5) / 2 + ((2 * Hc-r * x - fh - 3 * (2 * (\alpha x - fa) + 1)) * (-3 * -1) + 3 * -1 * (2 * Hc-r * x - fh + 3 * (2 * (\alpha x - fa) + 1) - (2 * (x - fx) + 1) - 5) + 3 * -1 * (-3 * -1)) * (1 / 2 - (1 - fa)) + (2 * Hc-r * x - fh - 3 * (2 * (\alpha x - fa) + 1)) * 1 * (1 / 2 - fx) + 3 * -1 * 1 * (1 / 2 - mx) + (Hc-r * x - (2 * Hc-r * x - fh) / 2) * (2 * (2 * Hc-r * x - fh) - (2 * x - f2) - 5) + pp) - (2 * lam-r * x * (x - fx) - 12 * lam-r * x * (\alpha x - fa) - 2 * ((2 * (lam-r * x) - fl) * (lam-r * x) - (2 * (lam-r * x) - fl) * (2 * (lam-r * x) - fl + 1) / 4)) = x * (4 * lam-r + 2 * lam-r * fx - 12 * lam-r * fa) + (9 * ((1 - fa) * (1 - 2 * (1 - fa))) - 3 * (1 * -1 * (fx + (1 - fa) - mx - 2 * fx * (1 - fa))) + (((fh)² - f2 * fh) / 2 - pp) + (fl - (fl)²) / 2)$

proof $-$

$-$ monomial expansions of the products

have $(((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) * ((((((2 * (23 /$

$$\begin{aligned}
& (20)) * x) - fh) + (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) - ((2 * (x - fx)) \\
& + 1)) - 5)) = \\
& (9) + (- 39 / 5) * (x) + (- 6) * (fx) + (6) * (fh) + (111 / 80) * (x * x) + \\
& (37 / 10) * (x * \\
& fx) + (- 33 / 5) * (x * fa) + (- 13 / 5) * (x * fh) + (12) * (fx * fa) + (- \\
& 2) * (fx * fh) + \\
& (- 36) * (fa * fa) + (1) * (fh * fh) \\
& \text{using algebra-simps by argo} \\
& \text{moreover have } (((((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) \\
& - fa)) + 1)))) * \\
& ((- 3) * (- 1))) + ((3 * (- 1)) * (((((((2 * (23 / 20)) * x) - fh) + (3 * ((2 \\
& * (((3 / 40) * x) - \\
& fa)) + 1)))) - ((2 * (x - fx)) + 1)) - 5))) + ((3 * (- 1)) * ((- 3) * (- 1)))) \\
& * ((1 / 2) - (1 - \\
& fa))) = (9 / 2) + (- 33 / 20) * (x) + (3) * (fx) + (- 27) * (fa) + (33 / 10) \\
& * (x * fa) + \\
& (- 6) * (fx * fa) + (36) * (fa * fa) \\
& \text{using algebra-simps by argo} \\
& \text{moreover have } (((((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) - \\
& fa)) + 1)))) * 1) * \\
& ((1 / 2) - fx)) = (- 3 / 2) + (37 / 40) * (x) + (3) * (fx) + (3) * (fa) + (- \\
& 1 / 2) * (fh) + \\
& (- 37 / 20) * (x * fx) + (- 6) * (fx * fa) + (1) * (fx * fh) \\
& \text{using algebra-simps by argo} \\
& \text{moreover have } (((((23 / 20) * x) - (((2 * (23 / 20)) * x) - fh) / 2)) * (((2 \\
& * (((2 * (23 / \\
& 20)) * x) - fh)) - ((2 * x) - f2)) - 5)) = \\
& (- 5 / 2) * (fh) + (13 / 10) * (x * fh) + (- 1) * (fh * fh) + (1 / 2) * (fh * \\
& f2) \\
& \text{using algebra-simps by argo} \\
& \text{moreover have } (((2 * (37 / 40)) * x) * (x - fx)) = (37 / 20) * (x * x) + (- \\
& 37 / 20) * (x * fx) \\
& \text{using algebra-simps by argo} \\
& \text{moreover have } (((12 * (37 / 40)) * x) * (((3 / 40) * x) - fa)) = \\
& (333 / 400) * (x * x) + (- 111 / 10) * (x * fa) \\
& \text{using algebra-simps by argo} \\
& \text{moreover have } (((2 * ((37 / 40) * x)) - fl) * ((37 / 40) * x)) = \\
& (1369 / 800) * (x * x) + (- 37 / 40) * (x * fl) \\
& \text{by (fastforce simp: algebra-simps)} \\
& \text{moreover have } (((2 * ((37 / 40) * x)) - fl) * (((2 * ((37 / 40) * x)) - fl) + \\
& 1)) = \\
& (37 / 20) * (x) + (- 1) * (fl) + (1369 / 400) * (x * x) + (- 37 / 10) * (x \\
& * fl) + (1) * (fl * \\
& fl) \\
& \text{by (auto simp: algebra-simps)} \\
& \text{moreover have } (x * (((4 * (37 / 40)) + ((2 * (37 / 40)) * fx)) - ((12 * (37 \\
& / 40)) * fa))) = \\
& (37 / 10) * (x) + (37 / 20) * (x * fx) + (- 111 / 10) * (x * fa) \\
& \text{by (simp add: algebra-simps)}
\end{aligned}$$

moreover have $((1 - fa) * (1 - (2 * (1 - fa)))) = (-1) + (3) * (fa) + (-2) * (fa * fa)$
by (*auto simp: algebra-simps*)
moreover have $((2 * fx) * (1 - fa)) = (2) * (fx) + (-2) * (fx * fa)$
by (*simp add: algebra-simps*)
moreover have $(f2 * fh) = (1) * (fh * f2)$
by (*auto simp: algebra-simps*)
ultimately show *?thesis* **unfolding** *Hc-r-def lam-r-def alpha-r-def power2-eq-square*
by *argo*
qed

lemma *R-eq-alg3*:

fixes *x fx fa fh f2 fl mx pp :: real*
shows $-((2 * Hc-r * x - fh - 3 * (2 * (alpha-r * x - fa) + 1)) * (2 * Hc-r * x - fh + 3 * (2 * (alpha-r * x - fa) + 1) - (2 * (x - fx) + 1) - 5) / 2 + ((2 * Hc-r * x - fh - 3 * (2 * (alpha-r * x - fa) + 1)) * (-3 * 1) + 3 * 1 * (2 * Hc-r * x - fh + 3 * (2 * (alpha-r * x - fa) + 1) - (2 * (x - fx) + 1) - 5) + 3 * 1 * (-3 * 1)) * (1 / 2 - fa) + (2 * Hc-r * x - fh - 3 * (2 * (alpha-r * x - fa) + 1)) * -1 * (1 / 2 - (1 - fx)) + 3 * 1 * -1 * (1 / 2 - mx) + (Hc-r * x - (2 * Hc-r * x - fh) / 2) * (2 * (2 * Hc-r * x - fh) - (2 * x - f2) - 5) + pp) - (2 * lam-r * x * (x - fx) - 12 * lam-r * x * (alpha-r * x - fa) - 2 * ((2 * (lam-r * x) - fl) * (lam-r * x) - (2 * (lam-r * x) - fl) * (2 * (lam-r * x) - fl + 1) / 4)) = x * (4 * lam-r + 2 * lam-r * fx - 12 * lam-r * fa) + (9 * (fa * (1 - 2 * fa)) - 3 * (-1 * 1 * (1 - fx + fa - mx - 2 * (1 - fx) * fa)) + (((fh)² - f2 * fh) / 2 - pp) + (fl - (fl)²) / 2)$
proof —

— monomial expansions of the products

have $(((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) * ((((((2 * (23 / 20)) * x) - fh) + (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) - ((2 * (x - fx)) + 1)) - 5)) = (9) + (-39 / 5) * (x) + (-6) * (fx) + (6) * (fh) + (111 / 80) * (x * x) + (37 / 10) * (x * fx) + (-33 / 5) * (x * fa) + (-13 / 5) * (x * fh) + (12) * (fx * fa) + (-2) * (fx * fh) + (-36) * (fa * fa) + (1) * (fh * fh)$
using *algebra-simps* **by** *argo*
moreover have $((((-3) * 1)) + ((3 * 1) * ((((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) * ((((((2 * (23 / 20)) * x) - fh) + (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) - ((2 * (x - fx)) + 1)) - 5)) = (9) + (-39 / 5) * (x) + (-6) * (fx) + (6) * (fh) + (111 / 80) * (x * x) + (37 / 10) * (x * fx) + (-33 / 5) * (x * fa) + (-13 / 5) * (x * fh) + (12) * (fx * fa) + (-2) * (fx * fh) + (-36) * (fa * fa) + (1) * (fh * fh)$

$$1))) - ((2 * (x - fx)) + 1) - 5))) + ((3 * 1) * ((- 3) * 1))) * ((1 / 2) - fa)) =$$

$$(- 9 / 2) + (- 33 / 20) * (x) + (3) * (fx) + (- 9) * (fa) + (33 / 10) * (x * fa) + (- 6) * (fx * fa) + (36) * (fa * fa)$$
using algebra-simps by argo
moreover have ((((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * ((3 / 40) * x) - fa)) + 1))) * (- 1)) * ((1 / 2) - (1 - fx))) = (- 3 / 2) + (37 / 40) * (x) + (3) * (fx) + (3) * (fa) + (- 1 / 2) * (fh) + (- 37 / 20) * (x * fx) + (- 6) * (fx * fa) + (1) * (fx * fh)
using algebra-simps by argo
moreover have (((((23 / 20) * x) - (((2 * (23 / 20)) * x) - fh) / 2)) * (((2 * ((2 * (23 / 20)) * x) - fh)) - ((2 * x) - f2)) - 5)) = (- 5 / 2) * (fh) + (13 / 10) * (x * fh) + (- 1) * (fh * fh) + (1 / 2) * (fh * f2)
using algebra-simps by argo
moreover have (((2 * (37 / 40)) * x) * (x - fx)) = (37 / 20) * (x * x) + (- 37 / 20) * (x * fx)
using algebra-simps by argo
moreover have (((12 * (37 / 40)) * x) * (((3 / 40) * x) - fa)) = (333 / 400) * (x * x) + (- 111 / 10) * (x * fa)
using algebra-simps by argo
moreover have (((2 * ((37 / 40) * x)) - fl) * ((37 / 40) * x)) = (1369 / 800) * (x * x) + (- 37 / 40) * (x * fl)
by (fastforce simp: algebra-simps)
moreover have (((2 * ((37 / 40) * x)) - fl) * (((2 * ((37 / 40) * x)) - fl) + 1)) = (37 / 20) * (x) + (- 1) * (fl) + (1369 / 400) * (x * x) + (- 37 / 10) * (x * fl) + (1) * (fl * fl)
by (auto simp: algebra-simps)
moreover have (x * (((4 * (37 / 40)) + ((2 * (37 / 40)) * fx)) - ((12 * (37 / 40)) * fa))) = (37 / 10) * (x) + (37 / 20) * (x * fx) + (- 111 / 10) * (x * fa)
by (auto simp: algebra-simps)
moreover have (fa * (1 - (2 * fa))) = (1) * (fa) + (- 2) * (fa * fa)
by (simp add: algebra-simps)
moreover have ((2 * (1 - fx)) * fa) = (2) * (fa) + (- 2) * (fx * fa)
by (auto simp: algebra-simps)
moreover have (f2 * fh) = (1) * (fh * f2)
by (simp add: algebra-simps)
ultimately show ?thesis unfolding Hc-r-def lam-r-def alpha-r-def power2-eq-square by argo
qed

lemma R-eq-alg4:
fixes x fx fa fh f2 fl mx pp :: real

shows $- ((2 * Hc-r * x - fh - 3 * (2 * (alpha-r * x - fa) + 1)) * (2 * Hc-r * x - fh + 3 * (2 * (alpha-r * x - fa) + 1) - (2 * (x - fx) + 1) - 5) / 2 + ((2 * Hc-r * x - fh - 3 * (2 * (alpha-r * x - fa) + 1)) * (- 3 * - 1) + 3 * - 1 * (2 * Hc-r * x - fh + 3 * (2 * (alpha-r * x - fa) + 1) - (2 * (x - fx) + 1) - 5) + 3 * - 1 * (- 3 * - 1)) * (1 / 2 - (1 - fa)) + (2 * Hc-r * x - fh - 3 * (2 * (alpha-r * x - fa) + 1)) * - 1 * (1 / 2 - (1 - fx)) + 3 * - 1 * - 1 * (1 / 2 - mx) + (Hc-r * x - (2 * Hc-r * x - fh) / 2) * (2 * (2 * Hc-r * x - fh) - (2 * x - f2) - 5) + pp) - (2 * lam-r * x * (x - fx) - 12 * lam-r * x * (alpha-r * x - fa) - 2 * ((2 * (lam-r * x) - fl) * (lam-r * x) - (2 * (lam-r * x) - fl) * (2 * (lam-r * x) - fl + 1) / 4)) = x * (4 * lam-r + 2 * lam-r * fx - 12 * lam-r * fa) + (9 * ((1 - fa) * (1 - 2 * (1 - fa))) - 3 * (- 1 * - 1 * (1 - fx + (1 - fa) - mx - 2 * (1 - fx) * (1 - fa))) + (((fh)² - f2 * fh) / 2 - pp) + (fl - (fl)²) / 2)$

proof $-$

$-$ monomial expansions of the products

have $(((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) * ((((((2 * (23 / 20)) * x) - fh) + (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) - ((2 * (x - fx) + 1) - 5))) = (9) + (- 39 / 5) * (x) + (- 6) * (fx) + (6) * (fh) + (111 / 80) * (x * x) + (37 / 10) * (x * fx) + (- 33 / 5) * (x * fa) + (- 13 / 5) * (x * fh) + (12) * (fx * fa) + (- 2) * (fx * fh) + (- 36) * (fa * fa) + (1) * (fh * fh)$

using algebra-simps by argo

moreover have $(((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) * ((((- 3) * (- 1))) + ((3 * (- 1)) * ((((((2 * (23 / 20)) * x) - fh) + (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) - ((2 * (x - fx) + 1) - 5))) + ((3 * (- 1)) * ((- 3) * (- 1)))) * ((1 / 2) - (1 - fa))) = (9 / 2) + (- 33 / 20) * (x) + (3) * (fx) + (- 27) * (fa) + (33 / 10) * (x * fa) + (- 6) * (fx * fa) + (36) * (fa * fa)$

using algebra-simps by argo

moreover have $(((((2 * (23 / 20)) * x) - fh) - (3 * ((2 * (((3 / 40) * x) - fa)) + 1))) * ((- 1)) * ((1 / 2) - (1 - fx))) = (- 3 / 2) + (37 / 40) * (x) + (3) * (fx) + (3) * (fa) + (- 1 / 2) * (fh) + (- 37 / 20) * (x * fx) + (- 6) * (fx * fa) + (1) * (fx * fh)$

using algebra-simps by argo

moreover have $(((((23 / 20) * x) - (((2 * (23 / 20)) * x) - fh) / 2)) * (((2$

$$* (((2 * (23 / 20)) * x) - fh)) - ((2 * x) - f2)) - 5)) =$$

$$(- 5 / 2) * (fh) + (13 / 10) * (x * fh) + (- 1) * (fh * fh) + (1 / 2) * (fh * f2)$$
using algebra-simps by argo
moreover have $((2 * (37 / 40)) * x) * (x - fx) = (37 / 20) * (x * x) + (- 37 / 20) * (x * fx)$
using algebra-simps by argo
moreover have $((12 * (37 / 40)) * x) * (((3 / 40) * x) - fa) =$

$$(333 / 400) * (x * x) + (- 111 / 10) * (x * fa)$$
using algebra-simps by argo
moreover have $((2 * ((37 / 40) * x)) - fl) * ((37 / 40) * x) =$

$$(1369 / 800) * (x * x) + (- 37 / 40) * (x * fl)$$
by (auto simp: algebra-simps)
moreover have $((2 * ((37 / 40) * x)) - fl) * (((2 * ((37 / 40) * x)) - fl) + 1)) =$

$$(37 / 20) * (x) + (- 1) * (fl) + (1369 / 400) * (x * x) + (- 37 / 10) * (x * fl) + (1) * (fl * fl)$$
by (simp add: algebra-simps)
moreover have $(x * (((4 * (37 / 40)) + ((2 * (37 / 40)) * fx)) - ((12 * (37 / 40)) * fa))) =$

$$(37 / 10) * (x) + (37 / 20) * (x * fx) + (- 111 / 10) * (x * fa)$$
by (force simp: algebra-simps)
moreover have $((1 - fa) * (1 - (2 * (1 - fa)))) = (- 1) + (3) * (fa) + (- 2) * (fa * fa)$
by (force simp: algebra-simps)
moreover have $((2 * (1 - fx)) * (1 - fa)) = (2) + (- 2) * (fx) + (- 2) * (fa) + (2) * (fx * fa)$
by (simp add: algebra-simps)
moreover have $(f2 * fh) = (1) * (fh * f2)$
by (auto simp: algebra-simps)
ultimately show ?thesis unfolding Hc-r-def lam-r-def alpha-r-def power2-eq-square by argo
qed

theorem R-eq:

$$R-r \ x = x * Fx \ x + Qc \ (\text{frac } x) \ (\text{frac } (\alpha\text{-}r * x)) \ (\text{frac } (2 * Hc\text{-}r * x)) \ (\text{frac } (2 * x)) \ (\text{frac } (2 * (\text{lam-r} * x)))$$

proof –

have $fl: \bigwedge u. \text{ of-int } (\text{floor } u) = u - \text{frac } (u :: \text{real})$ **by (simp add: frac-def)**
have $pe: \text{pos-r } (Hc\text{-}r * x - (2 * Hc\text{-}r * x - \text{frac } (2 * Hc\text{-}r * x)) / 2 - (2 * x - (2 * x - \text{frac } (2 * x))) / 2) = \text{pos-r } ((\text{frac } (2 * Hc\text{-}r * x) - \text{frac } (2 * x)) / 2)$
using algebra-simps by argo
show ?thesis
using es-cases[*of* frac x] es-cases[*of* frac (alpha-r * x)]
proof (elim disjE conjE)

```

assume  $es \text{ (frac } x) = 1 \text{ d0 (frac } x) = \text{frac } x$ 
and  $es \text{ (frac (alpha-r * x)) = 1 d0 (frac (alpha-r * x)) = frac (alpha-r * x)}$ 
thus ?thesis unfolding R-r-def Gam-r-def Nfun-r-def J-r-def Fx-def Let-def
gam-int
by (simp only: fl pe Qc-def) (auto simp: alpha-r-def lam-r-def Hc-r-def
field-simps
power2-eq-square)
next
assume  $es \text{ (frac } x) = 1 \text{ d0 (frac } x) = \text{frac } x$ 
and  $es \text{ (frac (alpha-r * x)) = - 1 d0 (frac (alpha-r * x)) = 1 - frac (alpha-r$ 
* x)
thus ?thesis unfolding R-r-def Gam-r-def Nfun-r-def J-r-def Fx-def Let-def
gam-int
by (simp only: fl pe Qc-def) (use R-eq-alg2 in fast)
next
assume  $es \text{ (frac } x) = - 1 \text{ d0 (frac } x) = 1 - \text{frac } x$ 
and  $es \text{ (frac (alpha-r * x)) = 1 d0 (frac (alpha-r * x)) = frac (alpha-r * x)}$ 
thus ?thesis unfolding R-r-def Gam-r-def Nfun-r-def J-r-def Fx-def Let-def
gam-int
by (simp only: fl pe Qc-def) (use R-eq-alg3 in blast)
next
assume  $es \text{ (frac } x) = - 1 \text{ d0 (frac } x) = 1 - \text{frac } x$ 
and  $es \text{ (frac (alpha-r * x)) = - 1 d0 (frac (alpha-r * x)) = 1 - frac (alpha-r$ 
* x)
thus ?thesis unfolding R-r-def Gam-r-def Nfun-r-def J-r-def Fx-def Let-def
gam-int
by (simp only: fl pe Qc-def) (use R-eq-alg4 in fast)
qed
qed

```

lemma *Qc-bounds*:

```

assumes  $0 \leq f \wedge f \leq 1 \wedge 0 \leq g \wedge g \leq 1 \wedge 0 \leq \tau \wedge \tau \leq 1 \wedge 0 \leq$ 
 $\sigma \wedge \sigma \leq 1 \wedge 0 \leq \eta \wedge \eta \leq 1$ 
shows  $- 1 / 2 \leq Qc \text{ f g } \tau \sigma \eta \wedge$ 
 $Qc \text{ f g } \tau \sigma \eta \leq 13 / 8$ 
proof -
have hp:  $0 \leq d0 \text{ f } \wedge d0 \text{ f } \leq 1 / 2$  and hq:  $0 \leq d0 \text{ g } \wedge d0 \text{ g } \leq 1 / 2$ 
by (rule d0-mem; simp add: assms)+
have  $0 \leq d0 \text{ g } * (1 - 2 * d0 \text{ g}) \wedge d0 \text{ g } * (1 - 2 * d0 \text{ g}) \leq 1 / 8$ 
proof -
have  $d0 \text{ g } * (1 - 2 * d0 \text{ g}) = 1 / 8 - 2 * (d0 \text{ g } - 1 / 4) ^ 2$ 
by (simp add: power2-eq-square algebra-simps)
moreover have  $0 \leq (d0 \text{ g } - 1 / 4) ^ 2$  by simp
moreover have  $0 \leq d0 \text{ g } * (1 - 2 * d0 \text{ g})$  using hq by auto
ultimately show ?thesis by argo
qed
moreover have  $0 \leq d0 \text{ f } + d0 \text{ g } - \max (d0 \text{ f}) (d0 \text{ g}) - 2 * d0 \text{ f } * d0 \text{ g } \wedge$ 
 $d0 \text{ f } + d0 \text{ g } - \max (d0 \text{ f}) (d0 \text{ g}) - 2 * d0 \text{ f } * d0 \text{ g } \leq 1 / 8$ 
proof (cases d0 f ≤ d0 g)

```

```

    case True
    have  $d0\ f + d0\ g - \max(d0\ f)\ (d0\ g) - 2 * d0\ f * d0\ g = d0\ f * (1 - 2 * d0\ g)$  using True
    by (simp add: max-def algebra-simps)
    moreover have  $d0\ f * (1 - 2 * d0\ g) \leq d0\ g * (1 - 2 * d0\ g)$  using True hq
    by (auto intro: mult-right-mono)
    moreover have  $d0\ g * (1 - 2 * d0\ g) = 1 / 8 - 2 * (d0\ g - 1 / 4) ^ 2$ 
    by (simp add: power2-eq-square algebra-simps)
    moreover have  $0 \leq (d0\ g - 1 / 4) ^ 2$  by simp
    moreover have  $0 \leq d0\ f * (1 - 2 * d0\ g)$  using hp hq by (auto intro: mult-nonneg-nonneg)
    ultimately show ?thesis by argo
  next
  case False
  have  $d0\ f + d0\ g - \max(d0\ f)\ (d0\ g) - 2 * d0\ f * d0\ g = d0\ g * (1 - 2 * d0\ f)$  using False
  by (simp add: max-def algebra-simps)
  moreover have  $d0\ g * (1 - 2 * d0\ f) \leq d0\ f * (1 - 2 * d0\ f)$  using False hp
  by (auto intro: mult-right-mono)
  moreover have  $d0\ f * (1 - 2 * d0\ f) = 1 / 8 - 2 * (d0\ f - 1 / 4) ^ 2$ 
  by (simp add: power2-eq-square algebra-simps)
  moreover have  $0 \leq (d0\ f - 1 / 4) ^ 2$  by simp
  moreover have  $0 \leq d0\ g * (1 - 2 * d0\ f)$  using hp hq by (auto intro: mult-nonneg-nonneg)
  ultimately show ?thesis by argo
qed
hence  $- 1 / 8 \leq es\ f * es\ g * (d0\ f + d0\ g - \max(d0\ f)\ (d0\ g) - 2 * d0\ f * d0\ g) \wedge$ 
 $es\ f * es\ g * (d0\ f + d0\ g - \max(d0\ f)\ (d0\ g) - 2 * d0\ f * d0\ g) \leq 1 / 8$ 
using es-pm[of f] es-pm[of g] by (elim disjE) simp-all
moreover have  $- 1 / 8 \leq (\tau ^ 2 - \sigma * \tau) / 2 - pos-r\ ((\tau - \sigma) / 2) \wedge (\tau ^ 2 - \sigma * \tau) / 2 - pos-r\ ((\tau - \sigma) / 2) \leq 0$ 
proof (cases  $\tau \leq \sigma$ )
case True
have  $p: pos-r\ ((\tau - \sigma) / 2) = 0$  using True unfolding pos-r-def
by force
have  $\tau ^ 2 - \sigma * \tau = (\tau - \sigma / 2) ^ 2 - \sigma ^ 2 / 4$ 
by (auto simp: power2-eq-square algebra-simps)
moreover have  $0 \leq (\tau - \sigma / 2) ^ 2$  by simp
moreover have  $\sigma ^ 2 \leq 1$  using assms(7,8)
using power-le-one by blast
moreover have  $\tau * (\tau - \sigma) \leq 0$  using True assms(5)
by (force simp: mult-nonneg-nonpos)
moreover have  $\tau ^ 2 - \sigma * \tau = \tau * (\tau - \sigma)$ 
by (simp add: power2-eq-square algebra-simps)
ultimately show ?thesis unfolding p
by argo
next

```

```

case False
have  $p$ :  $\text{pos-}r \ ((\tau - \sigma) / 2) = (\tau - \sigma) / 2$  using False
  unfolding pos-r-def
  by auto
have  $\tau^2 - \sigma * \tau - (\tau - \sigma) =$ 
   $(\tau - (1 + \sigma) / 2)^2 - ((1 - \sigma) / 2)^2$ 
  by (simp add: power2-eq-square field-simps)
moreover have  $0 \leq (\tau - (1 + \sigma) / 2)^2$  by simp
moreover have  $(1 - \sigma)^2 \leq 1$  using assms(7,8)
  by (auto simp: power-le-one)
hence  $((1 - \sigma) / 2)^2 \leq 1 / 4$ 
  by (simp add: power-divide)
moreover have  $(\tau - \sigma) * (\tau - 1) \leq 0$  using False assms(6)
  by (simp add: mult-nonneg-nonpos)
moreover have  $\tau^2 - \sigma * \tau - (\tau - \sigma) =$ 
 $(\tau - \sigma) * (\tau - 1)$ 
  by (simp add: power2-eq-square algebra-simps)
ultimately show ?thesis unfolding  $p$ 
  by argo
qed
moreover have  $0 \leq (\eta - \eta^2) / 2 \wedge (\eta - \eta^2) / 2 \leq 1 / 8$ 
proof -
  have  $\eta - \eta^2 = \eta * (1 - \eta)$ 
  by (force simp: power2-eq-square algebra-simps)
moreover have  $0 \leq \eta * (1 - \eta)$  using assms(9,10)
  by (auto intro: mult-nonneg-nonneg)
moreover have  $\eta - \eta^2 = 1 / 4 - (\eta - 1 / 2)^2$ 
  by (fastforce simp: power2-eq-square algebra-simps)
moreover have  $0 \leq (\eta - 1 / 2)^2$  by simp
ultimately show ?thesis by argo
qed
ultimately show ?thesis unfolding Qc-def by linarith
qed

```

(5.12)-(5.14) for every real x .

theorem *R-decomp-bound-all*:

```

   $-1 / 2 \leq R-r \ x - x * (4 * \text{lam-}r + 2 * \text{lam-}r * \text{frac } x - 12 * \text{lam-}r * \text{frac}$ 
 $(\text{alpha-}r * x)) \wedge$ 
 $R-r \ x - x * (4 * \text{lam-}r + 2 * \text{lam-}r * \text{frac } x - 12 * \text{lam-}r * \text{frac } (\text{alpha-}r * x))$ 
 $\leq 13 / 8$ 

```

proof -

```

  have  $-1 / 2 \leq Qc \ (\text{frac } x) \ (\text{frac } (\text{alpha-}r * x)) \ (\text{frac } (2 * Hc-r * x)) \ (\text{frac } (2 * x))$ 
 $(\text{frac } (2 * (\text{lam-}r * x))) \wedge$ 
 $Qc \ (\text{frac } x) \ (\text{frac } (\text{alpha-}r * x)) \ (\text{frac } (2 * Hc-r * x)) \ (\text{frac } (2 * x)) \ (\text{frac } (2 * (\text{lam-}r * x)))$ 
 $\leq 13 / 8$ 
  by (rule Qc-bounds) (simp-all add: frac-lt-1 less-imp-le)
  thus ?thesis unfolding R-eq Fx-def by simp

```

qed

$R(x) = A x + B$ on a piece where the five floors are known.

lemma *R-on-piece*:

fixes $k1\ k2\ k3\ k4\ k5 :: nat$
assumes $real\ k1 \leq x < real\ k1 + 1$ $real\ k2 \leq \alpha\text{-}r * x$ $\alpha\text{-}r * x < real\ k2 + 1$
 $real\ k3 \leq 2 * Hc\text{-}r * x$ $2 * Hc\text{-}r * x < real\ k3 + 1$ $real\ k4 \leq 2 * x$ $2 * x < real\ k4 + 1$
 $real\ k5 \leq 2 * (lam\text{-}r * x)$ $2 * (lam\text{-}r * x) < real\ k5 + 1$
shows $R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - real\ k1) - 12 * lam\text{-}r * (\alpha\text{-}r * x - real\ k2))$
 $+ Qc\ (x - real\ k1)\ (\alpha\text{-}r * x - real\ k2)\ (2 * Hc\text{-}r * x - real\ k3)$
 $(2 * x - real\ k4)\ (2 * (lam\text{-}r * x) - real\ k5)$
unfolding *R-eq* *Fx-def* *frac-nat*[*OF* *assms*(1,2)] *frac-nat*[*OF* *assms*(3,4)] *frac-nat*[*OF* *assms*(5,6)]
frac-nat[*OF* *assms*(7,8)] *frac-nat*[*OF* *assms*(9,10)] ..

26.2 $\int_{-\{1/3\}}^{\{1/2\}} d(y) dy = 9/640$, the mean of P , and $|C| < 16$

theorem *integral-dfun-eq*: *interval-lebesgue-integral lborel* (*ereal* (1 / 3)) (*ereal* (1 / 2)) *dfun-r* = 9 / 640

proof –

have *e1*: *interval-lebesgue-integrable lborel* (*ereal* (1 / 3)) (*ereal* (43 / 120))
dfun-r $\wedge$
interval-lebesgue-integral lborel (*ereal* (1 / 3)) (*ereal* (43 / 120)) *dfun-r*
 $= (9 / 40 * (43 / 120) + 0 * (43 / 120) ^ 2 / 2 + 0 * (43 / 120) ^ 3 / 3) - (9 / 40 * (1 / 3) + 0 * (1 / 3) ^ 2 / 2 + 0 * (1 / 3) ^ 3 / 3)$

proof (*rule piece-quad*)

show $1 / 3 \leq (43 / 120 :: real)$ **by** *simp*

next

fix $y :: real$ **assume** $y \in \{1 / 3 < .. < 43 / 120\}$

thus *dfun-r* $y = 9 / 40 + 0 * y + 0 * y ^ 2$ **unfolding** *dfun-r-def* *pos-r-def* *alpha-r-def* *max-def*

by (*force simp*: *field-simps*)

qed

have *e2*: *interval-lebesgue-integrable lborel* (*ereal* (43 / 120)) (*ereal* (13 / 30))
dfun-r $\wedge$

interval-lebesgue-integral lborel (*ereal* (43 / 120)) (*ereal* (13 / 30)) *dfun-r*
 $= (13 / 10 * (13 / 30) + (- 3) * (13 / 30) ^ 2 / 2 + 0 * (13 / 30) ^ 3 / 3) - (13 / 10 * (43 / 120) + (- 3) * (43 / 120) ^ 2 / 2 + 0 * (43 / 120) ^ 3 / 3)$

proof (*rule piece-quad*)

show $43 / 120 \leq (13 / 30 :: real)$ **by** *simp*

next

```

    fix y :: real assume y ∈ {43 / 120 <.. $<13 / 30$ }
    thus dfun-r y = 13 / 10 + (- 3) * y + 0 * y ^ 2
    unfolding dfun-r-def pos-r-def alpha-r-def max-def by (force simp: field-simps)
  qed
  have e3: interval-lebesgue-integrable lborel (ereal (13 / 30)) (ereal (1 / 2)) dfun-r
  ^
    interval-lebesgue-integral lborel (ereal (13 / 30)) (ereal (1 / 2)) dfun-r
    = (0 * (1 / 2) + 0 * (1 / 2) ^ 2 / 2 + 0 * (1 / 2) ^ 3 / 3) - (0 * (13
/ 30) + 0 * (13 /
    30) ^ 2 / 2 + 0 * (13 / 30) ^ 3 / 3)
  proof (rule piece-quad)
    show 13 / 30 ≤ (1 / 2 :: real) by simp
  next
    fix y :: real assume y ∈ {13 / 30 <.. $<1 / 2$ }
    thus dfun-r y = 0 + 0 * y + 0 * y ^ 2 unfolding dfun-r-def pos-r-def
  alpha-r-def max-def
    by (force simp: field-simps)
  qed
  from chain[OF - - e1 e2] have e12: interval-lebesgue-integrable lborel (ereal (1
/
    3)) (ereal (13 / 30)) dfun-r ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (13 / 30)) dfun-r =
    9 / 40 * (43 / 120 - 1 / 3) + (13 / 10 * (13 / 30 - 43 / 120) - 3 / 2
* ((13 / 30) ^ 2 -
    (43 / 120) ^ 2))
  by (simp add: power2-eq-square power3-eq-cube)
  from chain[OF - - e12 e3] show ?thesis by (simp add: power2-eq-square power3-eq-cube)
qed

lemma fract-quad-piece: x ∈ {real n <.. $<real\ n + 1$ } ⇒
  frac x * (1 - frac x) = (- real n - real n ^ 2) + (1 + 2 * real n) * x + (- 1)
* x ^ 2
proof -
  assume x: x ∈ {real n <.. $<real\ n + 1$ }
  have f: frac x = x - real n using x frac-nat[of n x] by fastforce
  show ?thesis unfolding f by (auto simp: power2-eq-square algebra-simps)
qed

lemma fract-quad-integral:
  interval-lebesgue-integrable lborel (ereal 0) (ereal (real n)) (λx. frac x * (1 -
  frac x)) ∧
  interval-lebesgue-integral lborel (ereal 0) (ereal (real n)) (λx. frac x * (1 -
  frac x)) = real n / 6
proof (induction n)
  case 0
  show ?case
  by (fastforce simp: interval-lebesgue-integrable-def set-integrable-def
    interval-lebesgue-integral-def set-lebesgue-integral-def)
next

```

```

case (Suc n)
have piece: interval-lebesgue-integrable lborel (ereal (real n)) (ereal (real (Suc
n)))
  (λx. frac x * (1 - frac x)) ∧
  interval-lebesgue-integral lborel (ereal (real n)) (ereal (real (Suc n))) (λx. frac
x
  * (1 - frac x)) = 1 / 6
proof -
have e: real (Suc n) = real n + 1 by simp
have interval-lebesgue-integrable lborel (ereal (real n)) (ereal (real n +
1)) (λx. frac x * (1 - frac x)) ∧
  interval-lebesgue-integral lborel (ereal (real n)) (ereal (real n +
1)) (λx. frac x * (1 - frac x))
  = ((- real n - real n ^ 2) * (real n + 1) + (1 + 2 * real n) * (real n +
1) ^ 2 / 2 +
    (- 1) * (real n + 1) ^ 3 / 3)
    - ((- real n - real n ^ 2) * real n + (1 + 2 * real n) * real n ^ 2 / 2
+ (- 1) *
    real n ^ 3 / 3)
proof (rule piece-quad)
show real n ≤ real n + 1 by simp
next
fix x assume x ∈ {real n <..< real n + 1}
thus frac x * (1 - frac x) = (- real n - real n ^ 2) + (1 + 2 * real n) * x
+ (- 1) * x ^ 2
by (rule fract-quad-piece)
qed
moreover have ((- real n - real n ^ 2) * (real n + 1) + (1 + 2 * real n) *
(real n + 1) ^ 2 /
  2 + (- 1) * (real n + 1) ^ 3 / 3)
  - ((- real n - real n ^ 2) * real n + (1 + 2 * real n) * real n ^ 2 / 2
+ (- 1) *
    real n ^ 3 / 3) = 1 / 6
by (simp add: field-simps power2-eq-square power3-eq-cube)
ultimately show ?thesis unfolding e
by presburger
qed
from chain[OF - - Suc piece] show ?case by simp
qed

```

Scaling $x \mapsto c x$ in an integral from 0.

lemma *integral-scale*:

```

fixes g :: real ⇒ real
assumes c: 0 < c and b: 0 ≤ b and interval-lebesgue-integrable lborel (ereal 0)
(ereal (c
  * b)) g
shows interval-lebesgue-integral lborel (ereal 0) (ereal b) (λx. g (c * x)) =
  (1 / c) * interval-lebesgue-integral lborel (ereal 0) (ereal (c * b)) g
proof -

```

have $0 \leq c * b$ **using** $c\ b$ **by** *simp*
hence L : *interval-lebesgue-integral lborel (ereal 0) (ereal (c * b)) g =*
 $(\int x. \text{indicator } \{0 <..< c * b\} x *_R g x \ \partial \text{lborel})$
unfolding *interval-lebesgue-integral-def set-lebesgue-integral-def*
by *simp*
have R : *interval-lebesgue-integral lborel (ereal 0) (ereal b) (\lambda x. g (c * x)) =*
 $(\int x. \text{indicator } \{0 <..< b\} x *_R g (c * x) \ \partial \text{lborel})$
unfolding *interval-lebesgue-integral-def set-lebesgue-integral-def* **using** b
by *simp*
have A : $(\int x. \text{indicator } \{0 <..< c * b\} x *_R g x \ \partial \text{lborel}) =$
 $|c| *_R (\int x. \text{indicator } \{0 <..< c * b\} (0 + c * x) *_R g (0 + c *$
 $x) \ \partial \text{lborel})$
using *lborel-integral-real-affine*[*of c \lambda x. indicator \{0 <..< c * b\} x *_R g x 0*] c
by *blast*
have I : $\bigwedge x. (\text{indicator } \{0 <..< c * b\} (0 + c * x) :: \text{real}) = \text{indicator } \{0 <..< b\}$
 x **using** c
by (*auto simp: indicator-def zero-less-mult-iff mult-less-cancel-left*)
show *?thesis* **unfolding** $L\ R\ A\ I$ **using** c
by (*simp add: field-simps*)
qed

lemma *integrable-scale*:

fixes $g :: \text{real} \Rightarrow \text{real}$
assumes c : $0 < c$ **and** b : $0 \leq b$ **and** i : *interval-lebesgue-integrable lborel (ereal*
 $0)$ *(ereal*
 $(c * b))\ g$
shows *interval-lebesgue-integrable lborel (ereal 0) (ereal b) (\lambda x. g (c * x))*
proof –
have $0 \leq c * b$ **using** $c\ b$ **by** *simp*
hence $i0$: *integrable lborel (\lambda x. indicator \{0 <..< c * b\} x *_R g x)* **using** i
unfolding *interval-lebesgue-integrable-def set-integrable-def*
by *simp*
have $i1$: *integrable lborel (\lambda x. indicator \{0 <..< c * b\} (0 + c * x) *_R g (0 + c **
 $x))$
using *lborel-integrable-real-affine*[*OF i0, of c 0*] c
by *blast*
have I : $\bigwedge x. (\text{indicator } \{0 <..< c * b\} (0 + c * x) :: \text{real}) = \text{indicator } \{0 <..< b\}$
 x **using** c
by (*simp add: indicator-def zero-less-mult-iff mult-less-cancel-left*)
show *?thesis* **unfolding** *interval-lebesgue-integrable-def set-integrable-def* **using**
 $b\ i1$ **unfolding** I
by *simp*
qed

theorem *Pper-mean-eq*: *interval-lebesgue-integral lborel (ereal 0) (ereal 40) Pper*
 $/\ 40 =$
 $2923 / 240$
proof –


```

let ?u = λx. frac x * (1 - frac x)
have u40: interval-lebesgue-integral lborel (ereal 0) (ereal 40) ?u = 40 / 6
  using fract-quad-integral[of 40] by simp
have u3: interval-lebesgue-integral lborel (ereal 0) (ereal 40) (λx. ?u (alpha-r *
x)) =
  (1 / alpha-r) * (3 / 6)
  using integral-scale[of alpha-r 40 ?u] fract-quad-integral[of 3] by (simp add:
alpha-r-def)
have Pper-eq: Pper = (λx. 74 * ?u (alpha-r * x) - lam-r * ?u x)
  by (simp add: Pper-def fun-eq-iff algebra-simps)
have interval-lebesgue-integral lborel (ereal 0) (ereal 40) Pper =
  74 * ((1 / alpha-r) * (3 / 6)) - lam-r * (40 / 6)
  unfolding Pper-eq
proof -
  have j40: interval-lebesgue-integrable lborel (ereal 0) (ereal 40) ?u
    using fract-quad-integral[of 40]
    by simp
  have interval-lebesgue-integrable lborel (ereal 0) (ereal (alpha-r * 40)) ?u
    using fract-quad-integral[of 3]
    by (simp add: alpha-r-def)
  hence interval-lebesgue-integrable lborel (ereal 0) (ereal 40) (λx. ?u (alpha-r *
x))
    using integrable-scale[of alpha-r 40 ?u]
    using alpha-r-def by linarith
  hence i1: interval-lebesgue-integrable lborel (ereal 0) (ereal 40) (λx. 74 *
?u (alpha-r * x))
    by fast
  have i2: interval-lebesgue-integrable lborel (ereal 0) (ereal 40) (λx. lam-r * ?u
x)
    using j40
    by fast
  have D: interval-lebesgue-integral lborel (ereal 0) (ereal 40) (λx. 74 * ?u (alpha-r
*
x) - lam-r * ?u x)
    = interval-lebesgue-integral lborel (ereal 0) (ereal 40) (λx. 74 * ?u (alpha-r *
x)) - interval-lebesgue-integral lborel (ereal 0) (ereal 40) (λx. lam-r * ?u x)
    by (rule interval-lebesgue-integral-diff(2)[OF i1 i2])
  show interval-lebesgue-integral lborel (ereal 0) (ereal 40) (λx. 74 * ?u (alpha-r
*
x) - lam-r * ?u x) = 74 * ((1 / alpha-r) * (3 / 6)) - lam-r * (40 / 6)
    unfolding D using u40 u3
    by simp
qed
thus ?thesis by (simp add: alpha-r-def lam-r-def)
qed

```

$|G_0(v)| \leq 81/5000$ on $[0, 1]$ and $|C(x)| < 16$.

lemma G_0 -r-bound:

assumes $0 \leq v \leq 1$

shows $|G0-r \ v| \leq 81 / 5000$
proof –
 define w where $w = 2 * v - 1$
 define u where $u = |w|$
 have $u0: 0 \leq u$ and $u1: u \leq 1$ using *assms* by (auto simp: u-def w-def)
 — cubic bound $u - u^3 \leq 243/625$ on $[0,1]$, via $r = 577/1000 \approx 1/\sqrt{3}$
 define $r :: real$ where $r = 577 / 1000$
 have $u^3 - u + 243 / 625 = (u - r)^2 * (u + 2 * r) + (3 * r^2 - 1) * u +$
 $(243 /$
 $625 - 2 * r^3)$
 by (simp add: algebra-simps power2-eq-square power3-eq-cube)
 moreover have $0 \leq (u - r)^2 * (u + 2 * r)$ using $u0$ by (simp add: r-def)
 moreover have $(3 * r^2 - 1) * u \geq 3 * r^2 - 1$
proof –
 have $3 * r^2 - 1 \leq 0$ by (simp add: r-def power2-eq-square)
 thus ?thesis using $u1$ by (metis mult-left-mono-neg mult.right-neutral)
qed
 moreover have $0 \leq (3 * r^2 - 1) + (243 / 625 - 2 * r^3)$
 by (simp add: r-def power2-eq-square power3-eq-cube)
 ultimately have $cub: u - u^3 \leq 243 / 625$ by linarith
 have $w - w^3 = 4 * (v * (1 - v) * (2 * v - 1))$ unfolding w-def
 by (simp add: power3-eq-cube algebra-simps)
 hence $G: G0-r \ v = (w - w^3) / 24$ by (simp add: G0-r-def)
 have $|w - w^3| = u - u^3$
proof –
 have $w - w^3 = w * (1 - w^2)$
 by (simp add: algebra-simps power2-eq-square power3-eq-cube)
 moreover have $0 \leq 1 - w^2$ using $u1$ by (simp add: u-def abs-square-le-1)
 ultimately have $|w - w^3| = u * (1 - w^2)$ by (simp add: abs-mult u-def)
 also have $w^2 = u^2$ by (simp add: u-def)
 finally show ?thesis by (simp add: algebra-simps power2-eq-square power3-eq-cube)
qed
 thus ?thesis unfolding G using cub by simp
qed

theorem *Cper-bound*: $|Cper \ x| < 16$
proof –
 have $b1: |G0-r \ (frac \ (alpha-r * x))| \leq 81 / 5000$
 by (rule G0-r-bound) (auto simp: frac-ge-0 frac-lt-1 less-imp-le)
 have $b2: |G0-r \ (frac \ x)| \leq 81 / 5000$
 by (rule G0-r-bound) (auto simp: frac-ge-0 frac-lt-1 less-imp-le)
 have $|Cper \ x| \leq 74 / alpha-r * |G0-r \ (frac \ (alpha-r * x))| + lam-r *$
 $|G0-r \ (frac \ x)|$
 unfolding Cper-def alpha-r-def lam-r-def by (simp add: abs-mult)
 also have $\dots \leq 74 / alpha-r * (81 / 5000) + lam-r * (81 / 5000)$
 using $b1 \ b2$ by (intro add-mono mult-left-mono) (auto simp: alpha-r-def
 lam-r-def)
 also have $\dots < 16$ by (simp add: alpha-r-def lam-r-def)
 finally show ?thesis .

qed

26.3 (5.10): $\int -\{1/3\}^{\sim\{2\lambda\}}$ *outerIntegrand* = *I-out* by 11 affine pieces

On each piece *outerIntegrand* $y = c0 + c1 y$; $\lfloor 1/y \rfloor$ is constant (fact *fl*). The pieces were generated from Lean's *gen/gen-outer.py* (29 pieces, merged where the formula agrees).

lemma *pos-eq1*: $0 \leq v \implies \text{pos-r } v = v$ **unfolding** *pos-r-def* **by** *force*

lemma *pos-eq0*: $v \leq 0 \implies \text{pos-r } v = 0$ **unfolding** *pos-r-def* **by** *argo*

lemma *outer-piece-0*: $y \in \{(1/3) <..<(43/120)\} \implies$

outer-integrand $y = (279/40) + (-9) * y$

proof –

assume *y*: $y \in \{(1/3) <..<(43/120)\}$

have *b*: $(1/3) < y < (43/120)$ **using** *y* **by** *auto*

have $0 < y \wedge \text{of-int } 2 * y \leq 1 \wedge 1 < (\text{of-int } 2 + 1) * y$ **using** *b* **by** *auto*

hence *fl*: $\text{floor } (1/y) = 2$ **using** *floor-inv-eq* **by** *blast*

have *s*: $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$

$\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) + \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y))))$

by (*auto simp: numeral-eq-Suc atLeastAtMostSuc-conv*)

have *cA*: $1/3 < y \wedge y < 1/2$ **using** *b* **by** *linarith*

have *mA*: $\min \text{alpha-r } (1 - 2 * y) = \text{alpha-r}$

by (*rule min-absorb1*) (*use b in* $\langle \text{unfold alpha-r-def, linarith} \rangle$)

have *pA*: $\text{pos-r } (1 + \text{alpha-r} - 3 * y) = 1 + \text{alpha-r} - 3 * y$

by (*rule pos-eq1*) (*use b in* $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)

have *pD*: $\text{pos-r } ((1 + 4 * \text{alpha-r} - 3 * y) - (1 + \text{alpha-r} - 3 * y)) =$

$(1 + 4 * \text{alpha-r} - 3 * y) - (1 + \text{alpha-r} - 3 * y)$

by (*rule pos-eq1*) (*use b in* $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)

have *s1*: $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) = 2 * \text{lam-r} - \text{real } 1 * y$

by (*rule pos-eq1*) (*use b in* $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)

have *s2*: $\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) = 2 * \text{lam-r} - \text{real } 2 * y$

by (*rule pos-eq1*) (*use b in* $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)

have *s3*: $\text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) = 2 * \text{lam-r} - \text{real } 3 * y$

by (*rule pos-eq1*) (*use b in* $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)

have *s4*: $\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) = 2 * \text{lam-r} - \text{real } 4 * y$

by (*rule pos-eq1*) (*use b in* $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)

have *s5*: $\text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y) = 2 * \text{lam-r} - \text{real } 5 * y$

by (*rule pos-eq1*) (*use b in* $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)

show *?thesis unfolding outer-integrand-def fl s R0-r-def dfun-r-def*
unfolding *if-P[OF cA] mA pA pD s1 s2 s3 s4 s5*
by (*simp add: alpha-r-def lam-r-def algebra-simps*)
qed

lemma *outer-piece-1: $y \in \{(43 / 120) <..< (37 / 100)\} \implies$*
*outer-integrand $y = (451 / 40) + (- 21) * y$*
proof –
assume *y: $y \in \{(43 / 120) <..< (37 / 100)\}$*
have *b: $(43 / 120) < y < (37 / 100)$ using y by auto*
have *0 < y $\wedge$ of-int 2 * y $\leq$ 1 $\wedge$ 1 < (of-int 2 + 1) * y using b by auto*
hence *fl: floor (1 / y) = 2 using floor-inv-eq by blast*
have *s: $(\sum j = 1..5. pos-r (2 * lam-r - real j * y)) =$*
*pos-r (2 * lam-r - real 1 * y) + (pos-r (2 * lam-r - real 2 * y) + (pos-r (2*
** lam-r - real 3 **
*y) + (pos-r (2 * lam-r - real 4 * y) + pos-r (2 * lam-r - real 5 * y))))*
by (*auto simp: numeral-eq-Suc atLeastAtMostSuc-conv*)
have *cA: $1 / 3 < y \wedge y < 1 / 2$ using b by linarith*
have *mA: min alpha-r (1 - 2 * y) = alpha-r*
by (*rule min-absorb1*) (*use b in <unfold alpha-r-def, linarith>*)
have *pA: pos-r (1 + alpha-r - 3 * y) = 0*
by (*rule pos-eq0*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have *pD: pos-r ((1 + 4 * alpha-r - 3 * y) - 0) = (1 + 4 * alpha-r - 3 * y)*
- 0
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have *s1: pos-r (2 * lam-r - real 1 * y) = 2 * lam-r - real 1 * y*
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have *s2: pos-r (2 * lam-r - real 2 * y) = 2 * lam-r - real 2 * y*
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have *s3: pos-r (2 * lam-r - real 3 * y) = 2 * lam-r - real 3 * y*
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have *s4: pos-r (2 * lam-r - real 4 * y) = 2 * lam-r - real 4 * y*
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have *s5: pos-r (2 * lam-r - real 5 * y) = 2 * lam-r - real 5 * y*
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
show *?thesis unfolding outer-integrand-def fl s R0-r-def dfun-r-def*
unfolding *if-P[OF cA] mA pA pD s1 s2 s3 s4 s5*
by (*simp add: alpha-r-def lam-r-def algebra-simps*)
qed

lemma *outer-piece-2: $y \in \{(37 / 100) <..< (13 / 30)\} \implies$*
*outer-integrand $y = (377 / 40) + (- 16) * y$*

proof –

assume $y: y \in \{(37 / 100) <..< (13 / 30)\}$
have $b: (37 / 100) < y < (13 / 30)$ **using** y **by** *auto*
have $0 < y \wedge \text{of-int } 2 * y \leq 1 \wedge 1 < (\text{of-int } 2 + 1) * y$ **using** b **by** *auto*
hence $\text{fl: floor } (1 / y) = 2$ **using** *floor-inv-eq* **by** *blast*
have $s: (\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$
 $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2$
 $* \text{lam-r} - \text{real } 3 *$
 $y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) + \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y))))$
by (*auto simp: numeral-eq-Suc atLeastAtMostSuc-conv*)
have $cA: 1 / 3 < y \wedge y < 1 / 2$ **using** b **by** *linarith*
have $mA: \min \text{alpha-r } (1 - 2 * y) = \text{alpha-r}$
by (*rule min-absorb1*) (*use b in <unfold alpha-r-def, linarith>*)
have $pA: \text{pos-r } (1 + \text{alpha-r} - 3 * y) = 0$
by (*rule pos-eq0*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have $pD: \text{pos-r } ((1 + 4 * \text{alpha-r} - 3 * y) - 0) = (1 + 4 * \text{alpha-r} - 3 * y)$
 $- 0$
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have $s1: \text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) = 2 * \text{lam-r} - \text{real } 1 * y$
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have $s2: \text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) = 2 * \text{lam-r} - \text{real } 2 * y$
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have $s3: \text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) = 2 * \text{lam-r} - \text{real } 3 * y$
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have $s4: \text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) = 2 * \text{lam-r} - \text{real } 4 * y$
by (*rule pos-eq1*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
have $s5: \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y) = 0$
by (*rule pos-eq0*) (*use b in <unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,*
linarith>)
show *?thesis* **unfolding** *outer-integrand-def fl s R0-r-def dfun-r-def*
unfolding *if-P[OF cA] mA pA pD s1 s2 s3 s4 s5*
by (*simp add: alpha-r-def lam-r-def algebra-simps*)

qed

lemma *outer-piece-3*: $y \in \{(13 / 30) <..< (37 / 80)\} \implies$

outer-integrand $y = (429 / 40) + (-19) * y$

proof –

assume $y: y \in \{(13 / 30) <..< (37 / 80)\}$
have $b: (13 / 30) < y < (37 / 80)$ **using** y **by** *auto*
have $0 < y \wedge \text{of-int } 2 * y \leq 1 \wedge 1 < (\text{of-int } 2 + 1) * y$ **using** b **by** *auto*
hence $\text{fl: floor } (1 / y) = 2$ **using** *floor-inv-eq* **by** *blast*
have $s: (\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$
 $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2$

```

* lam-r - real 3 *
  y) + (pos-r (2 * lam-r - real 4 * y) + pos-r (2 * lam-r - real 5 * y))))
  by (auto simp: numeral-eq-Suc atLeastAtMostSuc-conv)
have cA: 1 / 3 < y ∧ y < 1 / 2 using b by linarith
have mA: min alpha-r (1 - 2 * y) = alpha-r
  by (rule min-absorb1) (use b in ⟨unfold alpha-r-def, linarith⟩)
have pA: pos-r (1 + alpha-r - 3 * y) = 0
  by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
    linarith⟩)
have pD: pos-r ((1 + 4 * alpha-r - 3 * y) - 0) = 0
  by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
    linarith⟩)
have s1: pos-r (2 * lam-r - real 1 * y) = 2 * lam-r - real 1 * y
  by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
    linarith⟩)
have s2: pos-r (2 * lam-r - real 2 * y) = 2 * lam-r - real 2 * y
  by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
    linarith⟩)
have s3: pos-r (2 * lam-r - real 3 * y) = 2 * lam-r - real 3 * y
  by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
    linarith⟩)
have s4: pos-r (2 * lam-r - real 4 * y) = 2 * lam-r - real 4 * y
  by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
    linarith⟩)
have s5: pos-r (2 * lam-r - real 5 * y) = 0
  by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
    linarith⟩)
show ?thesis unfolding outer-integrand-def fl s R0-r-def dfun-r-def
  unfolding if-P[OF cA] mA pA pD s1 s2 s3 s4 s5
  by (simp add: alpha-r-def lam-r-def algebra-simps)
qed

```

lemma *outer-piece-4*: $y \in \{(37 / 80) <..< (1 / 2)\} \implies$

outer-integrand $y = (17 / 4) + (- 5) * y$

proof –

```

assume y: y ∈ {(37 / 80) <..< (1 / 2)}
have b: (37 / 80) < y ∧ y < (1 / 2) using y by auto
have 0 < y ∧ of-int 2 * y ≤ 1 ∧ 1 < (of-int 2 + 1) * y using b by auto
hence fl: floor (1 / y) = 2 using floor-inv-eq by blast
have s: (∑ j = 1..5. pos-r (2 * lam-r - real j * y)) =
  pos-r (2 * lam-r - real 1 * y) + (pos-r (2 * lam-r - real 2 * y) + (pos-r (2
* lam-r - real 3 *
  y) + (pos-r (2 * lam-r - real 4 * y) + pos-r (2 * lam-r - real 5 * y))))
  by (auto simp: numeral-eq-Suc atLeastAtMostSuc-conv)
have cA: 1 / 3 < y ∧ y < 1 / 2 using b by linarith
have mA: min alpha-r (1 - 2 * y) = 1 - 2 * y
  by (rule min-absorb2) (use b in ⟨unfold alpha-r-def, linarith⟩)
have pA: pos-r (1 + alpha-r - 3 * y) = 0
  by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,

```

$\text{linarith}\rangle$
have pD : $\text{pos-}r \ ((1 + 4 * \text{alpha-}r - 3 * y) - 0) = 0$
by (rule pos-eq0) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
have $s1$: $\text{pos-}r \ (2 * \text{lam-}r - \text{real } 1 * y) = 2 * \text{lam-}r - \text{real } 1 * y$
by (rule pos-eq1) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
have $s2$: $\text{pos-}r \ (2 * \text{lam-}r - \text{real } 2 * y) = 2 * \text{lam-}r - \text{real } 2 * y$
by (rule pos-eq1) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
have $s3$: $\text{pos-}r \ (2 * \text{lam-}r - \text{real } 3 * y) = 2 * \text{lam-}r - \text{real } 3 * y$
by (rule pos-eq1) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
have $s4$: $\text{pos-}r \ (2 * \text{lam-}r - \text{real } 4 * y) = 0$
by (rule pos-eq0) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
have $s5$: $\text{pos-}r \ (2 * \text{lam-}r - \text{real } 5 * y) = 0$
by (rule pos-eq0) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
show $?thesis \text{ unfolding outer-integrand-def fl } s \text{ R0-}r\text{-def dfun-}r\text{-def}$
 $\text{unfolding if-}P[OF \text{ cA}] \text{ mA } pA \text{ pD } s1 \text{ s2 } s3 \text{ s4 } s5$
by ($\text{simp add: alpha-}r\text{-def lam-}r\text{-def algebra-simps}$)
qed

lemma outer-piece-5 : $y \in \{(1 / 2) <..<(43 / 80)\} \implies$
 $\text{outer-integrand } y = (51 / 10) + (-3) * y$
proof –
assume y : $y \in \{(1 / 2) <..<(43 / 80)\}$
have b : $(1 / 2) < y \wedge y < (43 / 80)$ **using** y **by** auto
have $0 < y \wedge \text{of-int } 1 * y \leq 1 \wedge 1 < (\text{of-int } 1 + 1) * y$ **using** b **by** auto
hence fl : $\text{floor } (1 / y) = 1$ **using** floor-inv-eq **by** blast
have s : $(\sum j = 1..5. \text{pos-}r \ (2 * \text{lam-}r - \text{real } j * y)) =$
 $\text{pos-}r \ (2 * \text{lam-}r - \text{real } 1 * y) + (\text{pos-}r \ (2 * \text{lam-}r - \text{real } 2 * y) + (\text{pos-}r \ (2$
 $* \text{lam-}r - \text{real } 3 * y) + (\text{pos-}r \ (2 * \text{lam-}r - \text{real } 4 * y) + \text{pos-}r \ (2 * \text{lam-}r - \text{real } 5 * y)))$
by ($\text{auto simp: numeral-eq-Suc atLeastAtMostSuc-conv}$)
have nA : $\neg (1 / 3 < y \wedge y < 1 / 2)$ **using** b **by** linarith
have cB : $1 / 2 < y \wedge y < 1$ **using** b **by** linarith
have mB : $\text{min alpha-}r \ (1 - y) = \text{alpha-}r$
by (rule min-absorb1) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def, linarith}\rangle$)
have $pB1$: $\text{pos-}r \ (1 + \text{alpha-}r - 2 * y) = 1 + \text{alpha-}r - 2 * y$
by (rule pos-eq1) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
have $pB2$: $\text{pos-}r \ (1 + 4 * \text{alpha-}r - 2 * y) = 1 + 4 * \text{alpha-}r - 2 * y$
by (rule pos-eq1) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)
have $s1$: $\text{pos-}r \ (2 * \text{lam-}r - \text{real } 1 * y) = 2 * \text{lam-}r - \text{real } 1 * y$
by (rule pos-eq1) ($\text{use } b \text{ in } \langle \text{unfold alpha-}r\text{-def lam-}r\text{-def of-nat-numeral of-nat-1, linarith}\rangle$)

have $s2$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) = 2 * \text{lam-r} - \text{real } 2 * y$
by (rule pos-eq1) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have $s3$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) = 2 * \text{lam-r} - \text{real } 3 * y$
by (rule pos-eq1) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have $s4$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) = 0$
by (rule pos-eq0) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have $s5$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y) = 0$
by (rule pos-eq0) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
show ?thesis **unfolding** outer-integrand-def fl s R0-r-def dfun-r-def
unfolding if-not-P[OF nA] if-P[OF cB] mB pB1 pB2 s1 s2 s3 s4 s5
by (simp add: alpha-r-def lam-r-def algebra-simps)
qed

lemma outer-piece-6: $y \in \{(43 / 80) <..< (37 / 60)\} \implies$
 $\text{outer-integrand } y = (231 / 20) + (-15) * y$

proof –

assume y : $y \in \{(43 / 80) <..< (37 / 60)\}$
have b : $(43 / 80) < y < (37 / 60)$ **using** y **by** auto
have $0 < y \wedge \text{of-int } 1 * y \leq 1 \wedge 1 < (\text{of-int } 1 + 1) * y$ **using** b **by** auto
hence fl: $\text{floor } (1 / y) = 1$ **using** floor-inv-eq **by** blast
have s : $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$
 $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2$
 $* \text{lam-r} - \text{real } 3 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) + \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y)))$
by (auto simp: numeral-eq-Suc atLeastAtMostSuc-conv)
have nA: $\neg (1 / 3 < y \wedge y < 1 / 2)$ **using** b **by** linarith
have cB: $1 / 2 < y \wedge y < 1$ **using** b **by** linarith
have mB: $\min \text{alpha-r } (1 - y) = \text{alpha-r}$
by (rule min-absorb1) (use b in $\langle \text{unfold alpha-r-def, linarith} \rangle$)
have pB1: $\text{pos-r } (1 + \text{alpha-r} - 2 * y) = 0$
by (rule pos-eq0) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have pB2: $\text{pos-r } (1 + 4 * \text{alpha-r} - 2 * y) = 1 + 4 * \text{alpha-r} - 2 * y$
by (rule pos-eq1) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have s1: $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) = 2 * \text{lam-r} - \text{real } 1 * y$
by (rule pos-eq1) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have s2: $\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) = 2 * \text{lam-r} - \text{real } 2 * y$
by (rule pos-eq1) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have s3: $\text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) = 2 * \text{lam-r} - \text{real } 3 * y$
by (rule pos-eq1) (use b in $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith} \rangle$)
have s4: $\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) = 0$

by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
have s5: pos-r (2 * lam-r - real 5 * y) = 0
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
show ?thesis **unfolding** outer-integrand-def fl s R0-r-def dfun-r-def
unfolding if-not-P[OF nA] if-P[OF cB] mB pB1 pB2 s1 s2 s3 s4 s5
by (simp add: alpha-r-def lam-r-def algebra-simps)
qed

lemma outer-piece-7: $y \in \{(37 / 60) <..<(13 / 20)\} \implies$
 $\text{outer-integrand } y = (97 / 10) + (-12) * y$

proof –
assume y: $y \in \{(37 / 60) <..<(13 / 20)\}$
have b: $(37 / 60) < y \wedge y < (13 / 20)$ **using** y **by** auto
have 0 < y $\wedge$ of-int 1 * y $\leq$ 1 $\wedge$ 1 < (of-int 1 + 1) * y **using** b **by** auto
hence fl: floor (1 / y) = 1 **using** floor-inv-eq **by** blast
have s: $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$
 $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2$
 $* \text{lam-r} - \text{real } 3 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) + \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y)))$
by (auto simp: numeral-eq-Suc atLeastAtMostSuc-conv)
have nA: $\neg (1 / 3 < y \wedge y < 1 / 2)$ **using** b **by** linarith
have cB: $1 / 2 < y \wedge y < 1$ **using** b **by** linarith
have mB: min alpha-r (1 - y) = alpha-r
by (rule min-absorb1) (use b in ⟨unfold alpha-r-def, linarith⟩)
have pB1: pos-r (1 + alpha-r - 2 * y) = 0
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
have pB2: pos-r (1 + 4 * alpha-r - 2 * y) = 1 + 4 * alpha-r - 2 * y
by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
have s1: pos-r (2 * lam-r - real 1 * y) = 2 * lam-r - real 1 * y
by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
have s2: pos-r (2 * lam-r - real 2 * y) = 2 * lam-r - real 2 * y
by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
have s3: pos-r (2 * lam-r - real 3 * y) = 0
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
have s4: pos-r (2 * lam-r - real 4 * y) = 0
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
have s5: pos-r (2 * lam-r - real 5 * y) = 0
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1, linarith⟩)
show ?thesis **unfolding** outer-integrand-def fl s R0-r-def dfun-r-def
unfolding if-not-P[OF nA] if-P[OF cB] mB pB1 pB2 s1 s2 s3 s4 s5

by (simp add: alpha-r-def lam-r-def algebra-simps)
qed

lemma outer-piece-8: $y \in \{(13 / 20) <..<(37 / 40)\} \implies$
outer-integrand $y = (42 / 5) + (- 10) * y$

proof –

assume $y: y \in \{(13 / 20) <..<(37 / 40)\}$
have $b: (13 / 20) < y < (37 / 40)$ using y by auto
have $0 < y \wedge \text{of-int } 1 * y \leq 1 \wedge 1 < (\text{of-int } 1 + 1) * y$ using b by auto
hence $\text{fl}: \text{floor } (1 / y) = 1$ using floor-inv-eq by blast
have $s: (\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$
 $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2$
 $* \text{lam-r} - \text{real } 3 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) + \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y))))$
by (auto simp: numeral-eq-Suc atLeastAtMostSuc-conv)
have $nA: \neg (1 / 3 < y \wedge y < 1 / 2)$ using b by linarith
have $cB: 1 / 2 < y \wedge y < 1$ using b by linarith
have $mB: \min \text{alpha-r } (1 - y) = \text{alpha-r}$
by (rule min-absorb1) (use b in ⟨unfold alpha-r-def, linarith⟩)
have $pB1: \text{pos-r } (1 + \text{alpha-r} - 2 * y) = 0$
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
linarith⟩)
have $pB2: \text{pos-r } (1 + 4 * \text{alpha-r} - 2 * y) = 0$
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
linarith⟩)
have $s1: \text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) = 2 * \text{lam-r} - \text{real } 1 * y$
by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
linarith⟩)
have $s2: \text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) = 2 * \text{lam-r} - \text{real } 2 * y$
by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
linarith⟩)
have $s3: \text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) = 0$
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
linarith⟩)
have $s4: \text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) = 0$
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
linarith⟩)
have $s5: \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y) = 0$
by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
linarith⟩)
show ?thesis unfolding outer-integrand-def fl s R0-r-def dfun-r-def
unfolding if-not-P[OF nA] if-P[OF cB] mB pB1 pB2 s1 s2 s3 s4 s5
by (simp add: alpha-r-def lam-r-def algebra-simps)

qed

lemma outer-piece-9: $y \in \{(37 / 40) <..<1\} \implies$ outer-integrand $y = 1 + (- 2)$
 $* y$

proof –

assume $y: y \in \{(37 / 40) <..<1\}$

have b : $(37 / 40) < y \wedge y < 1$ **using** y **by** *auto*
have $0 < y \wedge \text{of-int } 1 * y \leq 1 \wedge 1 < (\text{of-int } 1 + 1) * y$ **using** b **by** *auto*
hence $\text{fl: floor } (1 / y) = 1$ **using** *floor-inv-eq* **by** *blast*
have s : $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$
 $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2$
 $* \text{lam-r} - \text{real } 3 *$
 $y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) + \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y))))$
by (*auto simp: numeral-eq-Suc atLeastAtMostSuc-conv*)
have nA : $\neg (1 / 3 < y \wedge y < 1 / 2)$ **using** b **by** *linarith*
have cB : $1 / 2 < y \wedge y < 1$ **using** b **by** *linarith*
have mB : $\min \text{alpha-r } (1 - y) = 1 - y$
by (*rule min-absorb2*) (*use* b **in** $\langle \text{unfold alpha-r-def, linarith} \rangle$)
have $pB1$: $\text{pos-r } (1 + \text{alpha-r} - 2 * y) = 0$
by (*rule pos-eq0*) (*use* b **in** $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,$
 $\text{linarith} \rangle$)
have $pB2$: $\text{pos-r } (1 + 4 * \text{alpha-r} - 2 * y) = 0$
by (*rule pos-eq0*) (*use* b **in** $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,$
 $\text{linarith} \rangle$)
have $s1$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) = 2 * \text{lam-r} - \text{real } 1 * y$
by (*rule pos-eq1*) (*use* b **in** $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,$
 $\text{linarith} \rangle$)
have $s2$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) = 0$
by (*rule pos-eq0*) (*use* b **in** $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,$
 $\text{linarith} \rangle$)
have $s3$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) = 0$
by (*rule pos-eq0*) (*use* b **in** $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,$
 $\text{linarith} \rangle$)
have $s4$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) = 0$
by (*rule pos-eq0*) (*use* b **in** $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,$
 $\text{linarith} \rangle$)
have $s5$: $\text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y) = 0$
by (*rule pos-eq0*) (*use* b **in** $\langle \text{unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,$
 $\text{linarith} \rangle$)
show *?thesis* **unfolding** *outer-integrand-def fl s R0-r-def dfun-r-def*
unfolding *if-not-P[OF nA] if-P[OF cB] mB pB1 pB2 s1 s2 s3 s4 s5*
by (*simp add: alpha-r-def lam-r-def algebra-simps*)

qed

lemma *outer-piece-10*: $y \in \{1 < .. < (37 / 20)\} \implies$

outer-integrand $y = (37 / 20) + (-1) * y$

proof –

assume y : $y \in \{1 < .. < (37 / 20)\}$
have b : $1 < y \wedge y < (37 / 20)$ **using** y **by** *auto*
have $0 < y \wedge \text{of-int } 0 * y \leq 1 \wedge 1 < (\text{of-int } 0 + 1) * y$ **using** b **by** *auto*
hence $\text{fl: floor } (1 / y) = 0$ **using** *floor-inv-eq* **by** *blast*
have s : $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * y)) =$
 $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) + (\text{pos-r } (2$
 $* \text{lam-r} - \text{real } 3 *$
 $y) + (\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) + \text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y))))$

```

  by (auto simp: numeral-eq-Suc atLeastAtMostSuc-conv)
  have nA:  $\neg (1 / 3 < y \wedge y < 1 / 2)$  using b by linarith
  have nB:  $\neg (1 / 2 < y \wedge y < 1)$  using b by linarith
  have s1:  $\text{pos-r } (2 * \text{lam-r} - \text{real } 1 * y) = 2 * \text{lam-r} - \text{real } 1 * y$ 
    by (rule pos-eq1) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
      linarith⟩)
  have s2:  $\text{pos-r } (2 * \text{lam-r} - \text{real } 2 * y) = 0$ 
    by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
      linarith⟩)
  have s3:  $\text{pos-r } (2 * \text{lam-r} - \text{real } 3 * y) = 0$ 
    by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
      linarith⟩)
  have s4:  $\text{pos-r } (2 * \text{lam-r} - \text{real } 4 * y) = 0$ 
    by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
      linarith⟩)
  have s5:  $\text{pos-r } (2 * \text{lam-r} - \text{real } 5 * y) = 0$ 
    by (rule pos-eq0) (use b in ⟨unfold alpha-r-def lam-r-def of-nat-numeral of-nat-1,
      linarith⟩)
  show ?thesis unfolding outer-integrand-def fl s R0-r-def dfun-r-def
    unfolding if-not-P[OF nA] if-not-P[OF nB] s1 s2 s3 s4 s5
    by (simp add: alpha-r-def lam-r-def algebra-simps)
qed

```

The pieces chained from $1/3$ to $2\lambda = 37/20$.

theorem *outer-integral-value:*

```

  interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (2 * lam-r)) outer-integrand
=
  of-rat Iout
proof -
  have interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (43 /
    120)) outer-integrand  $\wedge$ 
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (43 / 120)) outer-integrand
=
  ((279 / 40) * (43 / 120) + (- 9) * (43 / 120) ^ 2 / 2 + 0 * (43 / 120)
  ^ 3 / 3) - ((279 /
    40) * (1 / 3) + (- 9) * (1 / 3) ^ 2 / 2 + 0 * (1 / 3) ^ 3 / 3)
  by (rule piece-quad) (simp-all add: outer-piece-0)
  hence c0: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (43 /
    120)) outer-integrand  $\wedge$ 
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (43 / 120)) outer-integrand
=
  (309 / 3200)
  by (simp add: power2-eq-square power3-eq-cube)
  have o1: interval-lebesgue-integrable lborel (ereal (43 / 120)) (ereal (37 /
    100)) outer-integrand  $\wedge$ 
    interval-lebesgue-integral lborel (ereal (43 / 120)) (ereal (37 / 100)) outer-integrand
=
  ((451 / 40) * (37 / 100) + (- 21) * (37 / 100) ^ 2 / 2 + 0 * (37 / 100)
  ^ 3 / 3) - ((451 /

```

```

      40) * (43 / 120) + (- 21) * (43 / 120) ^ 2 / 2 + 0 * (43 / 120) ^ 3 / 3)
    by (rule piece-quad) (simp-all add: outer-piece-1)
  have c1: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (37 /
    100)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (37 / 100)) outer-integrand
  =
    (8333 / 60000)
  using chain[OF - - c0 o1] by (simp add: power2-eq-square power3-eq-cube)
  have o2: interval-lebesgue-integrable lborel (ereal (37 / 100)) (ereal (13 /
    30)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (37 / 100)) (ereal (13 / 30)) outer-integrand
  =
    ((377 / 40) * (13 / 30) + (- 16) * (13 / 30) ^ 2 / 2 + 0 * (13 / 30) ^
    3 / 3) - ((377 /
    40) * (37 / 100) + (- 16) * (37 / 100) ^ 2 / 2 + 0 * (37 / 100) ^ 3 / 3)
  by (rule piece-quad) (simp-all add: outer-piece-2)
  have c2: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (13 /
    30)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (13 / 30)) outer-integrand
  =
    (2959 / 9000)
  using chain[OF - - c1 o2] by (simp add: power2-eq-square power3-eq-cube)
  have o3: interval-lebesgue-integrable lborel (ereal (13 / 30)) (ereal (37 /
    80)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (13 / 30)) (ereal (37 / 80)) outer-integrand
  =
    ((429 / 40) * (37 / 80) + (- 19) * (37 / 80) ^ 2 / 2 + 0 * (37 / 80) ^
    3 / 3) - ((429 /
    40) * (13 / 30) + (- 19) * (13 / 30) ^ 2 / 2 + 0 * (13 / 30) ^ 3 / 3)
  by (rule piece-quad) (simp-all add: outer-piece-3)
  have c3: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (37 /
    80)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (37 / 80)) outer-integrand
  =
    (75527 / 192000)
  using chain[OF - - c2 o3] by (simp add: power2-eq-square power3-eq-cube)
  have o4: interval-lebesgue-integrable lborel (ereal (37 / 80)) (ereal (1 /
    2)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (37 / 80)) (ereal (1 / 2)) outer-integrand
  =
    ((17 / 4) * (1 / 2) + (- 5) * (1 / 2) ^ 2 / 2 + 0 * (1 / 2) ^ 3 / 3) -
    ((17 / 4) * (37 /
    80) + (- 5) * (37 / 80) ^ 2 / 2 + 0 * (37 / 80) ^ 3 / 3)
  by (rule piece-quad) (simp-all add: outer-piece-4)
  have c4: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (1 /
    2)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (1 / 2)) outer-integrand
  =
    (44401 / 96000)

```

```

using chain[OF - - c3 o4] by (simp add: power2-eq-square power3-eq-cube)
have o5: interval-lebesgue-integrable lborel (ereal (1 / 2)) (ereal (43 /
80)) outer-integrand  $\wedge$ 
interval-lebesgue-integral lborel (ereal (1 / 2)) (ereal (43 / 80)) outer-integrand
=
((51 / 10) * (43 / 80) + (- 3) * (43 / 80) ^ 2 / 2 + 0 * (43 / 80) ^ 3
/ 3) - ((51 / 10) *
(1 / 2) + (- 3) * (1 / 2) ^ 2 / 2 + 0 * (1 / 2) ^ 3 / 3)
by (rule piece-quad) (simp-all add: outer-piece-5)
have c5: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (43 /
80)) outer-integrand  $\wedge$ 
interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (43 / 80)) outer-integrand
=
(114317 / 192000)
using chain[OF - - c4 o5] by (simp add: power2-eq-square power3-eq-cube)
have o6: interval-lebesgue-integrable lborel (ereal (43 / 80)) (ereal (37 /
60)) outer-integrand  $\wedge$ 
interval-lebesgue-integral lborel (ereal (43 / 80)) (ereal (37 / 60)) outer-integrand
=
((231 / 20) * (37 / 60) + (- 15) * (37 / 60) ^ 2 / 2 + 0 * (37 / 60) ^
3 / 3) - ((231 /
20) * (43 / 80) + (- 15) * (43 / 80) ^ 2 / 2 + 0 * (43 / 80) ^ 3 / 3)
by (rule piece-quad) (simp-all add: outer-piece-6)
have c6: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (37 /
60)) outer-integrand  $\wedge$ 
interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (37 / 60)) outer-integrand
=
(79151 / 96000)
using chain[OF - - c5 o6] by (simp add: power2-eq-square power3-eq-cube)
have o7: interval-lebesgue-integrable lborel (ereal (37 / 60)) (ereal (13 /
20)) outer-integrand  $\wedge$ 
interval-lebesgue-integral lborel (ereal (37 / 60)) (ereal (13 / 20)) outer-integrand
=
((97 / 10) * (13 / 20) + (- 12) * (13 / 20) ^ 2 / 2 + 0 * (13 / 20) ^ 3
/ 3) - ((97 / 10) *
(37 / 60) + (- 12) * (37 / 60) ^ 2 / 2 + 0 * (37 / 60) ^ 3 / 3)
by (rule piece-quad) (simp-all add: outer-piece-7)
have c7: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (13 /
20)) outer-integrand  $\wedge$ 
interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (13 / 20)) outer-integrand
=
(85871 / 96000)
using chain[OF - - c6 o7] by (simp add: power2-eq-square power3-eq-cube)
have o8: interval-lebesgue-integrable lborel (ereal (13 / 20)) (ereal (37 /
40)) outer-integrand  $\wedge$ 
interval-lebesgue-integral lborel (ereal (13 / 20)) (ereal (37 / 40)) outer-integrand
=
((42 / 5) * (37 / 40) + (- 10) * (37 / 40) ^ 2 / 2 + 0 * (37 / 40) ^ 3
/ 3) - ((42 / 5) *

```

```

      (13 / 20) + (- 10) * (13 / 20) ^ 2 / 2 + 0 * (13 / 20) ^ 3 / 3)
    by (rule piece-quad) (simp-all add: outer-piece-8)
  have c8: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (37 /
    40)) outer-integrand ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (37 / 40)) outer-integrand
  =
    (99731 / 96000)
  using chain[OF - - c7 o8] by (simp add: power2-eq-square power3-eq-cube)
  have o9: interval-lebesgue-integrable lborel (ereal (37 / 40)) (ereal 1) outer-integrand
  ∧
    interval-lebesgue-integral lborel (ereal (37 / 40)) (ereal 1) outer-integrand =
      (1 * 1 + (- 2) * 1 ^ 2 / 2 + 0 * 1 ^ 3 / 3) - (1 * (37 / 40) + (- 2) *
    (37 / 40) ^ 2 / 2 +
      0 * (37 / 40) ^ 3 / 3)
  by (rule piece-quad) (simp-all add: outer-piece-9)
  have c9: interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal 1) outer-integrand
  ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal 1) outer-integrand =
    (93071 / 96000)
  using chain[OF - - c8 o9] by (simp add: power2-eq-square power3-eq-cube)
  have o10: interval-lebesgue-integrable lborel (ereal 1) (ereal (37 / 20)) outer-integrand
  ∧
    interval-lebesgue-integral lborel (ereal 1) (ereal (37 / 20)) outer-integrand =
      ((37 / 20) * (37 / 20) + (- 1) * (37 / 20) ^ 2 / 2 + 0 * (37 / 20) ^ 3
    / 3) - ((37 / 20) *
      1 + (- 1) * 1 ^ 2 / 2 + 0 * 1 ^ 3 / 3)
  by (rule piece-quad) (simp-all add: outer-piece-10)
  have interval-lebesgue-integrable lborel (ereal (1 / 3)) (ereal (37 / 20)) outer-integrand
  ∧
    interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (37 / 20)) outer-integrand
  =
    (127751 / 96000)
  using chain[OF - - c9 o10] by (simp add: power2-eq-square power3-eq-cube)
  thus ?thesis by (simp add: lam-r-def lout-def of-rat-divide)
qed

```

26.4 (5.18): $\int_{-3}^{20} R(x)/x^3 dx$ by 143 affine pieces

On each piece $R(x) = A x + B$ (Lean *gen/gen-inner.py*; breakpoints are the multiples of $1/2, 20/3, 20/37, 10/23, 10/3, 20/43$ in $[3, 20]$). The first fact of each piece gives the floors ($\lfloor x \rfloor, \lfloor \alpha x \rfloor, \lfloor 2H-c x \rfloor, \lfloor 2x \rfloor, \lfloor 2\lambda x \rfloor$) to feed *R-on-piece*; after that the identity follows from *Qc-def, d0-def, es-def, pos-r-def* with min/max/if resolved by the linear side conditions.

lemma *inner-piece-0*: $x \in \{3 <..<(70 / 23)\} \implies R\text{-}r\ x = (18 / 5) * x + (- 6)$
proof –
 assume $x: x \in \{3 <..<(70 / 23)\}$
 have $b: 3 < x < (70 / 23)$ using x by *auto*
 have $real\ 3 \leq x \wedge x < real\ 3 + 1 \wedge real\ 0 \leq \alpha\text{-}r * x \wedge$

$\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 6 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 6 + 1 \wedge \text{real } 6 \leq 2 * x \wedge$
 $2 * x < \text{real } 6 + 1 \wedge$
 $\text{real } 5 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 5 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$ **by** linarith
hence R : $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 3) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 3)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 6)\ (2 * x - \text{real } 6)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 5)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** R $Qc\text{-}def$ $d0\text{-}def$ $es\text{-}def$ $pos\text{-}r\text{-}def$ $min\text{-}def$ $max\text{-}def$ $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma inner-piece-1 : $x \in \{(70 / 23) <..< (120 / 37)\} \implies$

$R\text{-}r\ x = (49 / 20) * x + (-5 / 2)$

proof –

assume x : $x \in \{(70 / 23) <..< (120 / 37)\}$
have b : $(70 / 23) < x < (120 / 37)$ **using** x **by** auto
have $\text{real } 3 \leq x \wedge x < \text{real } 3 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 7 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 7 + 1 \wedge \text{real } 6 \leq 2 * x \wedge$
 $2 * x < \text{real } 6 + 1 \wedge$
 $\text{real } 5 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 5 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$ **by** linarith
hence R : $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 3) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 3)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 7)\ (2 * x - \text{real } 6)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 5)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** R $Qc\text{-}def$ $d0\text{-}def$ $es\text{-}def$ $pos\text{-}r\text{-}def$ $min\text{-}def$ $max\text{-}def$ $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma inner-piece-2 : $x \in \{(120 / 37) <..< (140 / 43)\} \implies$

$R\text{-}r\ x = (283 / 40) * x + (-35 / 2)$

proof –

assume x : $x \in \{(120 / 37) <..< (140 / 43)\}$
have b : $(120 / 37) < x < (140 / 43)$ **using** x **by** auto
have $\text{real } 3 \leq x \wedge x < \text{real } 3 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 7 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 7 + 1 \wedge \text{real } 6 \leq 2 * x \wedge$
 $2 * x < \text{real } 6 + 1 \wedge$

$real\ 6 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 6 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 3) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 3)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 7)\ (2 * x - real\ 6)\ (2 * (lam-r * x) - real\ 6)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-3*: $x \in \{(140 / 43) <..< (10 / 3)\} \implies$
 $R-r\ x = (283 / 40) * x + (- 35 / 2)$
proof –
assume *x*: $x \in \{(140 / 43) <..< (10 / 3)\}$
have *b*: $(140 / 43) < x < (10 / 3)$ **using** *x* **by** *auto*
have $real\ 3 \leq x \wedge x < real\ 3 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 7 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 7 + 1 \wedge real\ 6 \leq 2 * x \wedge$
 $2 * x < real\ 6 + 1 \wedge$
 $real\ 6 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 6 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 3) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 3)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 7)\ (2 * x - real\ 6)\ (2 * (lam-r * x) - real\ 6)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-4*: $x \in \{(10 / 3) <..< (80 / 23)\} \implies$
 $R-r\ x = (277 / 40) * x + (- 17)$
proof –
assume *x*: $x \in \{(10 / 3) <..< (80 / 23)\}$
have *b*: $(10 / 3) < x < (80 / 23)$ **using** *x* **by** *auto*
have $real\ 3 \leq x \wedge x < real\ 3 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 7 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 7 + 1 \wedge real\ 6 \leq 2 * x \wedge$
 $2 * x < real\ 6 + 1 \wedge$
 $real\ 6 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 6 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 3) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 3)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 7)\ (2 * x - real\ 6)\ (2 * (lam-r * x) - real\ 6)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

$(\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc (x - \text{real } 3) (\alpha\text{-}r * x - \text{real } 0) (2 * Hc\text{-}r * x - \text{real } 7) (2 * x - \text{real } 6) (2 * (lam\text{-}r * x) - \text{real } 6)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-5*: $x \in \{(80 / 23) <..< (7 / 2)\} \implies$
 $R\text{-}r\ x = (231 / 40) * x + (-13)$
proof –
assume $x: x \in \{(80 / 23) <..< (7 / 2)\}$
have $b: (80 / 23) < x < (7 / 2)$ **using** x **by** *auto*
have $\text{real } 3 \leq x \wedge x < \text{real } 3 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 8 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 8 + 1 \wedge \text{real } 6 \leq 2 * x \wedge 2 * x < \text{real } 6 + 1 \wedge \text{real } 6 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 6 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 3) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc (x - \text{real } 3) (\alpha\text{-}r * x - \text{real } 0) (2 * Hc\text{-}r * x - \text{real } 8) (2 * x - \text{real } 6) (2 * (lam\text{-}r * x) - \text{real } 6)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-6*: $x \in \{(7 / 2) <..< (160 / 43)\} \implies$
 $R\text{-}r\ x = (271 / 40) * x + (-33 / 2)$
proof –
assume $x: x \in \{(7 / 2) <..< (160 / 43)\}$
have $b: (7 / 2) < x < (160 / 43)$ **using** x **by** *auto*
have $\text{real } 3 \leq x \wedge x < \text{real } 3 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 8 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 8 + 1 \wedge \text{real } 7 \leq 2 * x \wedge 2 * x < \text{real } 7 + 1 \wedge \text{real } 6 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 6 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 3) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc (x - \text{real } 3) (\alpha\text{-}r * x - \text{real } 0) (2 * Hc\text{-}r * x - \text{real } 8) (2 * x - \text{real } 7) (2 * (lam\text{-}r * x) - \text{real } 6)$

$(\text{lam-r} * x) - \text{real } 6)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-7: $x \in \{(160 / 43) <..< (140 / 37)\} \implies$*
 *$R\text{-r } x = (71 / 20) * x + (-9 / 2)$*
proof –
assume *x: $x \in \{(160 / 43) <..< (140 / 37)\}$*
have *b: $(160 / 43) < x < (140 / 37)$* **using** *x* **by** *auto*
have *real 3 ≤ x ∧ x < real 3 + 1 ∧ real 0 ≤ alpha-r * x ∧*
*alpha-r * x < real 0 + 1 ∧*
*real 8 ≤ 2 * Hc-r * x ∧ 2 * Hc-r * x < real 8 + 1 ∧ real 7 ≤ 2 * x ∧*
*2 * x < real 7 + 1 ∧*
*real 6 ≤ 2 * (lam-r * x) ∧ 2 * (lam-r * x) < real 6 + 1*
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 3) - 12 * \text{lam-r} *$*
*(alpha-r * x - real 0))*
*+ Qc (x - real 3) (alpha-r * x - real 0) (2 * Hc-r * x - real 8) (2 * x -*
*real 7) (2 **
*(lam-r * x) - real 6)*
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-8: $x \in \{(140 / 37) <..< (90 / 23)\} \implies$*
 *$R\text{-r } x = (27 / 5) * x + (-23 / 2)$*
proof –
assume *x: $x \in \{(140 / 37) <..< (90 / 23)\}$*
have *b: $(140 / 37) < x < (90 / 23)$* **using** *x* **by** *auto*
have *real 3 ≤ x ∧ x < real 3 + 1 ∧ real 0 ≤ alpha-r * x ∧*
*alpha-r * x < real 0 + 1 ∧*
*real 8 ≤ 2 * Hc-r * x ∧ 2 * Hc-r * x < real 8 + 1 ∧ real 7 ≤ 2 * x ∧*
*2 * x < real 7 + 1 ∧*
*real 7 ≤ 2 * (lam-r * x) ∧ 2 * (lam-r * x) < real 7 + 1*
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 3) - 12 * \text{lam-r} *$*
*(alpha-r * x - real 0))*
*+ Qc (x - real 3) (alpha-r * x - real 0) (2 * Hc-r * x - real 8) (2 * x -*
*real 7) (2 **
*(lam-r * x) - real 7)*
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*

pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-9*: $x \in \{(90 / 23) < .. < 4\} \implies R\text{-}r\ x = (17 / 4) * x + (-\ 7)$
proof –
assume *x*: $x \in \{(90 / 23) < .. < 4\}$
have *b*: $(90 / 23) < x < 4$ **using** *x* **by** *auto*
have *real 3* $\leq x \wedge x < \text{real } 3 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 9 \leq 2 * H\text{-}c\text{-}r * x \wedge 2 * H\text{-}c\text{-}r * x < \text{real } 9 + 1 \wedge \text{real } 7 \leq 2 * x \wedge$
 $2 * x < \text{real } 7 + 1 \wedge$
 $\text{real } 7 \leq 2 * (\text{lam-}r * x) \wedge 2 * (\text{lam-}r * x) < \text{real } 7 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam-}r + 2 * \text{lam-}r * (x - \text{real } 3) - 12 * \text{lam-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Q\text{-}c\ (x - \text{real } 3) (\alpha\text{-}r * x - \text{real } 0) (2 * H\text{-}c\text{-}r * x - \text{real } 9) (2 * x - \text{real } 7) (2 * (\text{lam-}r * x) - \text{real } 7)$
using *R-on-piece* **by** *blast*
show *?thesis* **unfolding** *R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-10*: $x \in \{4 < .. < (180 / 43)\} \implies R\text{-}r\ x = (17 / 5) * x + (-\ 11)$
proof –
assume *x*: $x \in \{4 < .. < (180 / 43)\}$
have *b*: $4 < x < (180 / 43)$ **using** *x* **by** *auto*
have *real 4* $\leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 9 \leq 2 * H\text{-}c\text{-}r * x \wedge 2 * H\text{-}c\text{-}r * x < \text{real } 9 + 1 \wedge \text{real } 8 \leq 2 * x \wedge$
 $2 * x < \text{real } 8 + 1 \wedge$
 $\text{real } 7 \leq 2 * (\text{lam-}r * x) \wedge 2 * (\text{lam-}r * x) < \text{real } 7 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam-}r + 2 * \text{lam-}r * (x - \text{real } 4) - 12 * \text{lam-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Q\text{-}c\ (x - \text{real } 4) (\alpha\text{-}r * x - \text{real } 0) (2 * H\text{-}c\text{-}r * x - \text{real } 9) (2 * x - \text{real } 8) (2 * (\text{lam-}r * x) - \text{real } 7)$
using *R-on-piece* **by** *blast*
show *?thesis* **unfolding** *R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-11*: $x \in \{(180 / 43) < .. < (160 / 37)\} \implies$
 $R\text{-}r\ x = (17 / 5) * x + (- 11)$
proof –
assume $x: x \in \{(180 / 43) < .. < (160 / 37)\}$
have $b: (180 / 43) < x < (160 / 37)$ **using** x **by** *auto*
have $\text{real } 4 \leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 9 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 9 + 1 \wedge \text{real } 8 \leq 2 * x \wedge$
 $2 * x < \text{real } 8 + 1 \wedge$
 $\text{real } 7 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 7 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 4) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 4)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 9)\ (2 * x - \text{real } 8)\ (2 * (\text{lam-r} * x) - \text{real } 7)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-12*: $x \in \{(160 / 37) < .. < (100 / 23)\} \implies$
 $R\text{-}r\ x = (321 / 40) * x + (- 31)$
proof –
assume $x: x \in \{(160 / 37) < .. < (100 / 23)\}$
have $b: (160 / 37) < x < (100 / 23)$ **using** x **by** *auto*
have $\text{real } 4 \leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 9 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 9 + 1 \wedge \text{real } 8 \leq 2 * x \wedge$
 $2 * x < \text{real } 8 + 1 \wedge$
 $\text{real } 8 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 8 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 4) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 4)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 9)\ (2 * x - \text{real } 8)\ (2 * (\text{lam-r} * x) - \text{real } 8)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-13*: $x \in \{(100 / 23) < .. < (9 / 2)\} \implies$
 $R\text{-}r\ x = (55 / 8) * x + (- 26)$

proof –

assume $x: x \in \{(100 / 23) <..<(9 / 2)\}$
have $b: (100 / 23) < x < (9 / 2)$ **using** x **by** *auto*
have $\text{real } 4 \leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 10 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 10 + 1 \wedge \text{real } 8 \leq 2 * x \wedge$
 $2 * x < \text{real } 8 + 1 \wedge$
 $\text{real } 8 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 8 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 4) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 4) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 10) (2 * x - \text{real } 8) (2 * (\text{lam-r} * x) - \text{real } 8)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-14: $x \in \{(9 / 2) <..<(200 / 43)\} \implies$

$R\text{-r } x = (63 / 8) * x + (- 61 / 2)$

proof –

assume $x: x \in \{(9 / 2) <..<(200 / 43)\}$
have $b: (9 / 2) < x < (200 / 43)$ **using** x **by** *auto*
have $\text{real } 4 \leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 10 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 10 + 1 \wedge \text{real } 9 \leq 2 * x \wedge$
 $2 * x < \text{real } 9 + 1 \wedge$
 $\text{real } 8 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 8 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 4) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 4) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 10) (2 * x - \text{real } 9) (2 * (\text{lam-r} * x) - \text{real } 8)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-15: $x \in \{(200 / 43) <..<(110 / 23)\} \implies$

$R\text{-r } x = (93 / 20) * x + (- 31 / 2)$

proof –

assume $x: x \in \{(200 / 43) <..<(110 / 23)\}$
have $b: (200 / 43) < x < (110 / 23)$ **using** x **by** *auto*

have $\text{real } 4 \leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 10 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 10 + 1 \wedge \text{real } 9 \leq 2 * x \wedge$
 $2 * x < \text{real } 9 + 1 \wedge$
 $\text{real } 8 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 8 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** linarith
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 4) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 4) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 10) (2 * x - \text{real } 9) (2 * (\text{lam-r} * x) - \text{real } 8)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** $R Qc\text{-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def}$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma $\text{inner-piece-16: } x \in \{(110 / 23) < .. < (180 / 37)\} \implies$
 $R\text{-r } x = (7 / 2) * x + (-10)$

proof –

assume $x: x \in \{(110 / 23) < .. < (180 / 37)\}$
have $b: (110 / 23) < x < (180 / 37)$ **using** x **by** auto
have $\text{real } 4 \leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 11 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 11 + 1 \wedge \text{real } 9 \leq 2 * x \wedge$
 $2 * x < \text{real } 9 + 1 \wedge$
 $\text{real } 8 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 8 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** linarith
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 4) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 4) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 11) (2 * x - \text{real } 9) (2 * (\text{lam-r} * x) - \text{real } 8)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** $R Qc\text{-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def}$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma $\text{inner-piece-17: } x \in \{(180 / 37) < .. < 5\} \implies R\text{-r } x = (107 / 20) * x + (-19)$

proof –

assume $x: x \in \{(180 / 37) < .. < 5\}$
have $b: (180 / 37) < x < 5$ **using** x **by** auto
have $\text{real } 4 \leq x \wedge x < \text{real } 4 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 11 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 11 + 1 \wedge \text{real } 9 \leq 2 * x \wedge$

$2 * x < \text{real } 9 + 1 \wedge$
 $\text{real } 9 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 9 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 4) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 4) (\text{alpha-r} * x - \text{real } 0) (2 * Hc\text{-r} * x - \text{real } 11) (2 * x - \text{real } 9) (2 * (\text{lam-r} * x) - \text{real } 9)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-18*: $x \in \{5 < .. < (220 / 43)\} \implies R\text{-r } x = (9 / 2) * x + (-24)$
proof –
assume *x*: $x \in \{5 < .. < (220 / 43)\}$
have *b*: $5 < x < (220 / 43)$ **using** *x* **by** *auto*
have $\text{real } 5 \leq x \wedge x < \text{real } 5 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 0 + 1 \wedge \text{real } 11 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 11 + 1 \wedge \text{real } 10 \leq 2 * x \wedge 2 * x < \text{real } 10 + 1 \wedge \text{real } 9 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 9 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 5) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 5) (\text{alpha-r} * x - \text{real } 0) (2 * Hc\text{-r} * x - \text{real } 11) (2 * x - \text{real } 10) (2 * (\text{lam-r} * x) - \text{real } 9)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-19*: $x \in \{(220 / 43) < .. < (120 / 23)\} \implies R\text{-r } x = (9 / 2) * x + (-24)$

proof –
assume *x*: $x \in \{(220 / 43) < .. < (120 / 23)\}$
have *b*: $(220 / 43) < x < (120 / 23)$ **using** *x* **by** *auto*
have $\text{real } 5 \leq x \wedge x < \text{real } 5 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 0 + 1 \wedge \text{real } 11 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 11 + 1 \wedge \text{real } 10 \leq 2 * x \wedge 2 * x < \text{real } 10 + 1 \wedge \text{real } 9 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 9 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*

hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 5) - 12 * \text{lam-r} * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 5)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 11)\ (2 * x - \text{real } 10)\ (2 * (\text{lam-r} * x) - \text{real } 9)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}20$: $x \in \{(120 / 23) <..< (200 / 37)\} \implies R\text{-}r\ x = (67 / 20) * x + (-18)$

proof –

assume x : $x \in \{(120 / 23) <..< (200 / 37)\}$
have b : $(120 / 23) < x < (200 / 37)$ **using** x **by** $auto$
have $\text{real } 5 \leq x \wedge x < \text{real } 5 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 12 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 12 + 1 \wedge \text{real } 10 \leq 2 * x \wedge 2 * x < \text{real } 10 + 1 \wedge \text{real } 9 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 9 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$ **by** $linarith$
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 5) - 12 * \text{lam-r} * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 5)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 12)\ (2 * x - \text{real } 10)\ (2 * (\text{lam-r} * x) - \text{real } 9)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}21$: $x \in \{(200 / 37) <..< (11 / 2)\} \implies R\text{-}r\ x = (319 / 40) * x + (-43)$

proof –

assume x : $x \in \{(200 / 37) <..< (11 / 2)\}$
have b : $(200 / 37) < x < (11 / 2)$ **using** x **by** $auto$
have $\text{real } 5 \leq x \wedge x < \text{real } 5 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 12 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 12 + 1 \wedge \text{real } 10 \leq 2 * x \wedge 2 * x < \text{real } 10 + 1 \wedge \text{real } 10 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 10 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$ **by** $linarith$
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 5) - 12 * \text{lam-r} * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 5)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 12)\ (2 * x -$

$real\ 10) (2 * (lam-r * x) - real\ 10)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-22*: $x \in \{(11 / 2) <..<(240 / 43)\} \implies$
 $R-r\ x = (359 / 40) * x + (-\ 97 / 2)$

proof –
assume $x: x \in \{(11 / 2) <..<(240 / 43)\}$
have $b: (11 / 2) < x < (240 / 43)$ **using** *x by auto*
have $real\ 5 \leq x \wedge x < real\ 5 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 12 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 12 + 1 \wedge real\ 11 \leq 2 * x \wedge$
 $2 * x < real\ 11 + 1 \wedge$
 $real\ 10 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 10 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence $R: R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 5) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 5) (alpha-r * x - real\ 0) (2 * Hc-r * x - real\ 12) (2 * x -$
 $real\ 11) (2 * (lam-r * x) - real\ 10)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-23*: $x \in \{(240 / 43) <..<(130 / 23)\} \implies$
 $R-r\ x = (23 / 4) * x + (-\ 61 / 2)$

proof –
assume $x: x \in \{(240 / 43) <..<(130 / 23)\}$
have $b: (240 / 43) < x < (130 / 23)$ **using** *x by auto*
have $real\ 5 \leq x \wedge x < real\ 5 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 12 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 12 + 1 \wedge real\ 11 \leq 2 * x \wedge$
 $2 * x < real\ 11 + 1 \wedge$
 $real\ 10 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 10 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence $R: R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 5) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 5) (alpha-r * x - real\ 0) (2 * Hc-r * x - real\ 12) (2 * x -$
 $real\ 11) (2 * (lam-r * x) - real\ 10)$
using *R-on-piece* **by** *blast*

show *?thesis unfolding* R $Qc\text{-def}$ $d0\text{-def}$ $es\text{-def}$ $pos\text{-r-def}$ $min\text{-def}$ $max\text{-def}$ $al\text{-pha-r-def}$ $Hc\text{-r-def}$ $lam\text{-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-24*: $x \in \{(130 / 23) <..< (220 / 37)\} \implies$
 $R\text{-r } x = (23 / 5) * x + (-24)$

proof –

assume $x: x \in \{(130 / 23) <..< (220 / 37)\}$
have $b: (130 / 23) < x < (220 / 37)$ **using** x **by** *auto*
have $real\ 5 \leq x \wedge x < real\ 5 + 1 \wedge real\ 0 \leq \alpha\text{-r} * x \wedge$
 $\alpha\text{-r} * x < real\ 0 + 1 \wedge$
 $real\ 13 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < real\ 13 + 1 \wedge real\ 11 \leq 2 * x \wedge$
 $2 * x < real\ 11 + 1 \wedge$
 $real\ 10 \leq 2 * (lam\text{-r} * x) \wedge 2 * (lam\text{-r} * x) < real\ 10 + 1$
using b **unfolding** $\alpha\text{-r-def}$ $Hc\text{-r-def}$ $lam\text{-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * lam\text{-r} + 2 * lam\text{-r} * (x - real\ 5) - 12 * lam\text{-r} * ($
 $\alpha\text{-r} * x - real\ 0))$
 $+ Qc\ (x - real\ 5)\ (\alpha\text{-r} * x - real\ 0)\ (2 * Hc\text{-r} * x - real\ 13)\ (2 * x -$
 $real\ 11)\ (2 * ($
 $lam\text{-r} * x) - real\ 10)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding* R $Qc\text{-def}$ $d0\text{-def}$ $es\text{-def}$ $pos\text{-r-def}$ $min\text{-def}$ $max\text{-def}$ $al\text{-pha-r-def}$ $Hc\text{-r-def}$ $lam\text{-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-25*: $x \in \{(220 / 37) <..< 6\} \implies R\text{-r } x = (129 / 20) * x + (-35)$

proof –

assume $x: x \in \{(220 / 37) <..< 6\}$
have $b: (220 / 37) < x < 6$ **using** x **by** *auto*
have $real\ 5 \leq x \wedge x < real\ 5 + 1 \wedge real\ 0 \leq \alpha\text{-r} * x \wedge$
 $\alpha\text{-r} * x < real\ 0 + 1 \wedge$
 $real\ 13 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < real\ 13 + 1 \wedge real\ 11 \leq 2 * x \wedge$
 $2 * x < real\ 11 + 1 \wedge$
 $real\ 11 \leq 2 * (lam\text{-r} * x) \wedge 2 * (lam\text{-r} * x) < real\ 11 + 1$
using b **unfolding** $\alpha\text{-r-def}$ $Hc\text{-r-def}$ $lam\text{-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * lam\text{-r} + 2 * lam\text{-r} * (x - real\ 5) - 12 * lam\text{-r} * ($
 $\alpha\text{-r} * x - real\ 0))$
 $+ Qc\ (x - real\ 5)\ (\alpha\text{-r} * x - real\ 0)\ (2 * Hc\text{-r} * x - real\ 13)\ (2 * x -$
 $real\ 11)\ (2 * ($
 $lam\text{-r} * x) - real\ 11)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding* R $Qc\text{-def}$ $d0\text{-def}$ $es\text{-def}$ $pos\text{-r-def}$ $min\text{-def}$ $max\text{-def}$ $al\text{-pha-r-def}$ $Hc\text{-r-def}$ $lam\text{-r-def}$

using b by (simp add: field-simps power2-eq-square)
qed

lemma *inner-piece-26*: $x \in \{6 < .. < (260 / 43)\} \implies R\text{-}r\ x = (28 / 5) * x + (-41)$

proof –

assume $x: x \in \{6 < .. < (260 / 43)\}$
have $b: 6 < x \wedge x < (260 / 43)$ using x by auto
have $real\ 6 \leq x \wedge x < real\ 6 + 1 \wedge real\ 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < real\ 0 + 1 \wedge$
 $real\ 13 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 13 + 1 \wedge real\ 12 \leq 2 * x \wedge$
 $2 * x < real\ 12 + 1 \wedge$
 $real\ 11 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < real\ 11 + 1$
using b unfolding $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$ by linarith
hence $R: R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - real\ 6) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - real\ 0))$
 $+ Qc\ (x - real\ 6) (\alpha\text{-}r * x - real\ 0) (2 * Hc\text{-}r * x - real\ 13) (2 * x - real\ 12) (2 * (\text{lam}\text{-}r * x) - real\ 11)$
using $R\text{-on-piece}$ by blast
show ?thesis unfolding $R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$
using b by (simp add: field-simps power2-eq-square)
qed

lemma *inner-piece-27*: $x \in \{(260 / 43) < .. < (140 / 23)\} \implies R\text{-}r\ x = (28 / 5) * x + (-41)$

proof –

assume $x: x \in \{(260 / 43) < .. < (140 / 23)\}$
have $b: (260 / 43) < x \wedge x < (140 / 23)$ using x by auto
have $real\ 6 \leq x \wedge x < real\ 6 + 1 \wedge real\ 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < real\ 0 + 1 \wedge$
 $real\ 13 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 13 + 1 \wedge real\ 12 \leq 2 * x \wedge$
 $2 * x < real\ 12 + 1 \wedge$
 $real\ 11 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < real\ 11 + 1$
using b unfolding $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$ by linarith
hence $R: R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - real\ 6) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - real\ 0))$
 $+ Qc\ (x - real\ 6) (\alpha\text{-}r * x - real\ 0) (2 * Hc\text{-}r * x - real\ 13) (2 * x - real\ 12) (2 * (\text{lam}\text{-}r * x) - real\ 11)$
using $R\text{-on-piece}$ by blast
show ?thesis unfolding $R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$
using b by (simp add: field-simps power2-eq-square)
qed

lemma *inner-piece-28*: $x \in \{(140 / 23) < .. < (240 / 37)\} \implies$
 $R\text{-}r\ x = (89 / 20) * x + (- 34)$
proof –
assume $x: x \in \{(140 / 23) < .. < (240 / 37)\}$
have $b: (140 / 23) < x < (240 / 37)$ **using** x **by** *auto*
have $\text{real } 6 \leq x \wedge x < \text{real } 6 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 14 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 14 + 1 \wedge \text{real } 12 \leq 2 * x \wedge$
 $2 * x < \text{real } 12 + 1 \wedge$
 $\text{real } 11 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 11 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 6) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 6)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 14)\ (2 * x - \text{real } 12)\ (2 * (\text{lam-r} * x) - \text{real } 11)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-29*: $x \in \{(240 / 37) < .. < (13 / 2)\} \implies$
 $R\text{-}r\ x = (363 / 40) * x + (- 64)$
proof –
assume $x: x \in \{(240 / 37) < .. < (13 / 2)\}$
have $b: (240 / 37) < x < (13 / 2)$ **using** x **by** *auto*
have $\text{real } 6 \leq x \wedge x < \text{real } 6 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 14 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 14 + 1 \wedge \text{real } 12 \leq 2 * x \wedge$
 $2 * x < \text{real } 12 + 1 \wedge$
 $\text{real } 12 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 12 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 6) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 6)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 14)\ (2 * x - \text{real } 12)\ (2 * (\text{lam-r} * x) - \text{real } 12)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-30*: $x \in \{(13 / 2) < .. < (280 / 43)\} \implies$
 $R\text{-}r\ x = (403 / 40) * x + (- 141 / 2)$
proof –

assume $x: x \in \{(13 / 2) < .. < (280 / 43)\}$
have $b: (13 / 2) < x < (280 / 43)$ **using** x **by** *auto*
have $\text{real } 6 \leq x \wedge x < \text{real } 6 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 14 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 14 + 1 \wedge \text{real } 13 \leq 2 * x \wedge$
 $2 * x < \text{real } 13 + 1 \wedge$
 $\text{real } 12 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 12 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 6) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 6) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 14) (2 * x - \text{real } 13) (2 * (\text{lam-r} * x) - \text{real } 12)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-31: $x \in \{(280 / 43) < .. < (150 / 23)\} \implies$
 $R\text{-r } x = (137 / 20) * x + (- 99 / 2)$
proof –
assume $x: x \in \{(280 / 43) < .. < (150 / 23)\}$
have $b: (280 / 43) < x < (150 / 23)$ **using** x **by** *auto*
have $\text{real } 6 \leq x \wedge x < \text{real } 6 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 14 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 14 + 1 \wedge \text{real } 13 \leq 2 * x \wedge$
 $2 * x < \text{real } 13 + 1 \wedge$
 $\text{real } 12 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 12 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 6) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 6) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 14) (2 * x - \text{real } 13) (2 * (\text{lam-r} * x) - \text{real } 12)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-32: $x \in \{(150 / 23) < .. < (20 / 3)\} \implies$
 $R\text{-r } x = (57 / 10) * x + (- 42)$
proof –
assume $x: x \in \{(150 / 23) < .. < (20 / 3)\}$
have $b: (150 / 23) < x < (20 / 3)$ **using** x **by** *auto*
have $\text{real } 6 \leq x \wedge x < \text{real } 6 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$

$\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 15 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 15 + 1 \wedge \text{real } 13 \leq 2 * x \wedge$
 $2 * x < \text{real } 13 + 1 \wedge$
 $\text{real } 12 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 12 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$ **by** linarith
hence R : $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 6) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 6)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 15)\ (2 * x - \text{real } 13)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 12)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** R $Qc\text{-}def$ $d0\text{-}def$ $es\text{-}def$ $pos\text{-}r\text{-}def$ $min\text{-}def$ $max\text{-}def$ $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma inner-piece-33 : $x \in \{(20 / 3) <..<(160 / 23)\} \implies$

$R\text{-}r\ x = (69 / 10) * x + (-50)$

proof –

assume x : $x \in \{(20 / 3) <..<(160 / 23)\}$
have b : $(20 / 3) < x < (160 / 23)$ **using** x **by** auto
have $\text{real } 6 \leq x \wedge x < \text{real } 6 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 15 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 15 + 1 \wedge \text{real } 13 \leq 2 * x \wedge$
 $2 * x < \text{real } 13 + 1 \wedge$
 $\text{real } 12 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 12 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$ **by** linarith
hence R : $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 6) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 6)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 15)\ (2 * x - \text{real } 13)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 12)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** R $Qc\text{-}def$ $d0\text{-}def$ $es\text{-}def$ $pos\text{-}r\text{-}def$ $min\text{-}def$ $max\text{-}def$ $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma inner-piece-34 : $x \in \{(160 / 23) <..<(300 / 43)\} \implies$

$R\text{-}r\ x = (23 / 4) * x + (-42)$

proof –

assume x : $x \in \{(160 / 23) <..<(300 / 43)\}$
have b : $(160 / 23) < x < (300 / 43)$ **using** x **by** auto
have $\text{real } 6 \leq x \wedge x < \text{real } 6 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 16 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 16 + 1 \wedge \text{real } 13 \leq 2 * x \wedge$
 $2 * x < \text{real } 13 + 1 \wedge$

$real\ 12 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 12 + 1$
using b **unfolding** $alpha-r-def\ Hc-r-def\ lam-r-def$ **by** $linarith$
hence R : $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 6) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 6)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 16)\ (2 * x - real\ 13)\ (2 * (lam-r * x) - real\ 12)$
using $R-on-piece$ **by** $blast$
show $?thesis$ **unfolding** $R\ Qc-def\ d0-def\ es-def\ pos-r-def\ min-def\ max-def\ alpha-r-def\ Hc-r-def\ lam-r-def$
using b **by** $(simp\ add:\ field-simps\ power2-eq-square)$
qed

lemma $inner-piece-35$: $x \in \{(300 / 43) <..< 7\} \implies R-r\ x = (23 / 4) * x + (-42)$
proof –
assume x : $x \in \{(300 / 43) <..< 7\}$
have b : $(300 / 43) < x < 7$ **using** x **by** $auto$
have $real\ 6 \leq x \wedge x < real\ 6 + 1 \wedge real\ 0 \leq alpha-r * x \wedge alpha-r * x < real\ 0 + 1 \wedge real\ 16 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 16 + 1 \wedge real\ 13 \leq 2 * x \wedge 2 * x < real\ 13 + 1 \wedge real\ 12 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 12 + 1$
using b **unfolding** $alpha-r-def\ Hc-r-def\ lam-r-def$ **by** $linarith$
hence R : $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 6) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 6)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 16)\ (2 * x - real\ 13)\ (2 * (lam-r * x) - real\ 12)$
using $R-on-piece$ **by** $blast$
show $?thesis$ **unfolding** $R\ Qc-def\ d0-def\ es-def\ pos-r-def\ min-def\ max-def\ alpha-r-def\ Hc-r-def\ lam-r-def$
using b **by** $(simp\ add:\ field-simps\ power2-eq-square)$
qed

lemma $inner-piece-36$: $x \in \{7 <..< (260 / 37)\} \implies R-r\ x = (49 / 10) * x + (-49)$
proof –
assume x : $x \in \{7 <..< (260 / 37)\}$
have b : $7 < x < (260 / 37)$ **using** x **by** $auto$
have $real\ 7 \leq x \wedge x < real\ 7 + 1 \wedge real\ 0 \leq alpha-r * x \wedge alpha-r * x < real\ 0 + 1 \wedge real\ 16 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 16 + 1 \wedge real\ 14 \leq 2 * x \wedge 2 * x < real\ 14 + 1 \wedge real\ 12 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 12 + 1$
using b **unfolding** $alpha-r-def\ Hc-r-def\ lam-r-def$ **by** $linarith$
hence R : $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 7) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 7)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 14)\ (2 * x - real\ 13)\ (2 * (lam-r * x) - real\ 12)$
using $R-on-piece$ **by** $blast$
show $?thesis$ **unfolding** $R\ Qc-def\ d0-def\ es-def\ pos-r-def\ min-def\ max-def\ alpha-r-def\ Hc-r-def\ lam-r-def$
using b **by** $(simp\ add:\ field-simps\ power2-eq-square)$
qed

$(\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc (x - \text{real } 7) (\alpha\text{-}r * x - \text{real } 0) (2 * Hc\text{-}r * x - \text{real } 16) (2 * x - \text{real } 14) (2 * (lam\text{-}r * x) - \text{real } 12)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-37*: $x \in \{(260 / 37) <..< (170 / 23)\} \implies R\text{-}r x = (27 / 4) * x + (-62)$
proof –
assume $x: x \in \{(260 / 37) <..< (170 / 23)\}$
have $b: (260 / 37) < x < (170 / 23)$ **using** x **by** *auto*
have $\text{real } 7 \leq x \wedge x < \text{real } 7 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 16 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 16 + 1 \wedge \text{real } 14 \leq 2 * x \wedge 2 * x < \text{real } 14 + 1 \wedge \text{real } 13 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 13 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 7) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 0)) + Qc (x - \text{real } 7) (\alpha\text{-}r * x - \text{real } 0) (2 * Hc\text{-}r * x - \text{real } 16) (2 * x - \text{real } 14) (2 * (lam\text{-}r * x) - \text{real } 13)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-38*: $x \in \{(170 / 23) <..< (320 / 43)\} \implies R\text{-}r x = (28 / 5) * x + (-107 / 2)$
proof –
assume $x: x \in \{(170 / 23) <..< (320 / 43)\}$
have $b: (170 / 23) < x < (320 / 43)$ **using** x **by** *auto*
have $\text{real } 7 \leq x \wedge x < \text{real } 7 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 17 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 17 + 1 \wedge \text{real } 14 \leq 2 * x \wedge 2 * x < \text{real } 14 + 1 \wedge \text{real } 13 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 13 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 7) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 0)) + Qc (x - \text{real } 7) (\alpha\text{-}r * x - \text{real } 0) (2 * Hc\text{-}r * x - \text{real } 17) (2 * x - \text{real } 14) (2 * (lam\text{-}r * x) - \text{real } 13)$

$(\text{lam-r} * x) - \text{real } 13)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-39*: $x \in \{(320 / 43) < .. < (15 / 2)\} \implies$
 $R\text{-}r\ x = (19 / 8) * x + (- 59 / 2)$
proof –
assume $x: x \in \{(320 / 43) < .. < (15 / 2)\}$
have $b: (320 / 43) < x < (15 / 2)$ **using** *x* **by** *auto*
have $\text{real } 7 \leq x \wedge x < \text{real } 7 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 17 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 17 + 1 \wedge \text{real } 14 \leq 2 * x \wedge$
 $2 * x < \text{real } 14 + 1 \wedge$
 $\text{real } 13 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 13 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 7) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 7)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 17)\ (2 * x - \text{real } 14)\ (2 * (\text{lam-r} * x) - \text{real } 13)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-40*: $x \in \{(15 / 2) < .. < (280 / 37)\} \implies$
 $R\text{-}r\ x = (27 / 8) * x + (- 37)$
proof –
assume $x: x \in \{(15 / 2) < .. < (280 / 37)\}$
have $b: (15 / 2) < x < (280 / 37)$ **using** *x* **by** *auto*
have $\text{real } 7 \leq x \wedge x < \text{real } 7 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 17 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 17 + 1 \wedge \text{real } 15 \leq 2 * x \wedge$
 $2 * x < \text{real } 15 + 1 \wedge$
 $\text{real } 13 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 13 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 7) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 7)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 17)\ (2 * x - \text{real } 15)\ (2 * (\text{lam-r} * x) - \text{real } 13)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*

pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-41*: $x \in \{(280 / 37) <..<(180 / 23)\} \implies R\text{-}r\ x = 8 * x + (-72)$

proof –
assume *x*: $x \in \{(280 / 37) <..<(180 / 23)\}$
have *b*: $(280 / 37) < x < (180 / 23)$ **using** *x* **by** *auto*
have *real* $7 \leq x \wedge x < \text{real } 7 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 17 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 17 + 1 \wedge \text{real } 15 \leq 2 * x \wedge$
 $2 * x < \text{real } 15 + 1 \wedge$
 $\text{real } 14 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 14 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 7) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 7)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 17)\ (2 * x - \text{real } 15)\ (2 * (\text{lam-r} * x) - \text{real } 14)$
using *R-on-piece* **by** *blast*
show *?thesis* **unfolding** *R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-42*: $x \in \{(180 / 23) <..<(340 / 43)\} \implies R\text{-}r\ x = (137 / 20) * x + (-63)$

proof –
assume *x*: $x \in \{(180 / 23) <..<(340 / 43)\}$
have *b*: $(180 / 23) < x < (340 / 43)$ **using** *x* **by** *auto*
have *real* $7 \leq x \wedge x < \text{real } 7 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 18 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 18 + 1 \wedge \text{real } 15 \leq 2 * x \wedge$
 $2 * x < \text{real } 15 + 1 \wedge$
 $\text{real } 14 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 14 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 7) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 7)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 18)\ (2 * x - \text{real } 15)\ (2 * (\text{lam-r} * x) - \text{real } 14)$
using *R-on-piece* **by** *blast*
show *?thesis* **unfolding** *R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)

qed

lemma *inner-piece-43*: $x \in \{(340 / 43) < .. < 8\} \implies R\text{-}r\ x = (137 / 20) * x + (-63)$

proof –

assume $x: x \in \{(340 / 43) < .. < 8\}$
have $b: (340 / 43) < x < 8$ **using** x **by** *auto*
have $real\ 7 \leq x \wedge x < real\ 7 + 1 \wedge real\ 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < real\ 0 + 1 \wedge$
 $real\ 18 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 18 + 1 \wedge real\ 15 \leq 2 * x \wedge$
 $2 * x < real\ 15 + 1 \wedge$
 $real\ 14 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < real\ 14 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - real\ 7) - 12 * lam\text{-}r * ($
 $\alpha\text{-}r * x - real\ 0))$
 $+ Qc\ (x - real\ 7)\ (\alpha\text{-}r * x - real\ 0)\ (2 * Hc\text{-}r * x - real\ 18)\ (2 * x -$
 $real\ 15)\ (2 * ($
 $lam\text{-}r * x) - real\ 14)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $pha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def$
 $lam\text{-}r\text{-}def$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-44*: $x \in \{8 < .. < (300 / 37)\} \implies R\text{-}r\ x = 6 * x + (-71)$

proof –

assume $x: x \in \{8 < .. < (300 / 37)\}$
have $b: 8 < x < (300 / 37)$ **using** x **by** *auto*
have $real\ 8 \leq x \wedge x < real\ 8 + 1 \wedge real\ 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < real\ 0 + 1 \wedge$
 $real\ 18 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 18 + 1 \wedge real\ 16 \leq 2 * x \wedge$
 $2 * x < real\ 16 + 1 \wedge$
 $real\ 14 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < real\ 14 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - real\ 8) - 12 * lam\text{-}r * ($
 $\alpha\text{-}r * x - real\ 0))$
 $+ Qc\ (x - real\ 8)\ (\alpha\text{-}r * x - real\ 0)\ (2 * Hc\text{-}r * x - real\ 18)\ (2 * x -$
 $real\ 16)\ (2 * ($
 $lam\text{-}r * x) - real\ 14)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $pha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def$
 $lam\text{-}r\text{-}def$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-45*: $x \in \{(300 / 37) < .. < (190 / 23)\} \implies$
 $R\text{-}r\ x = (157 / 20) * x + (-86)$

proof –

assume $x: x \in \{(300 / 37) <..< (190 / 23)\}$
have $b: (300 / 37) < x < (190 / 23)$ **using** x **by** *auto*
have $\text{real } 8 \leq x \wedge x < \text{real } 8 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 18 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 18 + 1 \wedge \text{real } 16 \leq 2 * x \wedge$
 $2 * x < \text{real } 16 + 1 \wedge$
 $\text{real } 15 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 15 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 8) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 8) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 18) (2 * x - \text{real } 16) (2 * (\text{lam-r} * x) - \text{real } 15)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-46*: $x \in \{(190 / 23) <..< (360 / 43)\} \implies$

$R\text{-r } x = (67 / 10) * x + (- 153 / 2)$

proof –

assume $x: x \in \{(190 / 23) <..< (360 / 43)\}$
have $b: (190 / 23) < x < (360 / 43)$ **using** x **by** *auto*
have $\text{real } 8 \leq x \wedge x < \text{real } 8 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 19 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 19 + 1 \wedge \text{real } 16 \leq 2 * x \wedge$
 $2 * x < \text{real } 16 + 1 \wedge$
 $\text{real } 15 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 15 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 8) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 8) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 19) (2 * x - \text{real } 16) (2 * (\text{lam-r} * x) - \text{real } 15)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-47*: $x \in \{(360 / 43) <..< (17 / 2)\} \implies$

$R\text{-r } x = (139 / 40) * x + (- 99 / 2)$

proof –

assume $x: x \in \{(360 / 43) <..< (17 / 2)\}$
have $b: (360 / 43) < x < (17 / 2)$ **using** x **by** *auto*

have $\text{real } 8 \leq x \wedge x < \text{real } 8 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 19 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 19 + 1 \wedge \text{real } 16 \leq 2 * x \wedge$
 $2 * x < \text{real } 16 + 1 \wedge$
 $\text{real } 15 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 15 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** linarith
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 8) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 8) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 19) (2 * x - \text{real } 16) (2 * (\text{lam-r} * x) - \text{real } 15)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** $R Qc\text{-def } d0\text{-def } es\text{-def } pos\text{-r-def } min\text{-def } max\text{-def } al\text{-pha-r-def } Hc\text{-r-def } lam\text{-r-def}$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma $\text{inner-piece-48}: x \in \{(17 / 2) < .. < (320 / 37)\} \implies$
 $R\text{-r } x = (179 / 40) * x + (-58)$
proof –
assume $x: x \in \{(17 / 2) < .. < (320 / 37)\}$
have $b: (17 / 2) < x < (320 / 37)$ **using** x **by** auto
have $\text{real } 8 \leq x \wedge x < \text{real } 8 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 19 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 19 + 1 \wedge \text{real } 17 \leq 2 * x \wedge$
 $2 * x < \text{real } 17 + 1 \wedge$
 $\text{real } 15 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 15 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** linarith
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 8) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 8) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 19) (2 * x - \text{real } 17) (2 * (\text{lam-r} * x) - \text{real } 15)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** $R Qc\text{-def } d0\text{-def } es\text{-def } pos\text{-r-def } min\text{-def } max\text{-def } al\text{-pha-r-def } Hc\text{-r-def } lam\text{-r-def}$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma $\text{inner-piece-49}: x \in \{(320 / 37) < .. < (200 / 23)\} \implies$
 $R\text{-r } x = (91 / 10) * x + (-98)$
proof –
assume $x: x \in \{(320 / 37) < .. < (200 / 23)\}$
have $b: (320 / 37) < x < (200 / 23)$ **using** x **by** auto
have $\text{real } 8 \leq x \wedge x < \text{real } 8 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 19 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 19 + 1 \wedge \text{real } 17 \leq 2 * x \wedge$

$2 * x < \text{real } 17 + 1 \wedge$
 $\text{real } 16 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 16 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 8) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 8) (\text{alpha-r} * x - \text{real } 0) (2 * Hc\text{-r} * x - \text{real } 19) (2 * x - \text{real } 17) (2 * (\text{lam-r} * x) - \text{real } 16)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-50*: $x \in \{(200 / 23) < .. < (380 / 43)\} \implies$
 $R\text{-r } x = (159 / 20) * x + (-88)$
proof –
assume *x*: $x \in \{(200 / 23) < .. < (380 / 43)\}$
have *b*: $(200 / 23) < x < (380 / 43)$ **using** *x* **by** *auto*
have $\text{real } 8 \leq x \wedge x < \text{real } 8 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 20 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 20 + 1 \wedge \text{real } 17 \leq 2 * x \wedge$
 $2 * x < \text{real } 17 + 1 \wedge$
 $\text{real } 16 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 16 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 8) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 8) (\text{alpha-r} * x - \text{real } 0) (2 * Hc\text{-r} * x - \text{real } 20) (2 * x - \text{real } 17) (2 * (\text{lam-r} * x) - \text{real } 16)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-51*: $x \in \{(380 / 43) < .. < 9\} \implies R\text{-r } x = (159 / 20) * x + (-88)$
proof –
assume *x*: $x \in \{(380 / 43) < .. < 9\}$
have *b*: $(380 / 43) < x < 9$ **using** *x* **by** *auto*
have $\text{real } 8 \leq x \wedge x < \text{real } 8 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 20 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 20 + 1 \wedge \text{real } 17 \leq 2 * x \wedge$
 $2 * x < \text{real } 17 + 1 \wedge$
 $\text{real } 16 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 16 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*

hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 8) - 12 * \text{lam-r} * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 8)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 20)\ (2 * x - \text{real } 17)\ (2 * (\text{lam-r} * x) - \text{real } 16)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}52$: $x \in \{9 < .. < (210 / 23)\} \implies R\text{-}r\ x = (71 / 10) * x + (-97)$

proof –

assume x : $x \in \{9 < .. < (210 / 23)\}$
have b : $9 < x < (210 / 23)$ **using** x **by** $auto$
have $real\ 9 \leq x \wedge x < real\ 9 + 1 \wedge real\ 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < real\ 0 + 1 \wedge real\ 20 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 20 + 1 \wedge real\ 18 \leq 2 * x \wedge 2 * x < real\ 18 + 1 \wedge real\ 16 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < real\ 16 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$ **by** $linarith$
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 9) - 12 * \text{lam-r} * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 9)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 20)\ (2 * x - \text{real } 18)\ (2 * (\text{lam-r} * x) - \text{real } 16)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}53$: $x \in \{(210 / 23) < .. < (340 / 37)\} \implies$

$R\text{-}r\ x = (119 / 20) * x + (-173 / 2)$

proof –

assume x : $x \in \{(210 / 23) < .. < (340 / 37)\}$
have b : $(210 / 23) < x < (340 / 37)$ **using** x **by** $auto$
have $real\ 9 \leq x \wedge x < real\ 9 + 1 \wedge real\ 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < real\ 0 + 1 \wedge real\ 21 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 21 + 1 \wedge real\ 18 \leq 2 * x \wedge 2 * x < real\ 18 + 1 \wedge real\ 16 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < real\ 16 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam-r}\text{-}def$ **by** $linarith$
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 9) - 12 * \text{lam-r} * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 9)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 21)\ (2 * x -$

$real\ 18) (2 * (lam-r * x) - real\ 16)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-54*: $x \in \{(340 / 37) <..< (400 / 43)\} \implies$
 $R-r\ x = (39 / 5) * x + (-207 / 2)$
proof –
assume $x: x \in \{(340 / 37) <..< (400 / 43)\}$
have $b: (340 / 37) < x < (400 / 43)$ **using** *x by auto*
have $real\ 9 \leq x \wedge x < real\ 9 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 21 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 21 + 1 \wedge real\ 18 \leq 2 * x \wedge$
 $2 * x < real\ 18 + 1 \wedge$
 $real\ 17 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 17 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence $R: R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 9) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 9) (alpha-r * x - real\ 0) (2 * Hc-r * x - real\ 21) (2 * x -$
 $real\ 18) (2 * (lam-r * x) - real\ 17)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-55*: $x \in \{(400 / 43) <..< (19 / 2)\} \implies$
 $R-r\ x = (183 / 40) * x + (-147 / 2)$
proof –
assume $x: x \in \{(400 / 43) <..< (19 / 2)\}$
have $b: (400 / 43) < x < (19 / 2)$ **using** *x by auto*
have $real\ 9 \leq x \wedge x < real\ 9 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 21 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 21 + 1 \wedge real\ 18 \leq 2 * x \wedge$
 $2 * x < real\ 18 + 1 \wedge$
 $real\ 17 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 17 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence $R: R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 9) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 9) (alpha-r * x - real\ 0) (2 * Hc-r * x - real\ 21) (2 * x -$
 $real\ 18) (2 * (lam-r * x) - real\ 17)$
using *R-on-piece* **by** *blast*

show *?thesis unfolding* R Qc -def $d0$ -def es -def pos - r -def min -def max -def $alpha$ - r -def Hc - r -def
 lam - r -def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-56*: $x \in \{(19 / 2) < .. < (220 / 23)\} \implies$
 R - r $x = (223 / 40) * x + (- 83)$

proof –

assume $x: x \in \{(19 / 2) < .. < (220 / 23)\}$
have $b: (19 / 2) < x < (220 / 23)$ **using** x **by** *auto*
have $real\ 9 \leq x \wedge x < real\ 9 + 1 \wedge real\ 0 \leq alpha$ - r $* x \wedge$
 $alpha$ - r $* x < real\ 0 + 1 \wedge$
 $real\ 21 \leq 2 * Hc$ - r $* x \wedge 2 * Hc$ - r $* x < real\ 21 + 1 \wedge real\ 19 \leq 2 * x \wedge$
 $2 * x < real\ 19 + 1 \wedge$
 $real\ 17 \leq 2 * (lam$ - r $* x) \wedge 2 * (lam$ - r $* x) < real\ 17 + 1$
using b **unfolding** $alpha$ - r -def Hc - r -def lam - r -def **by** *linarith*
hence $R: R$ - r $x = x * (4 * lam$ - r $+ 2 * lam$ - r $* (x - real\ 9) - 12 * lam$ - r $*$
 $(alpha$ - r $* x - real\ 0))$
 $+ Qc$ $(x - real\ 9) (alpha$ - r $* x - real\ 0) (2 * Hc$ - r $* x - real\ 21) (2 * x -$
 $real\ 19) (2 *$
 $(lam$ - r $* x) - real\ 17)$
using R -on-piece **by** *blast*
show *?thesis unfolding* R Qc -def $d0$ -def es -def pos - r -def min -def max -def $alpha$ - r -def Hc - r -def
 lam - r -def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-57*: $x \in \{(220 / 23) < .. < (360 / 37)\} \implies$
 R - r $x = (177 / 40) * x + (- 72)$

proof –

assume $x: x \in \{(220 / 23) < .. < (360 / 37)\}$
have $b: (220 / 23) < x < (360 / 37)$ **using** x **by** *auto*
have $real\ 9 \leq x \wedge x < real\ 9 + 1 \wedge real\ 0 \leq alpha$ - r $* x \wedge$
 $alpha$ - r $* x < real\ 0 + 1 \wedge$
 $real\ 22 \leq 2 * Hc$ - r $* x \wedge 2 * Hc$ - r $* x < real\ 22 + 1 \wedge real\ 19 \leq 2 * x \wedge$
 $2 * x < real\ 19 + 1 \wedge$
 $real\ 17 \leq 2 * (lam$ - r $* x) \wedge 2 * (lam$ - r $* x) < real\ 17 + 1$
using b **unfolding** $alpha$ - r -def Hc - r -def lam - r -def **by** *linarith*
hence $R: R$ - r $x = x * (4 * lam$ - r $+ 2 * lam$ - r $* (x - real\ 9) - 12 * lam$ - r $*$
 $(alpha$ - r $* x - real\ 0))$
 $+ Qc$ $(x - real\ 9) (alpha$ - r $* x - real\ 0) (2 * Hc$ - r $* x - real\ 22) (2 * x -$
 $real\ 19) (2 *$
 $(lam$ - r $* x) - real\ 17)$
using R -on-piece **by** *blast*
show *?thesis unfolding* R Qc -def $d0$ -def es -def pos - r -def min -def max -def $alpha$ - r -def Hc - r -def
 lam - r -def

using b by (simp add: field-simps power2-eq-square)
qed

lemma inner-piece-58: $x \in \{(360 / 37) < .. < (420 / 43)\} \implies$
 $R\text{-}r\ x = (181 / 20) * x + (-117)$

proof –

assume $x: x \in \{(360 / 37) < .. < (420 / 43)\}$
have $b: (360 / 37) < x < (420 / 43)$ using x by auto
have $\text{real } 9 \leq x \wedge x < \text{real } 9 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 22 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 22 + 1 \wedge \text{real } 19 \leq 2 * x \wedge$
 $2 * x < \text{real } 19 + 1 \wedge$
 $\text{real } 18 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 18 + 1$
using b unfolding alpha-r-def Hc-r-def lam-r-def by linarith
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 9) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 9)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 22)\ (2 * x - \text{real } 19)\ (2 * (\text{lam-r} * x) - \text{real } 18)$
using $R\text{-on-piece}$ by blast
show ?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def
using b by (simp add: field-simps power2-eq-square)
qed

lemma inner-piece-59: $x \in \{(420 / 43) < .. < 10\} \implies$
 $R\text{-}r\ x = (181 / 20) * x + (-117)$

proof –

assume $x: x \in \{(420 / 43) < .. < 10\}$
have $b: (420 / 43) < x < 10$ using x by auto
have $\text{real } 9 \leq x \wedge x < \text{real } 9 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 22 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 22 + 1 \wedge \text{real } 19 \leq 2 * x \wedge$
 $2 * x < \text{real } 19 + 1 \wedge$
 $\text{real } 18 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 18 + 1$
using b unfolding alpha-r-def Hc-r-def lam-r-def by linarith
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 9) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 9)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 22)\ (2 * x - \text{real } 19)\ (2 * (\text{lam-r} * x) - \text{real } 18)$
using $R\text{-on-piece}$ by blast
show ?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def
using b by (simp add: field-simps power2-eq-square)
qed

lemma inner-piece-60: $x \in \{10 < .. < (440 / 43)\} \implies R\text{-}r\ x = (69 / 10) * x + (-114)$

proof –

assume $x: x \in \{10 < .. < (440 / 43)\}$
have $b: 10 < x < (440 / 43)$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 23 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 23 + 1 \wedge \text{real } 20 \leq 2 * x \wedge$
 $2 * x < \text{real } 20 + 1 \wedge$
 $\text{real } 18 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 18 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc\ (x - \text{real } 10)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 23)\ (2 * x$
 $- \text{real } 20)\ (2 * (\text{lam-r} * x) - \text{real } 18)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-61: $x \in \{(440 / 43) < .. < (380 / 37)\} \implies$

$R\text{-}r\ x = (147 / 40) * x + (-81)$

proof –

assume $x: x \in \{(440 / 43) < .. < (380 / 37)\}$
have $b: (440 / 43) < x < (380 / 37)$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 23 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 23 + 1 \wedge \text{real } 20 \leq 2 * x \wedge$
 $2 * x < \text{real } 20 + 1 \wedge$
 $\text{real } 18 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 18 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc\ (x - \text{real } 10)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 23)\ (2 * x$
 $- \text{real } 20)\ (2 * (\text{lam-r} * x) - \text{real } 18)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-62: $x \in \{(380 / 37) < .. < (240 / 23)\} \implies$

$R\text{-}r\ x = (221 / 40) * x + (-100)$
proof –
assume $x: x \in \{(380 / 37) <..< (240 / 23)\}$
have $b: (380 / 37) < x < (240 / 23)$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 23 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 23 + 1 \wedge \text{real } 20 \leq 2 * x \wedge$
 $2 * x < \text{real } 20 + 1 \wedge$
 $\text{real } 19 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 19 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc\ (x - \text{real } 10)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 23)\ (2 * x$
 $- \text{real } 20)\ (2 * (\text{lam-r} * x) - \text{real } 19)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-63: $x \in \{(240 / 23) <..< (21 / 2)\} \implies$
 $R\text{-}r\ x = (35 / 8) * x + (-88)$
proof –
assume $x: x \in \{(240 / 23) <..< (21 / 2)\}$
have $b: (240 / 23) < x < (21 / 2)$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 24 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 24 + 1 \wedge \text{real } 20 \leq 2 * x \wedge$
 $2 * x < \text{real } 20 + 1 \wedge$
 $\text{real } 19 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 19 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc\ (x - \text{real } 10)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * \text{Hc-r} * x - \text{real } 24)\ (2 * x$
 $- \text{real } 20)\ (2 * (\text{lam-r} * x) - \text{real } 19)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-64: $x \in \{(21 / 2) <..< (460 / 43)\} \implies$
 $R\text{-}r\ x = (43 / 8) * x + (-197 / 2)$

proof –

assume $x: x \in \{(21 / 2) < .. < (460 / 43)\}$
have $b: (21 / 2) < x < (460 / 43)$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 24 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 24 + 1 \wedge \text{real } 21 \leq 2 * x \wedge$
 $2 * x < \text{real } 21 + 1 \wedge$
 $\text{real } 19 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 19 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc (x - \text{real } 10) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 24) (2 * x$
 $- \text{real } 21) (2 * (\text{lam-r} * x) - \text{real } 19)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-65*: $x \in \{(460 / 43) < .. < (400 / 37)\} \implies$

$R\text{-r } x = (43 / 8) * x + (-197 / 2)$

proof –

assume $x: x \in \{(460 / 43) < .. < (400 / 37)\}$
have $b: (460 / 43) < x < (400 / 37)$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 24 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 24 + 1 \wedge \text{real } 21 \leq 2 * x \wedge$
 $2 * x < \text{real } 21 + 1 \wedge$
 $\text{real } 19 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 19 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc (x - \text{real } 10) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 24) (2 * x$
 $- \text{real } 21) (2 * (\text{lam-r} * x) - \text{real } 19)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-66*: $x \in \{(400 / 37) < .. < (250 / 23)\} \implies$

$R\text{-r } x = 10 * x + (-297 / 2)$

proof –

assume $x: x \in \{(400 / 37) < .. < (250 / 23)\}$
have $b: (400 / 37) < x < (250 / 23)$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 24 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 24 + 1 \wedge \text{real } 21 \leq 2 * x \wedge$
 $2 * x < \text{real } 21 + 1 \wedge$
 $\text{real } 20 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 20 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc (x - \text{real } 10) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 24) (2 * x$
 $- \text{real } 21) (2 * (\text{lam-r} * x) - \text{real } 20)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-67: $x \in \{(250 / 23) < .. < 11\} \implies$
 $R\text{-r } x = (177 / 20) * x + (-136)$
proof –
assume $x: x \in \{(250 / 23) < .. < 11\}$
have $b: (250 / 23) < x < 11$ **using** x **by** *auto*
have $\text{real } 10 \leq x \wedge x < \text{real } 10 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 25 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 25 + 1 \wedge \text{real } 21 \leq 2 * x \wedge$
 $2 * x < \text{real } 21 + 1 \wedge$
 $\text{real } 20 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 20 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 10) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 0))$
 $+ Qc (x - \text{real } 10) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 25) (2 * x$
 $- \text{real } 21) (2 * (\text{lam-r} * x) - \text{real } 20)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-68: $x \in \{11 < .. < (480 / 43)\} \implies R\text{-r } x = 8 * x + (-147)$
proof –
assume $x: x \in \{11 < .. < (480 / 43)\}$
have $b: 11 < x < (480 / 43)$ **using** x **by** *auto*

have $\text{real } 11 \leq x \wedge x < \text{real } 11 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 25 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 25 + 1 \wedge \text{real } 22 \leq 2 * x \wedge$
 $2 * x < \text{real } 22 + 1 \wedge$
 $\text{real } 20 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 20 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** linarith
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 11) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 11) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 25) (2 * x - \text{real } 22) (2 * (\text{lam-r} * x) - \text{real } 20)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** $R Qc\text{-def } d0\text{-def } es\text{-def } pos\text{-r-def } min\text{-def } max\text{-def } al\text{-pha-r-def } Hc\text{-r-def } lam\text{-r-def}$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma $\text{inner-piece-69}: x \in \{(480 / 43) < .. < (260 / 23)\} \implies$

$R\text{-r } x = (191 / 40) * x + (-111)$

proof –

assume $x: x \in \{(480 / 43) < .. < (260 / 23)\}$
have $b: (480 / 43) < x < (260 / 23)$ **using** x **by** auto
have $\text{real } 11 \leq x \wedge x < \text{real } 11 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 25 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 25 + 1 \wedge \text{real } 22 \leq 2 * x \wedge$
 $2 * x < \text{real } 22 + 1 \wedge$
 $\text{real } 20 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 20 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** linarith
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 11) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 11) (\text{alpha-r} * x - \text{real } 0) (2 * \text{Hc-r} * x - \text{real } 25) (2 * x - \text{real } 22) (2 * (\text{lam-r} * x) - \text{real } 20)$
using $R\text{-on-piece}$ **by** blast
show $?thesis$ **unfolding** $R Qc\text{-def } d0\text{-def } es\text{-def } pos\text{-r-def } min\text{-def } max\text{-def } al\text{-pha-r-def } Hc\text{-r-def } lam\text{-r-def}$
using b **by** $(\text{simp add: field-simps power2-eq-square})$
qed

lemma $\text{inner-piece-70}: x \in \{(260 / 23) < .. < (420 / 37)\} \implies$

$R\text{-r } x = (29 / 8) * x + (-98)$

proof –

assume $x: x \in \{(260 / 23) < .. < (420 / 37)\}$
have $b: (260 / 23) < x < (420 / 37)$ **using** x **by** auto
have $\text{real } 11 \leq x \wedge x < \text{real } 11 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$

$\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 26 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 26 + 1 \wedge \text{real } 22 \leq 2 * x \wedge$
 $2 * x < \text{real } 22 + 1 \wedge$
 $\text{real } 20 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 20 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 11) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 11)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 26)\ (2 * x - \text{real } 22)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 20)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-71*: $x \in \{(420 / 37) < .. < (23 / 2)\} \implies$

$R\text{-}r\ x = (219 / 40) * x + (-119)$

proof –

assume *x*: $x \in \{(420 / 37) < .. < (23 / 2)\}$
have *b*: $(420 / 37) < x < (23 / 2)$ **using** *x* **by** *auto*
have $\text{real } 11 \leq x \wedge x < \text{real } 11 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$
 $\text{real } 26 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 26 + 1 \wedge \text{real } 22 \leq 2 * x \wedge$
 $2 * x < \text{real } 22 + 1 \wedge$
 $\text{real } 21 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 21 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 11) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 11)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 26)\ (2 * x - \text{real } 22)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 21)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-72*: $x \in \{(23 / 2) < .. < (500 / 43)\} \implies$

$R\text{-}r\ x = (259 / 40) * x + (-261 / 2)$

proof –

assume *x*: $x \in \{(23 / 2) < .. < (500 / 43)\}$
have *b*: $(23 / 2) < x < (500 / 43)$ **using** *x* **by** *auto*
have $\text{real } 11 \leq x \wedge x < \text{real } 11 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 0 + 1 \wedge$

$real\ 26 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 26 + 1 \wedge real\ 23 \leq 2 * x \wedge$
 $2 * x < real\ 23 + 1 \wedge$
 $real\ 21 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 21 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 11) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 11)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 26)\ (2 * x - real\ 23)\ (2 * (lam-r * x) - real\ 21)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-73*: $x \in \{(500 / 43) < .. < (270 / 23)\} \implies$
 $R-r\ x = (259 / 40) * x + (- 261 / 2)$
proof –
assume *x*: $x \in \{(500 / 43) < .. < (270 / 23)\}$
have *b*: $(500 / 43) < x < (270 / 23)$ **using** *x* **by** *auto*
have $real\ 11 \leq x \wedge x < real\ 11 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 26 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 26 + 1 \wedge real\ 23 \leq 2 * x \wedge$
 $2 * x < real\ 23 + 1 \wedge$
 $real\ 21 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 21 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 11) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 11)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 26)\ (2 * x - real\ 23)\ (2 * (lam-r * x) - real\ 21)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-74*: $x \in \{(270 / 23) < .. < (440 / 37)\} \implies$
 $R-r\ x = (213 / 40) * x + (- 117)$
proof –
assume *x*: $x \in \{(270 / 23) < .. < (440 / 37)\}$
have *b*: $(270 / 23) < x < (440 / 37)$ **using** *x* **by** *auto*
have $real\ 11 \leq x \wedge x < real\ 11 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 27 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 27 + 1 \wedge real\ 23 \leq 2 * x \wedge$

$2 * x < \text{real } 23 + 1 \wedge$
 $\text{real } 21 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 21 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 11) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 11) (\text{alpha-r} * x - \text{real } 0) (2 * Hc\text{-r} * x - \text{real } 27) (2 * x - \text{real } 23) (2 * (\text{lam-r} * x) - \text{real } 21)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-75*: $x \in \{(440 / 37) < .. < 12\} \implies$
 $R\text{-r } x = (199 / 20) * x + (-172)$
proof –
assume *x*: $x \in \{(440 / 37) < .. < 12\}$
have *b*: $(440 / 37) < x < 12$ **using** *x* **by** *auto*
have $\text{real } 11 \leq x \wedge x < \text{real } 11 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 27 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 27 + 1 \wedge \text{real } 23 \leq 2 * x \wedge$
 $2 * x < \text{real } 23 + 1 \wedge$
 $\text{real } 22 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 22 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 11) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 11) (\text{alpha-r} * x - \text{real } 0) (2 * Hc\text{-r} * x - \text{real } 27) (2 * x - \text{real } 23) (2 * (\text{lam-r} * x) - \text{real } 22)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-76*: $x \in \{12 < .. < (520 / 43)\} \implies R\text{-r } x = (91 / 10) * x + (-184)$
proof –
assume *x*: $x \in \{12 < .. < (520 / 43)\}$
have *b*: $12 < x < (520 / 43)$ **using** *x* **by** *auto*
have $\text{real } 12 \leq x \wedge x < \text{real } 12 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 0 + 1 \wedge$
 $\text{real } 27 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 27 + 1 \wedge \text{real } 24 \leq 2 * x \wedge$
 $2 * x < \text{real } 24 + 1 \wedge$

$real\ 22 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 22 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 12) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 12)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 27)\ (2 * x - real\ 24)\ (2 * (lam-r * x) - real\ 22)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-77*: $x \in \{(520 / 43) <..<(280 / 23)\} \implies R-r\ x = (47 / 8) * x + (-145)$
proof –
assume *x*: $x \in \{(520 / 43) <..<(280 / 23)\}$
have *b*: $(520 / 43) < x < (280 / 23)$ **using** *x* **by** *auto*
have $real\ 12 \leq x \wedge x < real\ 12 + 1 \wedge real\ 0 \leq alpha-r * x \wedge alpha-r * x < real\ 0 + 1 \wedge real\ 27 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 27 + 1 \wedge real\ 24 \leq 2 * x \wedge 2 * x < real\ 24 + 1 \wedge real\ 22 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 22 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 12) - 12 * lam-r * (alpha-r * x - real\ 0))$
 $+ Qc\ (x - real\ 12)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 27)\ (2 * x - real\ 24)\ (2 * (lam-r * x) - real\ 22)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-78*: $x \in \{(280 / 23) <..<(460 / 37)\} \implies R-r\ x = (189 / 40) * x + (-131)$
proof –
assume *x*: $x \in \{(280 / 23) <..<(460 / 37)\}$
have *b*: $(280 / 23) < x < (460 / 37)$ **using** *x* **by** *auto*
have $real\ 12 \leq x \wedge x < real\ 12 + 1 \wedge real\ 0 \leq alpha-r * x \wedge alpha-r * x < real\ 0 + 1 \wedge real\ 28 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 28 + 1 \wedge real\ 24 \leq 2 * x \wedge 2 * x < real\ 24 + 1 \wedge real\ 22 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 22 + 1$

using b **unfolding** α -r-def H c-r-def λ m-r-def **by** *linarith*
hence R : $R\text{-}r\ x = x * (4 * \lambda m\text{-}r + 2 * \lambda m\text{-}r * (x - \text{real } 12) - 12 * \lambda m\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 12)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 28)\ (2 * x - \text{real } 24)\ (2 * (\lambda m\text{-}r * x) - \text{real } 22)$
using R -on-piece **by** *blast*
show ?thesis **unfolding** R Q c-def d 0-def e s-def p os-r-def m in-def m ax-def α -r-def H c-r-def λ m-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-79*: $x \in \{(460 / 37) <..< (25 / 2)\} \implies$

$R\text{-}r\ x = (263 / 40) * x + (-154)$

proof –

assume x : $x \in \{(460 / 37) <..< (25 / 2)\}$
have b : $(460 / 37) < x < (25 / 2)$ **using** x **by** *auto*
have $\text{real } 12 \leq x \wedge x < \text{real } 12 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 28 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 28 + 1 \wedge \text{real } 24 \leq 2 * x \wedge 2 * x < \text{real } 24 + 1 \wedge \text{real } 23 \leq 2 * (\lambda m\text{-}r * x) \wedge 2 * (\lambda m\text{-}r * x) < \text{real } 23 + 1$
using b **unfolding** α -r-def H c-r-def λ m-r-def **by** *linarith*
hence R : $R\text{-}r\ x = x * (4 * \lambda m\text{-}r + 2 * \lambda m\text{-}r * (x - \text{real } 12) - 12 * \lambda m\text{-}r * (\alpha\text{-}r * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 12)\ (\alpha\text{-}r * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 28)\ (2 * x - \text{real } 24)\ (2 * (\lambda m\text{-}r * x) - \text{real } 23)$
using R -on-piece **by** *blast*
show ?thesis **unfolding** R Q c-def d 0-def e s-def p os-r-def m in-def m ax-def α -r-def H c-r-def λ m-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-80*: $x \in \{(25 / 2) <..< (540 / 43)\} \implies$

$R\text{-}r\ x = (303 / 40) * x + (-333 / 2)$

proof –

assume x : $x \in \{(25 / 2) <..< (540 / 43)\}$
have b : $(25 / 2) < x < (540 / 43)$ **using** x **by** *auto*
have $\text{real } 12 \leq x \wedge x < \text{real } 12 + 1 \wedge \text{real } 0 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 0 + 1 \wedge \text{real } 28 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 28 + 1 \wedge \text{real } 25 \leq 2 * x \wedge 2 * x < \text{real } 25 + 1 \wedge \text{real } 23 \leq 2 * (\lambda m\text{-}r * x) \wedge 2 * (\lambda m\text{-}r * x) < \text{real } 23 + 1$
using b **unfolding** α -r-def H c-r-def λ m-r-def **by** *linarith*

hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 12) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 12)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 28)\ (2 * x - \text{real } 25)\ (2 * (\text{lam-r} * x) - \text{real } 23)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ \text{unfolding}\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ al\text{-}pha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}81$: $x \in \{(540 / 43) < .. < (290 / 23)\} \implies$

$R\text{-}r\ x = (303 / 40) * x + (- 333 / 2)$

proof –

assume x : $x \in \{(540 / 43) < .. < (290 / 23)\}$
have b : $(540 / 43) < x < (290 / 23)$ **using** x **by** $auto$
have $\text{real } 12 \leq x \wedge x < \text{real } 12 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 0 + 1 \wedge \text{real } 28 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 28 + 1 \wedge \text{real } 25 \leq 2 * x \wedge 2 * x < \text{real } 25 + 1 \wedge \text{real } 23 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 23 + 1$
using b **unfolding** $\text{alpha-r-def}\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$ **by** linarith
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 12) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$
 $+ Qc\ (x - \text{real } 12)\ (\text{alpha-r} * x - \text{real } 0)\ (2 * Hc\text{-}r * x - \text{real } 28)\ (2 * x - \text{real } 25)\ (2 * (\text{lam-r} * x) - \text{real } 23)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ \text{unfolding}\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ al\text{-}pha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}82$: $x \in \{(290 / 23) < .. < (480 / 37)\} \implies$

$R\text{-}r\ x = (257 / 40) * x + (- 152)$

proof –

assume x : $x \in \{(290 / 23) < .. < (480 / 37)\}$
have b : $(290 / 23) < x < (480 / 37)$ **using** x **by** $auto$
have $\text{real } 12 \leq x \wedge x < \text{real } 12 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 0 + 1 \wedge \text{real } 29 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 29 + 1 \wedge \text{real } 25 \leq 2 * x \wedge 2 * x < \text{real } 25 + 1 \wedge \text{real } 23 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 23 + 1$
using b **unfolding** $\text{alpha-r-def}\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$ **by** linarith
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 12) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 0))$

$(\text{alpha-r} * x - \text{real } 0))$
 $+ Qc (x - \text{real } 12) (\text{alpha-r} * x - \text{real } 0) (2 * Hc-r * x - \text{real } 29) (2 * x - \text{real } 25) (2 * (lam-r * x) - \text{real } 23)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-83*: $x \in \{(480 / 37) < .. < 13\} \implies R-r x = (221 / 20) * x + (-212)$
proof –
assume $x: x \in \{(480 / 37) < .. < 13\}$
have $b: (480 / 37) < x < 13$ **using** x **by** *auto*
have $\text{real } 12 \leq x \wedge x < \text{real } 12 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 0 + 1 \wedge \text{real } 29 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 29 + 1 \wedge \text{real } 25 \leq 2 * x \wedge 2 * x < \text{real } 25 + 1 \wedge \text{real } 24 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 12) - 12 * lam-r * (\text{alpha-r} * x - \text{real } 0)) + Qc (x - \text{real } 12) (\text{alpha-r} * x - \text{real } 0) (2 * Hc-r * x - \text{real } 29) (2 * x - \text{real } 25) (2 * (lam-r * x) - \text{real } 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-84*: $x \in \{13 < .. < (560 / 43)\} \implies R-r x = (51 / 5) * x + (-225)$
proof –
assume $x: x \in \{13 < .. < (560 / 43)\}$
have $b: 13 < x < (560 / 43)$ **using** x **by** *auto*
have $\text{real } 13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 0 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 0 + 1 \wedge \text{real } 29 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 29 + 1 \wedge \text{real } 26 \leq 2 * x \wedge 2 * x < \text{real } 26 + 1 \wedge \text{real } 24 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 13) - 12 * lam-r * (\text{alpha-r} * x - \text{real } 0)) + Qc (x - \text{real } 13) (\text{alpha-r} * x - \text{real } 0) (2 * Hc-r * x - \text{real } 29) (2 * x - \text{real } 26) (2 * (lam-r * x) - \text{real } 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

$real\ 0))$
 $+ Qc\ (x - real\ 13)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 29)\ (2 * x$
 $- real\ 26)\ (2 *$
 $(lam-r * x) - real\ 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-85*: $x \in \{(560 / 43) <..< (300 / 23)\} \implies$

$R-r\ x = (279 / 40) * x + (-183)$

proof –

assume $x: x \in \{(560 / 43) <..< (300 / 23)\}$
have $b: (560 / 43) < x < (300 / 23)$ **using** x **by** *auto*
have $real\ 13 \leq x \wedge x < real\ 13 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 29 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 29 + 1 \wedge real\ 26 \leq 2 * x \wedge$
 $2 * x < real\ 26 + 1 \wedge$
 $real\ 24 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 13) - 12 * lam-r * (alpha-r * x -$
 $real\ 0))$
 $+ Qc\ (x - real\ 13)\ (alpha-r * x - real\ 0)\ (2 * Hc-r * x - real\ 29)\ (2 * x$
 $- real\ 26)\ (2 *$
 $(lam-r * x) - real\ 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-86*: $x \in \{(300 / 23) <..< (40 / 3)\} \implies$

$R-r\ x = (233 / 40) * x + (-168)$

proof –

assume $x: x \in \{(300 / 23) <..< (40 / 3)\}$
have $b: (300 / 23) < x < (40 / 3)$ **using** x **by** *auto*
have $real\ 13 \leq x \wedge x < real\ 13 + 1 \wedge real\ 0 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 0 + 1 \wedge$
 $real\ 30 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 30 + 1 \wedge real\ 26 \leq 2 * x \wedge$
 $2 * x < real\ 26 + 1 \wedge$
 $real\ 24 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 13) - 12 * lam-r * (alpha-r * x -$
 $real\ 0))$

$+ Qc (x - \text{real } 13) (\alpha\text{-}r * x - \text{real } 0) (2 * Hc\text{-}r * x - \text{real } 30) (2 * x - \text{real } 26) (2 * (lam\text{-}r * x) - \text{real } 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-87*: $x \in \{(40 / 3) <..<(310 / 23)\} \implies$

$R\text{-}r x = (145 / 8) * x + (-184)$

proof –

assume $x: x \in \{(40 / 3) <..<(310 / 23)\}$
have $b: (40 / 3) < x < (310 / 23)$ **using** x **by** *auto*
have $\text{real } 13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 1 + 1 \wedge \text{real } 30 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 30 + 1 \wedge \text{real } 26 \leq 2 * x \wedge 2 * x < \text{real } 26 + 1 \wedge \text{real } 24 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 13) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 1))$
 $+ Qc (x - \text{real } 13) (\alpha\text{-}r * x - \text{real } 1) (2 * Hc\text{-}r * x - \text{real } 30) (2 * x - \text{real } 26) (2 * (lam\text{-}r * x) - \text{real } 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** *(simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-88*: $x \in \{(310 / 23) <..<(580 / 43)\} \implies$

$R\text{-}r x = (679 / 40) * x + (-337 / 2)$

proof –

assume $x: x \in \{(310 / 23) <..<(580 / 43)\}$
have $b: (310 / 23) < x < (580 / 43)$ **using** x **by** *auto*
have $\text{real } 13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 1 + 1 \wedge \text{real } 31 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 31 + 1 \wedge \text{real } 26 \leq 2 * x \wedge 2 * x < \text{real } 26 + 1 \wedge \text{real } 24 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 13) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 1))$
 $+ Qc (x - \text{real } 13) (\alpha\text{-}r * x - \text{real } 1) (2 * Hc\text{-}r * x - \text{real } 31) (2 * x - \text{real } 26) (2 * (lam\text{-}r * x) - \text{real } 24)$

$- \text{real } 26) (2 * (lam-r * x) - \text{real } 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-89*: $x \in \{(580 / 43) < .. < (27 / 2)\} \implies$
 $R-r \ x = (679 / 40) * x + (- 337 / 2)$

proof $-$
assume $x: x \in \{(580 / 43) < .. < (27 / 2)\}$
have $b: (580 / 43) < x < (27 / 2)$ **using** x **by** *auto*
have $\text{real } 13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 1 \leq \alpha-r * x \wedge$
 $\alpha-r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 31 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 31 + 1 \wedge \text{real } 26 \leq 2 * x \wedge$
 $2 * x < \text{real } 26 + 1 \wedge$
 $\text{real } 24 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r \ x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 13) - 12 * lam-r * (alpha-r * x -$
 $\text{real } 1))$
 $+ Qc \ (x - \text{real } 13) \ (\alpha-r * x - \text{real } 1) \ (2 * Hc-r * x - \text{real } 31) \ (2 * x$
 $- \text{real } 26) (2 * (lam-r * x) - \text{real } 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-90*: $x \in \{(27 / 2) < .. < (500 / 37)\} \implies$
 $R-r \ x = (719 / 40) * x + (- 182)$

proof $-$
assume $x: x \in \{(27 / 2) < .. < (500 / 37)\}$
have $b: (27 / 2) < x < (500 / 37)$ **using** x **by** *auto*
have $\text{real } 13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 1 \leq \alpha-r * x \wedge$
 $\alpha-r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 31 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 31 + 1 \wedge \text{real } 27 \leq 2 * x \wedge$
 $2 * x < \text{real } 27 + 1 \wedge$
 $\text{real } 24 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 24 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r \ x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 13) - 12 * lam-r * (alpha-r * x -$
 $\text{real } 1))$
 $+ Qc \ (x - \text{real } 13) \ (\alpha-r * x - \text{real } 1) \ (2 * Hc-r * x - \text{real } 31) \ (2 * x$
 $- \text{real } 27) (2 * (lam-r * x) - \text{real } 24)$

$(\text{lam-r} * x) - \text{real } 24)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-91*: $x \in \{(500 / 37) < .. < (320 / 23)\} \implies$
 $R\text{-r } x = (793 / 40) * x + (-207)$
proof –
assume $x: x \in \{(500 / 37) < .. < (320 / 23)\}$
have $b: (500 / 37) < x < (320 / 23)$ **using** *x* **by** *auto*
have $\text{real } 13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 1 \leq \alpha\text{-r} * x \wedge$
 $\alpha\text{-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 31 \leq 2 * H\text{c-r} * x \wedge 2 * H\text{c-r} * x < \text{real } 31 + 1 \wedge \text{real } 27 \leq 2 * x \wedge$
 $2 * x < \text{real } 27 + 1 \wedge$
 $\text{real } 25 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 25 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 13) - 12 * \text{lam-r} *$
 $(\alpha\text{-r} * x -$
 $\text{real } 1))$
 $+ Q\text{c } (x - \text{real } 13) (\alpha\text{-r} * x - \text{real } 1) (2 * H\text{c-r} * x - \text{real } 31) (2 * x$
 $- \text{real } 27) (2 *$
 $(\text{lam-r} * x) - \text{real } 25)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-92*: $x \in \{(320 / 23) < .. < (600 / 43)\} \implies$
 $R\text{-r } x = (747 / 40) * x + (-191)$
proof –
assume $x: x \in \{(320 / 23) < .. < (600 / 43)\}$
have $b: (320 / 23) < x < (600 / 43)$ **using** *x* **by** *auto*
have $\text{real } 13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 1 \leq \alpha\text{-r} * x \wedge$
 $\alpha\text{-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 32 \leq 2 * H\text{c-r} * x \wedge 2 * H\text{c-r} * x < \text{real } 32 + 1 \wedge \text{real } 27 \leq 2 * x \wedge$
 $2 * x < \text{real } 27 + 1 \wedge$
 $\text{real } 25 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 25 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 13) - 12 * \text{lam-r} *$
 $(\alpha\text{-r} * x -$
 $\text{real } 1))$
 $+ Q\text{c } (x - \text{real } 13) (\alpha\text{-r} * x - \text{real } 1) (2 * H\text{c-r} * x - \text{real } 32) (2 * x$
 $- \text{real } 27) (2 *$
 $(\text{lam-r} * x) - \text{real } 25)$

using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-93*: $x \in \{(600 / 43) < .. < 14\} \implies$
 $R\text{-}r\ x = (309 / 20) * x + (-146)$
proof –
assume *x*: $x \in \{(600 / 43) < .. < 14\}$
have *b*: $(600 / 43) < x < 14$ **using** *x* **by** *auto*
have *real* $13 \leq x \wedge x < \text{real } 13 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 32 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 32 + 1 \wedge \text{real } 27 \leq 2 * x \wedge$
 $2 * x < \text{real } 27 + 1 \wedge$
 $\text{real } 25 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 25 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 13) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ \text{Qc} (x - \text{real } 13) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 32) (2 * x$
 $- \text{real } 27) (2 * (\text{lam-r} * x) - \text{real } 25)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-94*: $x \in \{14 < .. < (520 / 37)\} \implies R\text{-}r\ x = (73 / 5) * x + (-160)$
proof –
assume *x*: $x \in \{14 < .. < (520 / 37)\}$
have *b*: $14 < x < (520 / 37)$ **using** *x* **by** *auto*
have *real* $14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 32 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 32 + 1 \wedge \text{real } 28 \leq 2 * x \wedge$
 $2 * x < \text{real } 28 + 1 \wedge$
 $\text{real } 25 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 25 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ \text{Qc} (x - \text{real } 14) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 32) (2 * x$
 $- \text{real } 28) (2 * (\text{lam-r} * x) - \text{real } 25)$
using *R-on-piece* **by** *blast*

show *?thesis unfolding* *R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-95*: $x \in \{(520 / 37) <..< (330 / 23)\} \implies$
 $R\text{-}r\ x = (769 / 40) * x + (- 225)$

proof –
assume $x: x \in \{(520 / 37) <..< (330 / 23)\}$
have $b: (520 / 37) < x < (330 / 23)$ **using** *x by auto*
have $\text{real } 14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 32 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 32 + 1 \wedge \text{real } 28 \leq 2 * x \wedge$
 $2 * x < \text{real } 28 + 1 \wedge$
 $\text{real } 26 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 26 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 14)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 32)\ (2 * x$
 $- \text{real } 28)\ (2 * (\text{lam-r} * x) - \text{real } 26)$
using *R-on-piece by blast*
show *?thesis unfolding* *R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-96*: $x \in \{(330 / 23) <..< (620 / 43)\} \implies$
 $R\text{-}r\ x = (723 / 40) * x + (- 417 / 2)$

proof –
assume $x: x \in \{(330 / 23) <..< (620 / 43)\}$
have $b: (330 / 23) < x < (620 / 43)$ **using** *x by auto*
have $\text{real } 14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 33 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 33 + 1 \wedge \text{real } 28 \leq 2 * x \wedge$
 $2 * x < \text{real } 28 + 1 \wedge$
 $\text{real } 26 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 26 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 14)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 33)\ (2 * x$
 $- \text{real } 28)\ (2 * (\text{lam-r} * x) - \text{real } 26)$
using *R-on-piece by blast*
show *?thesis unfolding* *R Qc-def d0-def es-def pos-r-def min-def max-def al-*

pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-97*: $x \in \{(620 / 43) <..< (29 / 2)\} \implies$
 $R\text{-}r\ x = (723 / 40) * x + (- 417 / 2)$

proof –
assume $x: x \in \{(620 / 43) <..< (29 / 2)\}$
have $b: (620 / 43) < x < (29 / 2)$ **using** x **by** *auto*
have $\text{real } 14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 33 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 33 + 1 \wedge \text{real } 28 \leq 2 * x \wedge$
 $2 * x < \text{real } 28 + 1 \wedge$
 $\text{real } 26 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 26 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 14)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 33)\ (2 * x$
 $- \text{real } 28)\ (2 * (\text{lam-r} * x) - \text{real } 26)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-98*: $x \in \{(29 / 2) <..< (540 / 37)\} \implies$
 $R\text{-}r\ x = (763 / 40) * x + (- 223)$

proof –
assume $x: x \in \{(29 / 2) <..< (540 / 37)\}$
have $b: (29 / 2) < x < (540 / 37)$ **using** x **by** *auto*
have $\text{real } 14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 33 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 33 + 1 \wedge \text{real } 29 \leq 2 * x \wedge$
 $2 * x < \text{real } 29 + 1 \wedge$
 $\text{real } 26 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 26 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 14)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 33)\ (2 * x$
 $- \text{real } 29)\ (2 * (\text{lam-r} * x) - \text{real } 26)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def

lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-99: $x \in \{(540 / 37) <..< (340 / 23)\} \implies$
 $R\text{-}r\ x = (837 / 40) * x + (- 250)$
proof –
assume $x: x \in \{(540 / 37) <..< (340 / 23)\}$
have $b: (540 / 37) < x < (340 / 23)$ **using** x **by** *auto*
have $\text{real } 14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 33 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 33 + 1 \wedge \text{real } 29 \leq 2 * x \wedge$
 $2 * x < \text{real } 29 + 1 \wedge$
 $\text{real } 27 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 27 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 14)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 33)\ (2 * x - \text{real } 29)\ (2 * (\text{lam-r} * x) - \text{real } 27)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-100: $x \in \{(340 / 23) <..< (640 / 43)\} \implies$
 $R\text{-}r\ x = (791 / 40) * x + (- 233)$
proof –
assume $x: x \in \{(340 / 23) <..< (640 / 43)\}$
have $b: (340 / 23) < x < (640 / 43)$ **using** x **by** *auto*
have $\text{real } 14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 34 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 34 + 1 \wedge \text{real } 29 \leq 2 * x \wedge$
 $2 * x < \text{real } 29 + 1 \wedge$
 $\text{real } 27 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 27 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 14)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 34)\ (2 * x - \text{real } 29)\ (2 * (\text{lam-r} * x) - \text{real } 27)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
 lam-r-def

using b by (simp add: field-simps power2-eq-square)
 qed

lemma inner-piece-101: $x \in \{(640 / 43) < .. < 15\} \implies$
 $R\text{-}r\ x = (331 / 20) * x + (-185)$
proof –
 assume $x: x \in \{(640 / 43) < .. < 15\}$
 have $b: (640 / 43) < x < 15$ using x by auto
 have $\text{real } 14 \leq x \wedge x < \text{real } 14 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 34 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 34 + 1 \wedge \text{real } 29 \leq 2 * x \wedge$
 $2 * x < \text{real } 29 + 1 \wedge$
 $\text{real } 27 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 27 + 1$
 using b unfolding alpha-r-def Hc-r-def lam-r-def by linarith
 hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 14) - 12 * \text{lam-r} * ($
 $\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 14)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 34)\ (2 * x$
 $- \text{real } 29)\ (2 * ($
 $\text{lam-r} * x) - \text{real } 27)$
 using R-on-piece by blast
 show ?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-
 pha-r-def Hc-r-def
 lam-r-def
 using b by (simp add: field-simps power2-eq-square)
 qed

lemma inner-piece-102: $x \in \{15 < .. < (560 / 37)\} \implies$
 $R\text{-}r\ x = (157 / 10) * x + (-200)$
proof –
 assume $x: x \in \{15 < .. < (560 / 37)\}$
 have $b: 15 < x < (560 / 37)$ using x by auto
 have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 34 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 34 + 1 \wedge \text{real } 30 \leq 2 * x \wedge$
 $2 * x < \text{real } 30 + 1 \wedge$
 $\text{real } 27 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 27 + 1$
 using b unfolding alpha-r-def Hc-r-def lam-r-def by linarith
 hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} * ($
 $\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 34)\ (2 * x$
 $- \text{real } 30)\ (2 * ($
 $\text{lam-r} * x) - \text{real } 27)$
 using R-on-piece by blast
 show ?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-
 pha-r-def Hc-r-def
 lam-r-def
 using b by (simp add: field-simps power2-eq-square)

qed

lemma *inner-piece-103*: $x \in \{(560 / 37) <..<(350 / 23)\} \implies$
 $R\text{-}r\ x = (813 / 40) * x + (- 270)$

proof –

assume $x: x \in \{(560 / 37) <..<(350 / 23)\}$
have $b: (560 / 37) < x < (350 / 23)$ **using** x **by** *auto*
have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 34 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 34 + 1 \wedge \text{real } 30 \leq 2 * x \wedge$
 $2 * x < \text{real } 30 + 1 \wedge$
 $\text{real } 28 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 28 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 34)\ (2 * x$
 $- \text{real } 30)\ (2 * (\text{lam-r} * x) - \text{real } 28)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)

qed

lemma *inner-piece-104*: $x \in \{(350 / 23) <..<(660 / 43)\} \implies$
 $R\text{-}r\ x = (767 / 40) * x + (- 505 / 2)$

proof –

assume $x: x \in \{(350 / 23) <..<(660 / 43)\}$
have $b: (350 / 23) < x < (660 / 43)$ **using** x **by** *auto*
have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 35 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 35 + 1 \wedge \text{real } 30 \leq 2 * x \wedge$
 $2 * x < \text{real } 30 + 1 \wedge$
 $\text{real } 28 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 28 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 35)\ (2 * x$
 $- \text{real } 30)\ (2 * (\text{lam-r} * x) - \text{real } 28)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)

qed

lemma *inner-piece-105*: $x \in \{(660 / 43) <..< (31 / 2)\} \implies$
 $R\text{-}r\ x = (767 / 40) * x + (- 505 / 2)$
proof –
assume $x: x \in \{(660 / 43) <..< (31 / 2)\}$
have $b: (660 / 43) < x < (31 / 2)$ **using** x **by** *auto*
have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 35 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 35 + 1 \wedge \text{real } 30 \leq 2 * x \wedge$
 $2 * x < \text{real } 30 + 1 \wedge$
 $\text{real } 28 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 28 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 35)\ (2 * x$
 $- \text{real } 30)\ (2 * (\text{lam-r} * x) - \text{real } 28)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-106*: $x \in \{(31 / 2) <..< (360 / 23)\} \implies$
 $R\text{-}r\ x = (807 / 40) * x + (- 268)$
proof –
assume $x: x \in \{(31 / 2) <..< (360 / 23)\}$
have $b: (31 / 2) < x < (360 / 23)$ **using** x **by** *auto*
have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 35 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 35 + 1 \wedge \text{real } 31 \leq 2 * x \wedge$
 $2 * x < \text{real } 31 + 1 \wedge$
 $\text{real } 28 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 28 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 35)\ (2 * x$
 $- \text{real } 31)\ (2 * (\text{lam-r} * x) - \text{real } 28)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-107*: $x \in \{(360 / 23) < .. < (580 / 37)\} \implies$
 $R\text{-}r\ x = (761 / 40) * x + (-250)$
proof –
assume $x: x \in \{(360 / 23) < .. < (580 / 37)\}$
have $b: (360 / 23) < x < (580 / 37)$ **using** x **by** *auto*
have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 36 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 36 + 1 \wedge \text{real } 31 \leq 2 * x \wedge$
 $2 * x < \text{real } 31 + 1 \wedge$
 $\text{real } 28 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 28 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} *$
 $(\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 36)\ (2 * x$
 $- \text{real } 31)\ (2 *$
 $(\text{lam-r} * x) - \text{real } 28)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-108*: $x \in \{(580 / 37) < .. < (680 / 43)\} \implies$
 $R\text{-}r\ x = (167 / 8) * x + (-279)$
proof –
assume $x: x \in \{(580 / 37) < .. < (680 / 43)\}$
have $b: (580 / 37) < x < (680 / 43)$ **using** x **by** *auto*
have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 36 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 36 + 1 \wedge \text{real } 31 \leq 2 * x \wedge$
 $2 * x < \text{real } 31 + 1 \wedge$
 $\text{real } 29 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 29 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} *$
 $(\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 36)\ (2 * x$
 $- \text{real } 31)\ (2 *$
 $(\text{lam-r} * x) - \text{real } 29)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-109*: $x \in \{(680 / 43) < .. < 16\} \implies$

$R\text{-}r\ x = (353 / 20) * x + (- 228)$
proof –
assume $x: x \in \{(680 / 43) < .. < 16\}$
have $b: (680 / 43) < x < 16$ **using** x **by** *auto*
have $\text{real } 15 \leq x \wedge x < \text{real } 15 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 36 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 36 + 1 \wedge \text{real } 31 \leq 2 * x \wedge$
 $2 * x < \text{real } 31 + 1 \wedge$
 $\text{real } 29 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 29 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 15) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 15)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 36)\ (2 * x$
 $- \text{real } 31)\ (2 * (\text{lam-r} * x) - \text{real } 29)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-110*: $x \in \{16 < .. < (370 / 23)\} \implies R\text{-}r\ x = (84 / 5) * x + (- 244)$
proof –
assume $x: x \in \{16 < .. < (370 / 23)\}$
have $b: 16 < x < (370 / 23)$ **using** x **by** *auto*
have $\text{real } 16 \leq x \wedge x < \text{real } 16 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 36 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 36 + 1 \wedge \text{real } 32 \leq 2 * x \wedge$
 $2 * x < \text{real } 32 + 1 \wedge$
 $\text{real } 29 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 29 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 16) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc\ (x - \text{real } 16)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * \text{Hc-r} * x - \text{real } 36)\ (2 * x$
 $- \text{real } 32)\ (2 * (\text{lam-r} * x) - \text{real } 29)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-111*: $x \in \{(370 / 23) < .. < (600 / 37)\} \implies$
 $R\text{-}r\ x = (313 / 20) * x + (- 451 / 2)$

proof –

assume $x: x \in \{(370 / 23) < .. < (600 / 37)\}$
have $b: (370 / 23) < x < (600 / 37)$ **using** x **by** *auto*
have $\text{real } 16 \leq x \wedge x < \text{real } 16 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 37 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 37 + 1 \wedge \text{real } 32 \leq 2 * x \wedge$
 $2 * x < \text{real } 32 + 1 \wedge$
 $\text{real } 29 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 29 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 16) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc (x - \text{real } 16) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 37) (2 * x$
 $- \text{real } 32) (2 * (\text{lam-r} * x) - \text{real } 29)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-112*: $x \in \{(600 / 37) < .. < (700 / 43)\} \implies$

$R\text{-r } x = (811 / 40) * x + (- 601 / 2)$

proof –

assume $x: x \in \{(600 / 37) < .. < (700 / 43)\}$
have $b: (600 / 37) < x < (700 / 43)$ **using** x **by** *auto*
have $\text{real } 16 \leq x \wedge x < \text{real } 16 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 37 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 37 + 1 \wedge \text{real } 32 \leq 2 * x \wedge$
 $2 * x < \text{real } 32 + 1 \wedge$
 $\text{real } 30 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 30 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 16) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc (x - \text{real } 16) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 37) (2 * x$
 $- \text{real } 32) (2 * (\text{lam-r} * x) - \text{real } 30)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
 $\text{pha-r-def Hc-r-def lam-r-def}$
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-113*: $x \in \{(700 / 43) < .. < (33 / 2)\} \implies$

$R\text{-r } x = (811 / 40) * x + (- 601 / 2)$

proof –

assume $x: x \in \{(700 / 43) <..<(33 / 2)\}$
have $b: (700 / 43) < x < (33 / 2)$ **using** x **by** *auto*
have $\text{real } 16 \leq x \wedge x < \text{real } 16 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 37 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 37 + 1 \wedge \text{real } 32 \leq 2 * x \wedge$
 $2 * x < \text{real } 32 + 1 \wedge$
 $\text{real } 30 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 30 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 16) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc (x - \text{real } 16) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 37) (2 * x$
 $- \text{real } 32) (2 * (\text{lam-r} * x) - \text{real } 30)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-114: $x \in \{(33 / 2) <..<(380 / 23)\} \implies$
 $R\text{-r } x = (851 / 40) * x + (- 317)$
proof –
assume $x: x \in \{(33 / 2) <..<(380 / 23)\}$
have $b: (33 / 2) < x < (380 / 23)$ **using** x **by** *auto*
have $\text{real } 16 \leq x \wedge x < \text{real } 16 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 37 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 37 + 1 \wedge \text{real } 33 \leq 2 * x \wedge$
 $2 * x < \text{real } 33 + 1 \wedge$
 $\text{real } 30 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 30 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 16) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc (x - \text{real } 16) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 37) (2 * x$
 $- \text{real } 33) (2 * (\text{lam-r} * x) - \text{real } 30)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma inner-piece-115: $x \in \{(380 / 23) <..<(50 / 3)\} \implies$
 $R\text{-r } x = (161 / 8) * x + (- 298)$
proof –
assume $x: x \in \{(380 / 23) <..<(50 / 3)\}$

have $b: (380 / 23) < x x < (50 / 3)$ **using** x **by** *auto*
have $\text{real } 16 \leq x \wedge x < \text{real } 16 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 38 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 38 + 1 \wedge \text{real } 33 \leq 2 * x \wedge$
 $2 * x < \text{real } 33 + 1 \wedge$
 $\text{real } 30 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 30 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 16) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc (x - \text{real } 16) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 38) (2 * x$
 $- \text{real } 33) (2 * (\text{lam-r} * x) - \text{real } 30)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-116*: $x \in \{(50 / 3) < .. < (720 / 43)\} \implies$
 $R\text{-r } x = (799 / 40) * x + (- 591 / 2)$
proof –
assume $x: x \in \{(50 / 3) < .. < (720 / 43)\}$
have $b: (50 / 3) < x x < (720 / 43)$ **using** x **by** *auto*
have $\text{real } 16 \leq x \wedge x < \text{real } 16 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 38 \leq 2 * \text{Hc-r} * x \wedge 2 * \text{Hc-r} * x < \text{real } 38 + 1 \wedge \text{real } 33 \leq 2 * x \wedge$
 $2 * x < \text{real } 33 + 1 \wedge$
 $\text{real } 30 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 30 + 1$
using b **unfolding** $\text{alpha-r-def Hc-r-def lam-r-def}$ **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 16) - 12 * \text{lam-r} * (\text{alpha-r} * x -$
 $\text{real } 1))$
 $+ Qc (x - \text{real } 16) (\text{alpha-r} * x - \text{real } 1) (2 * \text{Hc-r} * x - \text{real } 38) (2 * x$
 $- \text{real } 33) (2 * (\text{lam-r} * x) - \text{real } 30)$
using $R\text{-on-piece}$ **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-117*: $x \in \{(720 / 43) < .. < (620 / 37)\} \implies$
 $R\text{-r } x = (67 / 4) * x + (- 483 / 2)$
proof –
assume $x: x \in \{(720 / 43) < .. < (620 / 37)\}$
have $b: (720 / 43) < x x < (620 / 37)$ **using** x **by** *auto*

have $real\ 16 \leq x \wedge x < real\ 16 + 1 \wedge real\ 1 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < real\ 1 + 1 \wedge$
 $real\ 38 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 38 + 1 \wedge real\ 33 \leq 2 * x \wedge$
 $2 * x < real\ 33 + 1 \wedge$
 $real\ 30 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < real\ 30 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$ **by** $linarith$
hence $R: R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - real\ 16) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x -$
 $real\ 1))$
 $+ Qc\ (x - real\ 16)\ (\alpha\text{-}r * x - real\ 1)\ (2 * Hc\text{-}r * x - real\ 38)\ (2 * x$
 $- real\ 33)\ (2 * (\text{lam}\text{-}r * x) - real\ 30)$
using $R\text{-on-piece}$ **by** $blast$
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$
using b **by** $(simp\ add: field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}118: x \in \{(620 / 37) < .. < (390 / 23)\} \implies$
 $R\text{-}r\ x = (93 / 5) * x + (- 545 / 2)$
proof –
assume $x: x \in \{(620 / 37) < .. < (390 / 23)\}$
have $b: (620 / 37) < x < (390 / 23)$ **using** x **by** $auto$
have $real\ 16 \leq x \wedge x < real\ 16 + 1 \wedge real\ 1 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < real\ 1 + 1 \wedge$
 $real\ 38 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 38 + 1 \wedge real\ 33 \leq 2 * x \wedge$
 $2 * x < real\ 33 + 1 \wedge$
 $real\ 31 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < real\ 31 + 1$
using b **unfolding** $\alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$ **by** $linarith$
hence $R: R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - real\ 16) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x -$
 $real\ 1))$
 $+ Qc\ (x - real\ 16)\ (\alpha\text{-}r * x - real\ 1)\ (2 * Hc\text{-}r * x - real\ 38)\ (2 * x$
 $- real\ 33)\ (2 * (\text{lam}\text{-}r * x) - real\ 31)$
using $R\text{-on-piece}$ **by** $blast$
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ \alpha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ \text{lam}\text{-}r\text{-}def$
using b **by** $(simp\ add: field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}119: x \in \{(390 / 23) < .. < 17\} \implies$
 $R\text{-}r\ x = (349 / 20) * x + (- 253)$
proof –
assume $x: x \in \{(390 / 23) < .. < 17\}$
have $b: (390 / 23) < x < 17$ **using** x **by** $auto$
have $real\ 16 \leq x \wedge x < real\ 16 + 1 \wedge real\ 1 \leq \alpha\text{-}r * x \wedge$

$\alpha\text{-}r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 39 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 39 + 1 \wedge \text{real } 33 \leq 2 * x \wedge$
 $2 * x < \text{real } 33 + 1 \wedge$
 $\text{real } 31 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 31 + 1$
using *b* **unfolding** $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$ **by** *linarith*
hence R : $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 16) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 1))$
 $+ Qc\ (x - \text{real } 16)\ (\alpha\text{-}r * x - \text{real } 1)\ (2 * Hc\text{-}r * x - \text{real } 39)\ (2 * x - \text{real } 33)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 31)$
using *R-on-piece* **by** *blast*
show *?thesis* **unfolding** R $Qc\text{-}def$ $d0\text{-}def$ $es\text{-}def$ $pos\text{-}r\text{-}def$ $min\text{-}def$ $max\text{-}def$ $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-120*: $x \in \{17 < .. < (740 / 43)\} \implies R\text{-}r\ x = (83 / 5) * x + (-270)$

proof –

assume x : $x \in \{17 < .. < (740 / 43)\}$
have b : $17 < x < (740 / 43)$ **using** x **by** *auto*
have $\text{real } 17 \leq x \wedge x < \text{real } 17 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 39 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 39 + 1 \wedge \text{real } 34 \leq 2 * x \wedge$
 $2 * x < \text{real } 34 + 1 \wedge$
 $\text{real } 31 \leq 2 * (\text{lam}\text{-}r * x) \wedge 2 * (\text{lam}\text{-}r * x) < \text{real } 31 + 1$
using *b* **unfolding** $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$ **by** *linarith*
hence R : $R\text{-}r\ x = x * (4 * \text{lam}\text{-}r + 2 * \text{lam}\text{-}r * (x - \text{real } 17) - 12 * \text{lam}\text{-}r * (\alpha\text{-}r * x - \text{real } 1))$
 $+ Qc\ (x - \text{real } 17)\ (\alpha\text{-}r * x - \text{real } 1)\ (2 * Hc\text{-}r * x - \text{real } 39)\ (2 * x - \text{real } 34)\ (2 * (\text{lam}\text{-}r * x) - \text{real } 31)$
using *R-on-piece* **by** *blast*
show *?thesis* **unfolding** R $Qc\text{-}def$ $d0\text{-}def$ $es\text{-}def$ $pos\text{-}r\text{-}def$ $min\text{-}def$ $max\text{-}def$ $\alpha\text{-}r\text{-}def$ $Hc\text{-}r\text{-}def$ $\text{lam}\text{-}r\text{-}def$
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-121*: $x \in \{(740 / 43) < .. < (640 / 37)\} \implies$

$R\text{-}r\ x = (83 / 5) * x + (-270)$

proof –

assume x : $x \in \{(740 / 43) < .. < (640 / 37)\}$
have b : $(740 / 43) < x < (640 / 37)$ **using** x **by** *auto*
have $\text{real } 17 \leq x \wedge x < \text{real } 17 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 1 + 1 \wedge$

$real\ 39 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 39 + 1 \wedge real\ 34 \leq 2 * x \wedge$
 $2 * x < real\ 34 + 1 \wedge$
 $real\ 31 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 31 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 17) - 12 * lam-r * (alpha-r * x -$
 $real\ 1))$
 $+ Qc\ (x - real\ 17)\ (alpha-r * x - real\ 1)\ (2 * Hc-r * x - real\ 39)\ (2 * x$
 $- real\ 34)\ (2 * (lam-r * x) - real\ 31)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-122*: $x \in \{(640 / 37) < .. < (400 / 23)\} \implies$
 $R-r\ x = (849 / 40) * x + (- 350)$
proof –
assume *x*: $x \in \{(640 / 37) < .. < (400 / 23)\}$
have *b*: $(640 / 37) < x < (400 / 23)$ **using** *x* **by** *auto*
have $real\ 17 \leq x \wedge x < real\ 17 + 1 \wedge real\ 1 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 1 + 1 \wedge$
 $real\ 39 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 39 + 1 \wedge real\ 34 \leq 2 * x \wedge$
 $2 * x < real\ 34 + 1 \wedge$
 $real\ 32 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 32 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 17) - 12 * lam-r * (alpha-r * x -$
 $real\ 1))$
 $+ Qc\ (x - real\ 17)\ (alpha-r * x - real\ 1)\ (2 * Hc-r * x - real\ 39)\ (2 * x$
 $- real\ 34)\ (2 * (lam-r * x) - real\ 32)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-123*: $x \in \{(400 / 23) < .. < (35 / 2)\} \implies$
 $R-r\ x = (803 / 40) * x + (- 330)$

proof –
assume *x*: $x \in \{(400 / 23) < .. < (35 / 2)\}$
have *b*: $(400 / 23) < x < (35 / 2)$ **using** *x* **by** *auto*
have $real\ 17 \leq x \wedge x < real\ 17 + 1 \wedge real\ 1 \leq alpha-r * x \wedge$
 $alpha-r * x < real\ 1 + 1 \wedge$
 $real\ 40 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 40 + 1 \wedge real\ 34 \leq 2 * x \wedge$

$2 * x < \text{real } 34 + 1 \wedge$
 $\text{real } 32 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 32 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 17) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 1))$
 $+ Qc (x - \text{real } 17) (\text{alpha-r} * x - \text{real } 1) (2 * Hc\text{-r} * x - \text{real } 40) (2 * x - \text{real } 34) (2 * (\text{lam-r} * x) - \text{real } 32)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-124*: $x \in \{(35 / 2) < .. < (760 / 43)\} \implies$
 $R\text{-r } x = (843 / 40) * x + (- 695 / 2)$
proof –
assume *x*: $x \in \{(35 / 2) < .. < (760 / 43)\}$
have *b*: $(35 / 2) < x < (760 / 43)$ **using** *x* **by** *auto*
have $\text{real } 17 \leq x \wedge x < \text{real } 17 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 40 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 40 + 1 \wedge \text{real } 35 \leq 2 * x \wedge$
 $2 * x < \text{real } 35 + 1 \wedge$
 $\text{real } 32 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 32 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 17) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 1))$
 $+ Qc (x - \text{real } 17) (\text{alpha-r} * x - \text{real } 1) (2 * Hc\text{-r} * x - \text{real } 40) (2 * x - \text{real } 35) (2 * (\text{lam-r} * x) - \text{real } 32)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-125*: $x \in \{(760 / 43) < .. < (410 / 23)\} \implies$
 $R\text{-r } x = (357 / 20) * x + (- 581 / 2)$
proof –
assume *x*: $x \in \{(760 / 43) < .. < (410 / 23)\}$
have *b*: $(760 / 43) < x < (410 / 23)$ **using** *x* **by** *auto*
have $\text{real } 17 \leq x \wedge x < \text{real } 17 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$
 $\text{alpha-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 40 \leq 2 * Hc\text{-r} * x \wedge 2 * Hc\text{-r} * x < \text{real } 40 + 1 \wedge \text{real } 35 \leq 2 * x \wedge$
 $2 * x < \text{real } 35 + 1 \wedge$

$real\ 32 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 32 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 17) - 12 * lam-r * (alpha-r * x - real\ 1))$
 $+ Qc\ (x - real\ 17)\ (alpha-r * x - real\ 1)\ (2 * Hc-r * x - real\ 40)\ (2 * x - real\ 35)\ (2 * (lam-r * x) - real\ 32)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-126*: $x \in \{(410 / 23) < .. < (660 / 37)\} \implies R-r\ x = (167 / 10) * x + (-270)$
proof –
assume *x*: $x \in \{(410 / 23) < .. < (660 / 37)\}$
have *b*: $(410 / 23) < x < (660 / 37)$ **using** *x* **by** *auto*
have $real\ 17 \leq x \wedge x < real\ 17 + 1 \wedge real\ 1 \leq alpha-r * x \wedge alpha-r * x < real\ 1 + 1 \wedge real\ 41 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 41 + 1 \wedge real\ 35 \leq 2 * x \wedge 2 * x < real\ 35 + 1 \wedge real\ 32 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 32 + 1$
using *b* **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence *R*: $R-r\ x = x * (4 * lam-r + 2 * lam-r * (x - real\ 17) - 12 * lam-r * (alpha-r * x - real\ 1))$
 $+ Qc\ (x - real\ 17)\ (alpha-r * x - real\ 1)\ (2 * Hc-r * x - real\ 41)\ (2 * x - real\ 35)\ (2 * (lam-r * x) - real\ 32)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def alpha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-127*: $x \in \{(660 / 37) < .. < 18\} \implies R-r\ x = (371 / 20) * x + (-303)$
proof –
assume *x*: $x \in \{(660 / 37) < .. < 18\}$
have *b*: $(660 / 37) < x < 18$ **using** *x* **by** *auto*
have $real\ 17 \leq x \wedge x < real\ 17 + 1 \wedge real\ 1 \leq alpha-r * x \wedge alpha-r * x < real\ 1 + 1 \wedge real\ 41 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < real\ 41 + 1 \wedge real\ 35 \leq 2 * x \wedge 2 * x < real\ 35 + 1 \wedge real\ 33 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < real\ 33 + 1$

using b **unfolding** α - r -def Hc - r -def lam - r -def **by** $linarith$
hence R : $R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - real\ 17) - 12 * lam\text{-}r * (\alpha\text{-}r * x - real\ 1))$
 $+ Qc\ (x - real\ 17)\ (\alpha\text{-}r * x - real\ 1)\ (2 * Hc\text{-}r * x - real\ 41)\ (2 * x - real\ 35)\ (2 * (lam\text{-}r * x) - real\ 33)$
using R -on-piece **by** $blast$
show $?thesis$ **unfolding** R Qc -def $d0$ -def es -def pos - r -def min -def max -def α - r -def Hc - r -def lam - r -def
using b **by** ($simp\ add$: $field\text{-}simps\ power2\text{-}eq\text{-}square$)
qed

lemma $inner\text{-}piece\text{-}128$: $x \in \{18 <..<(780 / 43)\} \implies$

$R\text{-}r\ x = (177 / 10) * x + (-321)$

proof –

assume x : $x \in \{18 <..<(780 / 43)\}$

have b : $18 < x < (780 / 43)$ **using** x **by** $auto$

have $real\ 18 \leq x \wedge x < real\ 18 + 1 \wedge real\ 1 \leq \alpha\text{-}r * x \wedge$

$\alpha\text{-}r * x < real\ 1 + 1 \wedge$

$real\ 41 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 41 + 1 \wedge real\ 36 \leq 2 * x \wedge$

$2 * x < real\ 36 + 1 \wedge$

$real\ 33 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < real\ 33 + 1$

using b **unfolding** α - r -def Hc - r -def lam - r -def **by** $linarith$

hence R : $R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - real\ 18) - 12 * lam\text{-}r * (\alpha\text{-}r * x - real\ 1))$
 $+ Qc\ (x - real\ 18)\ (\alpha\text{-}r * x - real\ 1)\ (2 * Hc\text{-}r * x - real\ 41)\ (2 * x - real\ 36)\ (2 * (lam\text{-}r * x) - real\ 33)$

using R -on-piece **by** $blast$

show $?thesis$ **unfolding** R Qc -def $d0$ -def es -def pos - r -def min -def max -def α - r -def Hc - r -def lam - r -def
using b **by** ($simp\ add$: $field\text{-}simps\ power2\text{-}eq\text{-}square$)

qed

lemma $inner\text{-}piece\text{-}129$: $x \in \{(780 / 43) <..<(420 / 23)\} \implies$

$R\text{-}r\ x = (177 / 10) * x + (-321)$

proof –

assume x : $x \in \{(780 / 43) <..<(420 / 23)\}$

have b : $(780 / 43) < x < (420 / 23)$ **using** x **by** $auto$

have $real\ 18 \leq x \wedge x < real\ 18 + 1 \wedge real\ 1 \leq \alpha\text{-}r * x \wedge$

$\alpha\text{-}r * x < real\ 1 + 1 \wedge$

$real\ 41 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 41 + 1 \wedge real\ 36 \leq 2 * x \wedge$

$2 * x < real\ 36 + 1 \wedge$

$real\ 33 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < real\ 33 + 1$

using b **unfolding** α - r -def Hc - r -def lam - r -def **by** $linarith$

hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 18) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 1))$
 $+ Qc\ (x - \text{real } 18)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * Hc\text{-}r * x - \text{real } 41)\ (2 * x - \text{real } 36)\ (2 * (\text{lam-r} * x) - \text{real } 33)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ al\text{-}pha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}130$: $x \in \{(420 / 23) < .. < (680 / 37)\} \implies R\text{-}r\ x = (331 / 20) * x + (-300)$

proof –
assume x : $x \in \{(420 / 23) < .. < (680 / 37)\}$
have b : $(420 / 23) < x < (680 / 37)$ **using** x **by** $auto$
have $\text{real } 18 \leq x \wedge x < \text{real } 18 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 1 + 1 \wedge \text{real } 42 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 42 + 1 \wedge \text{real } 36 \leq 2 * x \wedge 2 * x < \text{real } 36 + 1 \wedge \text{real } 33 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 33 + 1$
using b **unfolding** $\text{alpha-r}\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$ **by** $linarith$
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 18) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 1))$
 $+ Qc\ (x - \text{real } 18)\ (\text{alpha-r} * x - \text{real } 1)\ (2 * Hc\text{-}r * x - \text{real } 42)\ (2 * x - \text{real } 36)\ (2 * (\text{lam-r} * x) - \text{real } 33)$
using $R\text{-on-piece}$ **by** blast
show $?thesis\ unfolding\ R\ Qc\text{-}def\ d0\text{-}def\ es\text{-}def\ pos\text{-}r\text{-}def\ min\text{-}def\ max\text{-}def\ al\text{-}pha\text{-}r\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$
using b **by** $(simp\ add:\ field\text{-}simps\ power2\text{-}eq\text{-}square)$
qed

lemma $inner\text{-}piece\text{-}131$: $x \in \{(680 / 37) < .. < (37 / 2)\} \implies R\text{-}r\ x = (847 / 40) * x + (-385)$

proof –
assume x : $x \in \{(680 / 37) < .. < (37 / 2)\}$
have b : $(680 / 37) < x < (37 / 2)$ **using** x **by** $auto$
have $\text{real } 18 \leq x \wedge x < \text{real } 18 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge \text{alpha-r} * x < \text{real } 1 + 1 \wedge \text{real } 42 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 42 + 1 \wedge \text{real } 36 \leq 2 * x \wedge 2 * x < \text{real } 36 + 1 \wedge \text{real } 34 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 34 + 1$
using b **unfolding** $\text{alpha-r}\text{-}def\ Hc\text{-}r\text{-}def\ lam\text{-}r\text{-}def$ **by** $linarith$
hence R : $R\text{-}r\ x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 18) - 12 * \text{lam-r} * (\text{alpha-r} * x - \text{real } 1))$

$(\text{alpha-r} * x - \text{real } 1))$
 $+ Qc (x - \text{real } 18) (\text{alpha-r} * x - \text{real } 1) (2 * Hc-r * x - \text{real } 42) (2 * x - \text{real } 36) (2 * (lam-r * x) - \text{real } 34)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-132*: $x \in \{(37 / 2) < .. < (800 / 43)\} \implies$

$R-r x = (887 / 40) * x + (- 807 / 2)$

proof –

assume $x: x \in \{(37 / 2) < .. < (800 / 43)\}$

have $b: (37 / 2) < x < (800 / 43)$ **using** x **by** *auto*

have $\text{real } 18 \leq x \wedge x < \text{real } 18 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$

$\text{alpha-r} * x < \text{real } 1 + 1 \wedge$

$\text{real } 42 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 42 + 1 \wedge \text{real } 37 \leq 2 * x \wedge$

$2 * x < \text{real } 37 + 1 \wedge$

$\text{real } 34 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 34 + 1$

using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*

hence $R: R-r x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 18) - 12 * lam-r * (\text{alpha-r} * x - \text{real } 1))$

$+ Qc (x - \text{real } 18) (\text{alpha-r} * x - \text{real } 1) (2 * Hc-r * x - \text{real } 42) (2 * x - \text{real } 37) (2 * (lam-r * x) - \text{real } 34)$
using *R-on-piece* **by** *blast*

show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*

lam-r-def

using b **by** (*simp add: field-simps power2-eq-square*)

qed

lemma *inner-piece-133*: $x \in \{(800 / 43) < .. < (430 / 23)\} \implies$

$R-r x = (379 / 20) * x + (- 687 / 2)$

proof –

assume $x: x \in \{(800 / 43) < .. < (430 / 23)\}$

have $b: (800 / 43) < x < (430 / 23)$ **using** x **by** *auto*

have $\text{real } 18 \leq x \wedge x < \text{real } 18 + 1 \wedge \text{real } 1 \leq \text{alpha-r} * x \wedge$

$\text{alpha-r} * x < \text{real } 1 + 1 \wedge$

$\text{real } 42 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 42 + 1 \wedge \text{real } 37 \leq 2 * x \wedge$

$2 * x < \text{real } 37 + 1 \wedge$

$\text{real } 34 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 34 + 1$

using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*

hence $R: R-r x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 18) - 12 * lam-r * (\text{alpha-r} * x -$

$\text{real } 1))$
 $+ Qc (x - \text{real } 18) (\alpha\text{-}r * x - \text{real } 1) (2 * Hc\text{-}r * x - \text{real } 42) (2 * x$
 $- \text{real } 37) (2 *$
 $(\text{lam-}r * x) - \text{real } 34)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-134*: $x \in \{(430 / 23) < .. < (700 / 37)\} \implies$
 $R\text{-}r x = (89 / 5) * x + (- 322)$
proof –
assume $x: x \in \{(430 / 23) < .. < (700 / 37)\}$
have $b: (430 / 23) < x < (700 / 37)$ **using** x **by** *auto*
have $\text{real } 18 \leq x \wedge x < \text{real } 18 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 43 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 43 + 1 \wedge \text{real } 37 \leq 2 * x \wedge$
 $2 * x < \text{real } 37 + 1 \wedge$
 $\text{real } 34 \leq 2 * (\text{lam-}r * x) \wedge 2 * (\text{lam-}r * x) < \text{real } 34 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r x = x * (4 * \text{lam-}r + 2 * \text{lam-}r * (x - \text{real } 18) - 12 * \text{lam-}r *$
 $(\alpha\text{-}r * x -$
 $\text{real } 1))$
 $+ Qc (x - \text{real } 18) (\alpha\text{-}r * x - \text{real } 1) (2 * Hc\text{-}r * x - \text{real } 43) (2 * x$
 $- \text{real } 37) (2 *$
 $(\text{lam-}r * x) - \text{real } 34)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-135*: $x \in \{(700 / 37) < .. < 19\} \implies$
 $R\text{-}r x = (393 / 20) * x + (- 357)$
proof –
assume $x: x \in \{(700 / 37) < .. < 19\}$
have $b: (700 / 37) < x < 19$ **using** x **by** *auto*
have $\text{real } 18 \leq x \wedge x < \text{real } 18 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 43 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 43 + 1 \wedge \text{real } 37 \leq 2 * x \wedge$
 $2 * x < \text{real } 37 + 1 \wedge$
 $\text{real } 35 \leq 2 * (\text{lam-}r * x) \wedge 2 * (\text{lam-}r * x) < \text{real } 35 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r x = x * (4 * \text{lam-}r + 2 * \text{lam-}r * (x - \text{real } 18) - 12 * \text{lam-}r *$
 $(\alpha\text{-}r * x -$
 $\text{real } 1))$

$+ Qc (x - \text{real } 18) (\alpha\text{-}r * x - \text{real } 1) (2 * Hc\text{-}r * x - \text{real } 43) (2 * x - \text{real } 37) (2 * (lam\text{-}r * x) - \text{real } 35)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-136*: $x \in \{19 < .. < (820 / 43)\} \implies R\text{-}r x = (94 / 5) * x + (-376)$

proof –
assume *x*: $x \in \{19 < .. < (820 / 43)\}$
have *b*: $19 < x < (820 / 43)$ **using** *x by auto*
have $\text{real } 19 \leq x \wedge x < \text{real } 19 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 1 + 1 \wedge \text{real } 43 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 43 + 1 \wedge \text{real } 38 \leq 2 * x \wedge 2 * x < \text{real } 38 + 1 \wedge \text{real } 35 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 35 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence *R*: $R\text{-}r x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 19) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 1)) + Qc (x - \text{real } 19) (\alpha\text{-}r * x - \text{real } 1) (2 * Hc\text{-}r * x - \text{real } 43) (2 * x - \text{real } 38) (2 * (lam\text{-}r * x) - \text{real } 35)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b by (simp add: field-simps power2-eq-square)*
qed

lemma *inner-piece-137*: $x \in \{(820 / 43) < .. < (440 / 23)\} \implies$

$R\text{-}r x = (94 / 5) * x + (-376)$
proof –
assume *x*: $x \in \{(820 / 43) < .. < (440 / 23)\}$
have *b*: $(820 / 43) < x < (440 / 23)$ **using** *x by auto*
have $\text{real } 19 \leq x \wedge x < \text{real } 19 + 1 \wedge \text{real } 1 \leq \alpha\text{-}r * x \wedge \alpha\text{-}r * x < \text{real } 1 + 1 \wedge \text{real } 43 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < \text{real } 43 + 1 \wedge \text{real } 38 \leq 2 * x \wedge 2 * x < \text{real } 38 + 1 \wedge \text{real } 35 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < \text{real } 35 + 1$
using *b unfolding alpha-r-def Hc-r-def lam-r-def by linarith*
hence *R*: $R\text{-}r x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - \text{real } 19) - 12 * lam\text{-}r * (\alpha\text{-}r * x - \text{real } 1)) + Qc (x - \text{real } 19) (\alpha\text{-}r * x - \text{real } 1) (2 * Hc\text{-}r * x - \text{real } 43) (2 * x$

$- \text{real } 38) (2 * (lam-r * x) - \text{real } 35)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-138*: $x \in \{(440 / 23) < .. < (720 / 37)\} \implies$
 $R-r \ x = (353 / 20) * x + (- 354)$

proof $-$
assume $x: x \in \{(440 / 23) < .. < (720 / 37)\}$
have $b: (440 / 23) < x < (720 / 37)$ **using** x **by** *auto*
have $\text{real } 19 \leq x \wedge x < \text{real } 19 + 1 \wedge \text{real } 1 \leq \alpha-r * x \wedge$
 $\alpha-r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 44 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 44 + 1 \wedge \text{real } 38 \leq 2 * x \wedge$
 $2 * x < \text{real } 38 + 1 \wedge$
 $\text{real } 35 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 35 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r \ x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 19) - 12 * lam-r * (alpha-r * x -$
 $\text{real } 1))$
 $+ Qc \ (x - \text{real } 19) \ (\alpha-r * x - \text{real } 1) \ (2 * Hc-r * x - \text{real } 44) \ (2 * x$
 $- \text{real } 38) (2 * (lam-r * x) - \text{real } 35)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-pha-r-def Hc-r-def lam-r-def*
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-139*: $x \in \{(720 / 37) < .. < (39 / 2)\} \implies$
 $R-r \ x = (891 / 40) * x + (- 444)$

proof $-$
assume $x: x \in \{(720 / 37) < .. < (39 / 2)\}$
have $b: (720 / 37) < x < (39 / 2)$ **using** x **by** *auto*
have $\text{real } 19 \leq x \wedge x < \text{real } 19 + 1 \wedge \text{real } 1 \leq \alpha-r * x \wedge$
 $\alpha-r * x < \text{real } 1 + 1 \wedge$
 $\text{real } 44 \leq 2 * Hc-r * x \wedge 2 * Hc-r * x < \text{real } 44 + 1 \wedge \text{real } 38 \leq 2 * x \wedge$
 $2 * x < \text{real } 38 + 1 \wedge$
 $\text{real } 36 \leq 2 * (lam-r * x) \wedge 2 * (lam-r * x) < \text{real } 36 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R-r \ x = x * (4 * lam-r + 2 * lam-r * (x - \text{real } 19) - 12 * lam-r * (alpha-r * x -$
 $\text{real } 1))$
 $+ Qc \ (x - \text{real } 19) \ (\alpha-r * x - \text{real } 1) \ (2 * Hc-r * x - \text{real } 44) \ (2 * x$
 $- \text{real } 38) (2 * (lam-r * x) - \text{real } 35)$

$(\text{lam-r} * x) - \text{real } 36)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-140*: $x \in \{(39 / 2) < .. < (840 / 43)\} \implies$
 $R\text{-r } x = (931 / 40) * x + (- 927 / 2)$
proof –
assume $x: x \in \{(39 / 2) < .. < (840 / 43)\}$
have $b: (39 / 2) < x < (840 / 43)$ **using** x **by** *auto*
have $\text{real } 19 \leq x \wedge x < \text{real } 19 + 1 \wedge \text{real } 1 \leq \alpha\text{-r} * x \wedge$
 $\alpha\text{-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 44 \leq 2 * H\text{c-r} * x \wedge 2 * H\text{c-r} * x < \text{real } 44 + 1 \wedge \text{real } 39 \leq 2 * x \wedge$
 $2 * x < \text{real } 39 + 1 \wedge$
 $\text{real } 36 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 36 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 19) - 12 * \text{lam-r} * (\alpha\text{-r} * x -$
 $\text{real } 1))$
 $+ Q\text{c } (x - \text{real } 19) (\alpha\text{-r} * x - \text{real } 1) (2 * H\text{c-r} * x - \text{real } 44) (2 * x$
 $- \text{real } 39) (2 * (\text{lam-r} * x) - \text{real } 36)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-141*: $x \in \{(840 / 43) < .. < (450 / 23)\} \implies$
 $R\text{-r } x = (401 / 20) * x + (- 801 / 2)$
proof –
assume $x: x \in \{(840 / 43) < .. < (450 / 23)\}$
have $b: (840 / 43) < x < (450 / 23)$ **using** x **by** *auto*
have $\text{real } 19 \leq x \wedge x < \text{real } 19 + 1 \wedge \text{real } 1 \leq \alpha\text{-r} * x \wedge$
 $\alpha\text{-r} * x < \text{real } 1 + 1 \wedge$
 $\text{real } 44 \leq 2 * H\text{c-r} * x \wedge 2 * H\text{c-r} * x < \text{real } 44 + 1 \wedge \text{real } 39 \leq 2 * x \wedge$
 $2 * x < \text{real } 39 + 1 \wedge$
 $\text{real } 36 \leq 2 * (\text{lam-r} * x) \wedge 2 * (\text{lam-r} * x) < \text{real } 36 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-r } x = x * (4 * \text{lam-r} + 2 * \text{lam-r} * (x - \text{real } 19) - 12 * \text{lam-r} * (\alpha\text{-r} * x -$
 $\text{real } 1))$
 $+ Q\text{c } (x - \text{real } 19) (\alpha\text{-r} * x - \text{real } 1) (2 * H\text{c-r} * x - \text{real } 44) (2 * x$
 $- \text{real } 39) (2 * (\text{lam-r} * x) - \text{real } 36)$

using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using *b* **by** (*simp add: field-simps power2-eq-square*)
qed

lemma *inner-piece-142*: $x \in \{(450 / 23) < .. < 20\} \implies$
 $R\text{-}r\ x = (189 / 10) * x + (- 378)$
proof –
assume $x: x \in \{(450 / 23) < .. < 20\}$
have $b: (450 / 23) < x < 20$ **using** x **by** *auto*
have $real\ 19 \leq x \wedge x < real\ 19 + 1 \wedge real\ 1 \leq \alpha\text{-}r * x \wedge$
 $\alpha\text{-}r * x < real\ 1 + 1 \wedge$
 $real\ 45 \leq 2 * Hc\text{-}r * x \wedge 2 * Hc\text{-}r * x < real\ 45 + 1 \wedge real\ 39 \leq 2 * x \wedge$
 $2 * x < real\ 39 + 1 \wedge$
 $real\ 36 \leq 2 * (lam\text{-}r * x) \wedge 2 * (lam\text{-}r * x) < real\ 36 + 1$
using b **unfolding** *alpha-r-def Hc-r-def lam-r-def* **by** *linarith*
hence $R: R\text{-}r\ x = x * (4 * lam\text{-}r + 2 * lam\text{-}r * (x - real\ 19) - 12 * lam\text{-}r * ($
 $\alpha\text{-}r * x -$
 $real\ 1))$
 $+ Qc\ (x - real\ 19)\ (\alpha\text{-}r * x - real\ 1)\ (2 * Hc\text{-}r * x - real\ 45)\ (2 * x$
 $- real\ 39)\ (2 * ($
 $lam\text{-}r * x) - real\ 36)$
using *R-on-piece* **by** *blast*
show *?thesis unfolding R Qc-def d0-def es-def pos-r-def min-def max-def al-*
pha-r-def Hc-r-def
lam-r-def
using b **by** (*simp add: field-simps power2-eq-square*)
qed

theorem *R-integral-3-20*:
 $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ 3)\ (ereal\ 20)\ (\lambda x. R\text{-}r\ x / x^3) =$
 $322437603634266857629 / 7535670527041937280000$
proof –
have $interval\text{-}lebesgue\text{-}integrable\ lborel\ (ereal\ 3)\ (ereal\ (70 / 23))\ (\lambda x. R\text{-}r\ x /$
 $x^3) \wedge$
 $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ 3)\ (ereal\ (70 / 23))\ (\lambda x. R\text{-}r\ x / x^3)$
 $=$
 $(- (18 / 5) / (70 / 23) - (- 6) / (2 * (70 / 23)^2)) - (- (18 / 5) /$
 $3 - (- 6) / (2 *$
 $3^2))$
by (*rule piece-rat*) (*simp-all add: inner-piece-0*)
hence $s0: interval\text{-}lebesgue\text{-}integrable\ lborel\ (ereal\ 3)\ (ereal\ (70 / 23))\ (\lambda x. R\text{-}r$
 $x /$
 $x^3) \wedge$
 $interval\text{-}lebesgue\text{-}integral\ lborel\ (ereal\ 3)\ (ereal\ (70 / 23))\ (\lambda x. R\text{-}r\ x / x^3)$
 $=$
 $(113 / 14700)$

```

by (simp add: power2-eq-square power3-eq-cube)
have r1: interval-lebesgue-integrable lborel (ereal (70 / 23)) (ereal (120 /
37)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (70 / 23)) (ereal (120 / 37)) (λx. R-r
x /
x ^ 3) = (- (49 / 20) / (120 / 37) - (- 5 / 2) / (2 * (120 / 37) ^ 2))
- (- (49 / 20) /
(70 / 23) - (- 5 / 2) / (2 * (70 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-1)
have s1: interval-lebesgue-integrable lborel (ereal 3) (ereal (120 / 37)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (120 / 37)) (λx. R-r x / x ^
3) =
(3319 / 80640)
using chain[OF - - s0 r1] by (simp add: power2-eq-square power3-eq-cube)
have r2: interval-lebesgue-integrable lborel (ereal (120 / 37)) (ereal (140 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (120 / 37)) (ereal (140 / 43)) (λx. R-r
x /
x ^ 3) = (- (283 / 40) / (140 / 43) - (- 35 / 2) / (2 * (140 / 43) ^
2)) - (- (283 / 40) /
(120 / 37) - (- 35 / 2) / (2 * (120 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-2)
have s2: interval-lebesgue-integrable lborel (ereal 3) (ereal (140 / 43)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (140 / 43)) (λx. R-r x / x ^
3) =
(967 / 22400)
using chain[OF - - s1 r2] by (simp add: power2-eq-square power3-eq-cube)
have r3: interval-lebesgue-integrable lborel (ereal (140 / 43)) (ereal (10 /
3)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (140 / 43)) (ereal (10 / 3)) (λx. R-r x
/
x ^ 3) = (- (283 / 40) / (10 / 3) - (- 35 / 2) / (2 * (10 / 3) ^ 2)) -
(- (283 / 40) /
(140 / 43) - (- 35 / 2) / (2 * (140 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-3)
have s3: interval-lebesgue-integrable lborel (ereal 3) (ereal (10 / 3)) (λx. R-r x
/
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (10 / 3)) (λx. R-r x / x ^ 3)
=
(1249 / 22400)
using chain[OF - - s2 r3] by (simp add: power2-eq-square power3-eq-cube)
have r4: interval-lebesgue-integrable lborel (ereal (10 / 3)) (ereal (80 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (10 / 3)) (ereal (80 / 23)) (λx. R-r x /

```

$$x^3 = (- (277 / 40) / (80 / 23) - (- 17) / (2 * (80 / 23)^2)) - (- (277 / 40) / (10 / 3) - (- 17) / (2 * (10 / 3)^2))$$
by (rule piece-rat) (simp-all add: inner-piece-4)
 have s4: interval-lebesgue-integrable lborel (ereal 3) (ereal (80 / 23)) (λx. R-r x / x^3) ∧ interval-lebesgue-integral lborel (ereal 3) (ereal (80 / 23)) (λx. R-r x / x^3) =

$$(7159 / 89600)$$
using chain[OF - - s3 r4] **by** (simp add: power2-eq-square power3-eq-cube)
 have r5: interval-lebesgue-integrable lborel (ereal (80 / 23)) (ereal (7 / 2)) (λx. R-r x / x^3) ∧ interval-lebesgue-integral lborel (ereal (80 / 23)) (ereal (7 / 2)) (λx. R-r x / x^3) = (- (231 / 40) / (7 / 2) - (- 13) / (2 * (7 / 2)^2)) - (- (231 / 40) / (80 / 23) - (- 13) / (2 * (80 / 23)^2))
by (rule piece-rat) (simp-all add: inner-piece-5)
 have s5: interval-lebesgue-integrable lborel (ereal 3) (ereal (7 / 2)) (λx. R-r x / x^3) ∧ interval-lebesgue-integral lborel (ereal 3) (ereal (7 / 2)) (λx. R-r x / x^3) =

$$(6551 / 78400)$$
using chain[OF - - s4 r5] **by** (simp add: power2-eq-square power3-eq-cube)
 have r6: interval-lebesgue-integrable lborel (ereal (7 / 2)) (ereal (160 / 43)) (λx. R-r x / x^3) ∧ interval-lebesgue-integral lborel (ereal (7 / 2)) (ereal (160 / 43)) (λx. R-r x / x^3) = (- (271 / 40) / (160 / 43) - (- 33 / 2) / (2 * (160 / 43)^2)) - (- (271 / 40) / (7 / 2) - (- 33 / 2) / (2 * (7 / 2)^2))
by (rule piece-rat) (simp-all add: inner-piece-6)
 have s6: interval-lebesgue-integrable lborel (ereal 3) (ereal (160 / 43)) (λx. R-r x / x^3) ∧ interval-lebesgue-integral lborel (ereal 3) (ereal (160 / 43)) (λx. R-r x / x^3) =

$$(17331 / 143360)$$
using chain[OF - - s5 r6] **by** (simp add: power2-eq-square power3-eq-cube)
 have r7: interval-lebesgue-integrable lborel (ereal (160 / 43)) (ereal (140 / 37)) (λx. R-r x / x^3) ∧ interval-lebesgue-integral lborel (ereal (160 / 43)) (ereal (140 / 37)) (λx. R-r x / x^3) = (- (71 / 20) / (140 / 37) - (- 9 / 2) / (2 * (140 / 37)^2)) - (- (71 / 20) / (160 / 43) - (- 9 / 2) / (2 * (160 / 43)^2))
by (rule piece-rat) (simp-all add: inner-piece-7)
 have s7: interval-lebesgue-integrable lborel (ereal 3) (ereal (140 / 37)) (λx. R-r x / x^3) ∧ interval-lebesgue-integral lborel (ereal 3) (ereal (140 / 37)) (λx. R-r x / x^3) =

$3) =$
 $(16481 / 125440)$
using chain[OF - - s6 r7] **by** (simp add: power2-eq-square power3-eq-cube)
have r8: interval-lebesgue-integrable lborel (ereal (140 / 37)) (ereal (90 / 23)) ($\lambda x. R\text{-}r\ x / x^3$) $\wedge$
interval-lebesgue-integral lborel (ereal (140 / 37)) (ereal (90 / 23)) ($\lambda x. R\text{-}r\ x /$
 x^3) =
 $(- (27 / 5) / (90 / 23) - (- 23 / 2) / (2 * (90 / 23)^2)) -$
 $(- (27 / 5) / (140 / 37) - (- 23 / 2) / (2 * (140 / 37)^2))$
by (rule piece-rat) (simp-all add: inner-piece-8)
have s8: interval-lebesgue-integrable lborel (ereal 3) (ereal (90 / 23)) ($\lambda x. R\text{-}r\ x /$
 x^3) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (90 / 23)) ($\lambda x. R\text{-}r\ x / x^3$)
 $=$
 $(1106299 / 7257600)$
using chain[OF - - s7 r8] **by** (simp add: power2-eq-square power3-eq-cube)
have r9: interval-lebesgue-integrable lborel (ereal (90 / 23)) (ereal 4) ($\lambda x. R\text{-}r\ x /$
 x^3) $\wedge$
interval-lebesgue-integral lborel (ereal (90 / 23)) (ereal 4) ($\lambda x. R\text{-}r\ x / x^3$)
 $=$
 $(- (17 / 4) / 4 - (- 7) / (2 * 4^2)) - (- (17 / 4) / (90 / 23) - (-$
 $7) / (2 * (90 / 23)^2))$
by (rule piece-rat) (simp-all add: inner-piece-9)
have s9: interval-lebesgue-integrable lborel (ereal 3) (ereal 4) ($\lambda x. R\text{-}r\ x /$
 x^3) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal 4) ($\lambda x. R\text{-}r\ x / x^3$) =
 $(26807 / 161280)$
using chain[OF - - s8 r9] **by** (simp add: power2-eq-square power3-eq-cube)
have r10: interval-lebesgue-integrable lborel (ereal 4) (ereal (180 / 43)) ($\lambda x. R\text{-}r\ x /$
 x^3) $\wedge$
interval-lebesgue-integral lborel (ereal 4) (ereal (180 / 43)) ($\lambda x. R\text{-}r\ x / x^3$)
 $=$
 $(- (17 / 5) / (180 / 43) - (- 11) / (2 * (180 / 43)^2)) - (- (17 / 5)$
 $/ 4 - (- 11) / (2 * 4^2))$
by (rule piece-rat) (simp-all add: inner-piece-10)
have s10: interval-lebesgue-integrable lborel (ereal 3) (ereal (180 / 43)) ($\lambda x. R\text{-}r\ x /$
 x^3) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (180 / 43)) ($\lambda x. R\text{-}r\ x / x^3$)
 $=$
 $(1263659 / 7257600)$
using chain[OF - - s9 r10] **by** (simp add: power2-eq-square power3-eq-cube)
have r11: interval-lebesgue-integrable lborel (ereal (180 / 43)) (ereal (160 / 43)) ($\lambda x. R\text{-}r\ x /$
 x^3) $\wedge$
interval-lebesgue-integral lborel (ereal (180 / 43)) (ereal (160 / 43)) ($\lambda x. R\text{-}r\ x / x^3$)

```

37)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (180 / 43)) (ereal (160 / 37)) (λx. R-r
x /
x ^ 3) = (- (17 / 5) / (160 / 37) - (- 11) / (2 * (160 / 37) ^ 2)) - (-
(17 / 5) / (180 /
43) - (- 11) / (2 * (180 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-11)
have s11: interval-lebesgue-integrable lborel (ereal 3) (ereal (160 / 37)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (160 / 37)) (λx. R-r x / x ^
3) =
(581689 / 3225600)
using chain[OF - - s10 r11] by (simp add: power2-eq-square power3-eq-cube)
have r12: interval-lebesgue-integrable lborel (ereal (160 / 37)) (ereal (100 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (160 / 37)) (ereal (100 / 23)) (λx. R-r
x /
x ^ 3) = (- (321 / 40) / (100 / 23) - (- 31) / (2 * (100 / 23) ^ 2)) -
(- (321 / 40) /
(160 / 37) - (- 31) / (2 * (160 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-12)
have s12: interval-lebesgue-integrable lborel (ereal 3) (ereal (100 / 23)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (100 / 23)) (λx. R-r x / x ^
3) =
(1828811 / 10080000)
using chain[OF - - s11 r12] by (simp add: power2-eq-square power3-eq-cube)
have r13: interval-lebesgue-integrable lborel (ereal (100 / 23)) (ereal (9 /
2)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (100 / 23)) (ereal (9 / 2)) (λx. R-r x /
x ^ 3) = (- (55 / 8) / (9 / 2) - (- 26) / (2 * (9 / 2) ^ 2)) - (- (55 /
8) / (100 / 23) -
(- 26) / (2 * (100 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-13)
have s13: interval-lebesgue-integrable lborel (ereal 3) (ereal (9 / 2)) (λx. R-r x
/
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (9 / 2)) (λx. R-r x / x ^ 3) =
(3432431 / 18144000)
using chain[OF - - s12 r13] by (simp add: power2-eq-square power3-eq-cube)
have r14: interval-lebesgue-integrable lborel (ereal (9 / 2)) (ereal (200 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (9 / 2)) (ereal (200 / 43)) (λx. R-r x /
x ^ 3) = (- (63 / 8) / (200 / 43) - (- 61 / 2) / (2 * (200 / 43) ^ 2))
- (- (63 / 8) / (9 /
2) - (- 61 / 2) / (2 * (9 / 2) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-14)

```



```

have s14: interval-lebesgue-integrable lborel (ereal 3) (ereal (200 / 43)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (200 / 43)) (λx. R-r x / x ^
3) =
  (332467 / 1680000)
using chain[OF - - s13 r14] by (simp add: power2-eq-square power3-eq-cube)
have r15: interval-lebesgue-integrable lborel (ereal (200 / 43)) (ereal (110 /
23)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (200 / 43)) (ereal (110 / 23)) (λx. R-r
x /
  x ^ 3) = (- (93 / 20) / (110 / 23) - (- 31 / 2) / (2 * (110 / 23) ^ 2))
- (- (93 / 20) /
  (200 / 43) - (- 31 / 2) / (2 * (200 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-15)
have s15: interval-lebesgue-integrable lborel (ereal 3) (ereal (110 / 23)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (110 / 23)) (λx. R-r x / x ^
3) =
  (16746439 / 81312000)
using chain[OF - - s14 r15] by (simp add: power2-eq-square power3-eq-cube)
have r16: interval-lebesgue-integrable lborel (ereal (110 / 23)) (ereal (180 /
37)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (110 / 23)) (ereal (180 / 37)) (λx. R-r
x /
  x ^ 3) = (- (7 / 2) / (180 / 37) - (- 10) / (2 * (180 / 37) ^ 2)) - (-
(7 / 2) / (110 /
  23) - (- 10) / (2 * (110 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-16)
have s16: interval-lebesgue-integrable lborel (ereal 3) (ereal (180 / 37)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (180 / 37)) (λx. R-r x / x ^
3) =
  (42111623 / 199584000)
using chain[OF - - s15 r16] by (simp add: power2-eq-square power3-eq-cube)
have r17: interval-lebesgue-integrable lborel (ereal (180 / 37)) (ereal 5) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (180 / 37)) (ereal 5) (λx. R-r x / x ^
3) =
  (- (107 / 20) / 5 - (- 19) / (2 * 5 ^ 2)) - (- (107 / 20) / (180 / 37)
- (- 19) / (2 *
  (180 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-17)
have s17: interval-lebesgue-integrable lborel (ereal 3) (ereal 5) (λx. R-r x /
x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal 5) (λx. R-r x / x ^ 3) =

```

```

(4863527 / 22176000)
using chain[OF - - s16 r17] by (simp add: power2-eq-square power3-eq-cube)
have r18: interval-lebesgue-integrable lborel (ereal 5) (ereal (220 / 43)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 5) (ereal (220 / 43)) (λx. R-r x / x ^
3) =
  (− (9 / 2) / (220 / 43) − (− 24) / (2 * (220 / 43) ^ 2)) − (− (9 / 2) /
5 − (− 24) / (2 *
  5 ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-18)
have s18: interval-lebesgue-integrable lborel (ereal 3) (ereal (220 / 43)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (220 / 43)) (λx. R-r x / x ^
3) =
  (53226637 / 243936000)
using chain[OF - - s17 r18] by (simp add: power2-eq-square power3-eq-cube)
have r19: interval-lebesgue-integrable lborel (ereal (220 / 43)) (ereal (120 /
23)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (220 / 43)) (ereal (120 / 23)) (λx. R-r
x /
  x ^ 3) = (− (9 / 2) / (120 / 23) − (− 24) / (2 * (120 / 23) ^ 2)) − (−
(9 / 2) / (220 /
  43) − (− 24) / (2 * (220 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-19)
have s19: interval-lebesgue-integrable lborel (ereal 3) (ereal (120 / 23)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (120 / 23)) (λx. R-r x / x ^
3) =
  (4826567 / 22176000)
using chain[OF - - s18 r19] by (simp add: power2-eq-square power3-eq-cube)
have r20: interval-lebesgue-integrable lborel (ereal (120 / 23)) (ereal (200 /
37)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (120 / 23)) (ereal (200 / 37)) (λx. R-r
x /
  x ^ 3) = (− (67 / 20) / (200 / 37) − (− 18) / (2 * (200 / 37) ^ 2)) −
(− (67 / 20) / (120 /
  23) − (− 18) / (2 * (120 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-20)
have s20: interval-lebesgue-integrable lborel (ereal 3) (ereal (200 / 37)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (200 / 37)) (λx. R-r x / x ^
3) =
  (24103267 / 110880000)
using chain[OF - - s19 r20] by (simp add: power2-eq-square power3-eq-cube)
have r21: interval-lebesgue-integrable lborel (ereal (200 / 37)) (ereal (11 /

```

```

2)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (200 / 37)) (ereal (11 / 2)) (λx. R-r x
/
x ^ 3) = (- (319 / 40) / (11 / 2) - (- 43) / (2 * (11 / 2) ^ 2)) - (-
(319 / 40) / (200 /
37) - (- 43) / (2 * (200 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-21)
have s21: interval-lebesgue-integrable lborel (ereal 3) (ereal (11 / 2)) (λx. R-r x
/
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (11 / 2)) (λx. R-r x / x ^ 3)
=
(53095807 / 243936000)
using chain[OF - - s20 r21] by (simp add: power2-eq-square power3-eq-cube)
have r22: interval-lebesgue-integrable lborel (ereal (11 / 2)) (ereal (240 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (11 / 2)) (ereal (240 / 43)) (λx. R-r x
/
x ^ 3) = (- (359 / 40) / (240 / 43) - (- 97 / 2) / (2 * (240 / 43) ^
2)) - (- (359 / 40) /
(11 / 2) - (- 97 / 2) / (2 * (11 / 2) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-22)
have s22: interval-lebesgue-integrable lborel (ereal 3) (ereal (240 / 43)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (240 / 43)) (λx. R-r x / x ^
3) =
(6453191 / 29568000)
using chain[OF - - s21 r22] by (simp add: power2-eq-square power3-eq-cube)
have r23: interval-lebesgue-integrable lborel (ereal (240 / 43)) (ereal (130 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (240 / 43)) (ereal (130 / 23)) (λx. R-r
x /
x ^ 3) = (- (23 / 4) / (130 / 23) - (- 61 / 2) / (2 * (130 / 23) ^ 2))
- (- (23 / 4) /
(240 / 43) - (- 61 / 2) / (2 * (240 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-23)
have s23: interval-lebesgue-integrable lborel (ereal 3) (ereal (130 / 23)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (130 / 23)) (λx. R-r x / x ^
3) =
(205157387 / 936936000)
using chain[OF - - s22 r23] by (simp add: power2-eq-square power3-eq-cube)
have r24: interval-lebesgue-integrable lborel (ereal (130 / 23)) (ereal (220 /
37)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (130 / 23)) (ereal (220 / 37)) (λx. R-r
x /
x ^ 3) = (- (23 / 5) / (220 / 37) - (- 24) / (2 * (220 / 37) ^ 2)) - (-

```

```

(23 / 5) / (130 /
  23) - (- 24) / (2 * (130 / 23) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-24)
  have s24: interval-lebesgue-integrable lborel (ereal 3) (ereal (220 / 37)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (220 / 37)) (λx. R-r x / x ^
3) =
      (176773789 / 792792000)
  using chain[OF - - s23 r24] by (simp add: power2-eq-square power3-eq-cube)
  have r25: interval-lebesgue-integrable lborel (ereal (220 / 37)) (ereal 6) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (220 / 37)) (ereal 6) (λx. R-r x / x ^
3) =
      (- (129 / 20) / 6 - (- 35) / (2 * 6 ^ 2)) - (- (129 / 20) / (220 / 37))
- (- 35) / (2 *
  (220 / 37) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-25)
  have s25: interval-lebesgue-integrable lborel (ereal 3) (ereal 6) (λx. R-r x /
x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal 6) (λx. R-r x / x ^ 3) =
      (488933 / 2184000)
  using chain[OF - - s24 r25] by (simp add: power2-eq-square power3-eq-cube)
  have r26: interval-lebesgue-integrable lborel (ereal 6) (ereal (260 / 43)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 6) (ereal (260 / 43)) (λx. R-r x / x ^
3) =
      (- (28 / 5) / (260 / 43) - (- 41) / (2 * (260 / 43) ^ 2)) - (- (28 / 5)
/ 6 - (- 41) / (2 *
  6 ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-26)
  have s26: interval-lebesgue-integrable lborel (ereal 3) (ereal (260 / 43)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (260 / 43)) (λx. R-r x / x ^
3) =
      (18936577 / 85176000)
  using chain[OF - - s25 r26] by (simp add: power2-eq-square power3-eq-cube)
  have r27: interval-lebesgue-integrable lborel (ereal (260 / 43)) (ereal (140 /
23)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (260 / 43)) (ereal (140 / 23)) (λx. R-r
x /
  x ^ 3) = (- (28 / 5) / (140 / 23) - (- 41) / (2 * (140 / 23) ^ 2)) - (-
(28 / 5) / (260 /
  43) - (- 41) / (2 * (260 / 43) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-27)
  have s27: interval-lebesgue-integrable lborel (ereal 3) (ereal (140 / 23)) (λx. R-r

```

```

x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (140 / 23)) (λx. R-r x / x ^
3) =
  (10138243 / 45864000)
  using chain[OF - - s26 r27] by (simp add: power2-eq-square power3-eq-cube)
  have r28: interval-lebesgue-integrable lborel (ereal (140 / 23)) (ereal (240 /
37)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (140 / 23)) (ereal (240 / 37)) (λx. R-r
x /
  x ^ 3) = (- (89 / 20) / (240 / 37) - (- 34) / (2 * (240 / 37) ^ 2)) -
(- (89 / 20) / (140 /
23) - (- 34) / (2 * (140 / 23) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-28)
  have s28: interval-lebesgue-integrable lborel (ereal 3) (ereal (240 / 37)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (240 / 37)) (λx. R-r x / x ^
3) =
  (5537711 / 26208000)
  using chain[OF - - s27 r28] by (simp add: power2-eq-square power3-eq-cube)
  have r29: interval-lebesgue-integrable lborel (ereal (240 / 37)) (ereal (13 /
2)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (240 / 37)) (ereal (13 / 2)) (λx. R-r x
/
  x ^ 3) = (- (363 / 40) / (13 / 2) - (- 64) / (2 * (13 / 2) ^ 2)) - (-
(363 / 40) / (240 /
37) - (- 64) / (2 * (240 / 37) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-29)
  have s29: interval-lebesgue-integrable lborel (ereal 3) (ereal (13 / 2)) (λx. R-r x
/
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (13 / 2)) (λx. R-r x / x ^ 3)
=
  (71904913 / 340704000)
  using chain[OF - - s28 r29] by (simp add: power2-eq-square power3-eq-cube)
  have r30: interval-lebesgue-integrable lborel (ereal (13 / 2)) (ereal (280 /
43)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (13 / 2)) (ereal (280 / 43)) (λx. R-r x
/
  x ^ 3) = (- (403 / 40) / (280 / 43) - (- 141 / 2) / (2 * (280 / 43) ^
2)) - (- (403 / 40) /
(13 / 2) - (- 141 / 2) / (2 * (13 / 2) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-30)
  have s30: interval-lebesgue-integrable lborel (ereal 3) (ereal (280 / 43)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (280 / 43)) (λx. R-r x / x ^
3) =

```

```

(9669913 / 45864000)
using chain[OF - - s29 r30] by (simp add: power2-eq-square power3-eq-cube)
have r31: interval-lebesgue-integrable lborel (ereal (280 / 43)) (ereal (150 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (280 / 43)) (ereal (150 / 23)) (λx. R-r
x /
x ^ 3) = (- (137 / 20) / (150 / 23) - (- 99 / 2) / (2 * (150 / 23) ^
2)) - (- (137 / 20) /
(280 / 43) - (- 99 / 2) / (2 * (280 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-31)
have s31: interval-lebesgue-integrable lborel (ereal 3) (ereal (150 / 23)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (150 / 23)) (λx. R-r x / x ^
3) =
(27605051 / 131040000)
using chain[OF - - s30 r31] by (simp add: power2-eq-square power3-eq-cube)
have r32: interval-lebesgue-integrable lborel (ereal (150 / 23)) (ereal (20 /
3)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (150 / 23)) (ereal (20 / 3)) (λx. R-r x
/
x ^ 3) = (- (57 / 10) / (20 / 3) - (- 42) / (2 * (20 / 3) ^ 2)) - (-
(57 / 10) / (150 /
23) - (- 42) / (2 * (150 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-32)
have s32: interval-lebesgue-integrable lborel (ereal 3) (ereal (20 / 3)) (λx. R-r x
/
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (20 / 3)) (λx. R-r x / x ^ 3)
=
(5462479 / 26208000)
using chain[OF - - s31 r32] by (simp add: power2-eq-square power3-eq-cube)
have r33: interval-lebesgue-integrable lborel (ereal (20 / 3)) (ereal (160 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (20 / 3)) (ereal (160 / 23)) (λx. R-r x
/
x ^ 3) = (- (69 / 10) / (160 / 23) - (- 50) / (2 * (160 / 23) ^ 2)) -
(- (69 / 10) / (20 /
3) - (- 50) / (2 * (20 / 3) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-33)
have s33: interval-lebesgue-integrable lborel (ereal 3) (ereal (160 / 23)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (160 / 23)) (λx. R-r x / x ^
3) =
(21559171 / 104832000)
using chain[OF - - s32 r33] by (simp add: power2-eq-square power3-eq-cube)
have r34: interval-lebesgue-integrable lborel (ereal (160 / 23)) (ereal (300 /
43)) (λx. R-r x / x ^ 3) ∧

```

```

      interval-lebesgue-integral lborel (ereal (160 / 23)) (ereal (300 / 43)) (λx. R-r
x /
      x ^ 3) = (- (23 / 4) / (300 / 43) - (- 42) / (2 * (300 / 43) ^ 2)) - (-
(23 / 4) / (160 /
      23) - (- 42) / (2 * (160 / 23) ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-34)
    have s34: interval-lebesgue-integrable lborel (ereal 3) (ereal (300 / 43)) (λx. R-r
x /
      x ^ 3) ∧
      interval-lebesgue-integral lborel (ereal 3) (ereal (300 / 43)) (λx. R-r x / x ^
3) =
      (1683359 / 8190000)
    using chain[OF - - s33 r34] by (simp add: power2-eq-square power3-eq-cube)
    have r35: interval-lebesgue-integrable lborel (ereal (300 / 43)) (ereal 7) (λx. R-r
x /
      x ^ 3) ∧
      interval-lebesgue-integral lborel (ereal (300 / 43)) (ereal 7) (λx. R-r x / x ^
3) =
      (- (23 / 4) / 7 - (- 42) / (2 * 7 ^ 2)) - (- (23 / 4) / (300 / 43) -
(- 42) / (2 * (300 /
      43) ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-35)
    have s35: interval-lebesgue-integrable lborel (ereal 3) (ereal 7) (λx. R-r x /
x ^ 3) ∧
      interval-lebesgue-integral lborel (ereal 3) (ereal 7) (λx. R-r x / x ^ 3) =
      (48067 / 234000)
    using chain[OF - - s34 r35] by (simp add: power2-eq-square power3-eq-cube)
    have r36: interval-lebesgue-integrable lborel (ereal 7) (ereal (260 / 37)) (λx. R-r
x /
      x ^ 3) ∧
      interval-lebesgue-integral lborel (ereal 7) (ereal (260 / 37)) (λx. R-r x / x ^
3) =
      (- (49 / 10) / (260 / 37) - (- 49) / (2 * (260 / 37) ^ 2)) - (- (49 /
10) / 7 - (- 49) /
      (2 * 7 ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-36)
    have s36: interval-lebesgue-integrable lborel (ereal 3) (ereal (260 / 37)) (λx. R-r
x /
      x ^ 3) ∧
      interval-lebesgue-integral lborel (ereal 3) (ereal (260 / 37)) (λx. R-r x / x ^
3) =
      (1242767 / 6084000)
    using chain[OF - - s35 r36] by (simp add: power2-eq-square power3-eq-cube)
    have r37: interval-lebesgue-integrable lborel (ereal (260 / 37)) (ereal (170 /
23)) (λx. R-r x / x ^ 3) ∧
      interval-lebesgue-integral lborel (ereal (260 / 37)) (ereal (170 / 23)) (λx. R-r
x /
      x ^ 3) = (- (27 / 4) / (170 / 23) - (- 62) / (2 * (170 / 23) ^ 2)) - (-
(27 / 4) / (260 /

```

$37) - (-62) / (2 * (260 / 37) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-37)
have s37: interval-lebesgue-integrable lborel (ereal 3) (ereal (170 / 23)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (170 / 23)) ($\lambda x. R-r x / x ^$
 $3) =$
(25867391 / 135252000)
using chain[OF - - s36 r37] **by** (simp add: power2-eq-square power3-eq-cube)
have r38: interval-lebesgue-integrable lborel (ereal (170 / 23)) (ereal (320 /
43)) ($\lambda x. R-r x / x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal (170 / 23)) (ereal (320 / 43)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) = (- (28 / 5) / (320 / 43) - (- 107 / 2) / (2 * (320 / 43) ^ 2))$
 $- (- (28 / 5) /$
 $(170 / 23) - (- 107 / 2) / (2 * (170 / 23) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-38)
have s38: interval-lebesgue-integrable lborel (ereal 3) (ereal (320 / 43)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (320 / 43)) ($\lambda x. R-r x / x ^$
 $3) =$
(773023451 / 4073472000)
using chain[OF - - s37 r38] **by** (simp add: power2-eq-square power3-eq-cube)
have r39: interval-lebesgue-integrable lborel (ereal (320 / 43)) (ereal (15 /
2)) ($\lambda x. R-r x / x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal (320 / 43)) (ereal (15 / 2)) ($\lambda x. R-r x$
 $/$
 $x ^ 3) = (- (19 / 8) / (15 / 2) - (- 59 / 2) / (2 * (15 / 2) ^ 2)) - (-$
 $(19 / 8) / (320 /$
43) - (- 59 / 2) / (2 * (320 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-39)
have s39: interval-lebesgue-integrable lborel (ereal 3) (ereal (15 / 2)) ($\lambda x. R-r x$
 $/$
 $x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (15 / 2)) ($\lambda x. R-r x / x ^ 3$
 $) =$
(47896621 / 254592000)
using chain[OF - - s38 r39] **by** (simp add: power2-eq-square power3-eq-cube)
have r40: interval-lebesgue-integrable lborel (ereal (15 / 2)) (ereal (280 /
37)) ($\lambda x. R-r x / x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal (15 / 2)) (ereal (280 / 37)) ($\lambda x. R-r x$
 $/$
 $x ^ 3) = (- (27 / 8) / (280 / 37) - (- 37) / (2 * (280 / 37) ^ 2)) - (-$
 $(27 / 8) / (15 /$
2) - (- 37) / (2 * (15 / 2) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-40)
have s40: interval-lebesgue-integrable lborel (ereal 3) (ereal (280 / 37)) ($\lambda x. R-r$
 $x /$


```

x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (280 / 37)) (λx. R-r x / x ^
3) =
(2324118389 / 12475008000)
using chain[OF - - s39 r40] by (simp add: power2-eq-square power3-eq-cube)
have r41: interval-lebesgue-integrable lborel (ereal (280 / 37)) (ereal (180 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (280 / 37)) (ereal (180 / 23)) (λx. R-r
x /
x ^ 3) = (- 8 / (180 / 23) - (- 72) / (2 * (180 / 23) ^ 2)) - (- 8 /
(280 / 37) - (- 72) /
(2 * (280 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-41)
have s41: interval-lebesgue-integrable lborel (ereal 3) (ereal (180 / 23)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (180 / 23)) (λx. R-r x / x ^
3) =
(321459427 / 1782144000)
using chain[OF - - s40 r41] by (simp add: power2-eq-square power3-eq-cube)
have r42: interval-lebesgue-integrable lborel (ereal (180 / 23)) (ereal (340 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (180 / 23)) (ereal (340 / 43)) (λx. R-r
x /
x ^ 3) = (- (137 / 20) / (340 / 43) - (- 63) / (2 * (340 / 43) ^ 2)) -
(- (137 / 20) /
(180 / 23) - (- 63) / (2 * (180 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-42)
have s42: interval-lebesgue-integrable lborel (ereal 3) (ereal (340 / 43)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (340 / 43)) (λx. R-r x / x ^
3) =
(1806305713 / 10098816000)
using chain[OF - - s41 r42] by (simp add: power2-eq-square power3-eq-cube)
have r43: interval-lebesgue-integrable lborel (ereal (340 / 43)) (ereal 8) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (340 / 43)) (ereal 8) (λx. R-r x / x ^
3) =
(- (137 / 20) / 8 - (- 63) / (2 * 8 ^ 2)) - (- (137 / 20) / (340 / 43)
- (- 63) / (2 *
(340 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-43)
have s43: interval-lebesgue-integrable lborel (ereal 3) (ereal 8) (λx. R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal 8) (λx. R-r x / x ^ 3) =
(105317369 / 594048000)
using chain[OF - - s42 r43] by (simp add: power2-eq-square power3-eq-cube)

```

```

have r44: interval-lebesgue-integrable lborel (ereal 8) (ereal (300 / 37)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 8) (ereal (300 / 37)) (λx. R-r x / x ^
3) =
  (- 6 / (300 / 37) - (- 71) / (2 * (300 / 37) ^ 2)) - (- 6 / 8 - (- 71)
/ (2 * 8 ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-44)
have s44: interval-lebesgue-integrable lborel (ereal 3) (ereal (300 / 37)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (300 / 37)) (λx. R-r x / x ^
3) =
  (1537942031 / 8910720000)
  using chain[OF - - s43 r44] by (simp add: power2-eq-square power3-eq-cube)
have r45: interval-lebesgue-integrable lborel (ereal (300 / 37)) (ereal (190 /
23)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (300 / 37)) (ereal (190 / 23)) (λx. R-r
x /
  x ^ 3) = (- (157 / 20) / (190 / 23) - (- 86) / (2 * (190 / 23) ^ 2)) -
(- (157 / 20) /
  (300 / 37) - (- 86) / (2 * (300 / 37) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-45)
have s45: interval-lebesgue-integrable lborel (ereal 3) (ereal (190 / 23)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (190 / 23)) (λx. R-r x / x ^
3) =
  (2380854663 / 14296755200)
  using chain[OF - - s44 r45] by (simp add: power2-eq-square power3-eq-cube)
have r46: interval-lebesgue-integrable lborel (ereal (190 / 23)) (ereal (360 /
43)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (190 / 23)) (ereal (360 / 43)) (λx. R-r
x /
  x ^ 3) = (- (67 / 10) / (360 / 43) - (- 153 / 2) / (2 * (360 / 43) ^
2)) - (- (67 / 10) /
  (190 / 23) - (- 153 / 2) / (2 * (190 / 23) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-46)
have s46: interval-lebesgue-integrable lborel (ereal 3) (ereal (360 / 43)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (360 / 43)) (λx. R-r x / x ^
3) =
  (1100556809 / 6772147200)
  using chain[OF - - s45 r46] by (simp add: power2-eq-square power3-eq-cube)
have r47: interval-lebesgue-integrable lborel (ereal (360 / 43)) (ereal (17 /
2)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (360 / 43)) (ereal (17 / 2)) (λx. R-r x
/

```

$$x^{\wedge} 3) = (- (139 / 40) / (17 / 2) - (- 99 / 2) / (2 * (17 / 2)^{\wedge} 2)) -$$

$$(- (139 / 40) /$$

$$(360 / 43) - (- 99 / 2) / (2 * (360 / 43)^{\wedge} 2))$$
by (rule piece-rat) (simp-all add: inner-piece-47)
 have s47: interval-lebesgue-integrable lborel (ereal 3) (ereal (17 / 2)) ($\lambda x. R\text{-}r\ x$ /
 /

$$x^{\wedge} 3) \wedge$$
 interval-lebesgue-integral lborel (ereal 3) (ereal (17 / 2)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$)
 =
 (6071429023 / 38375500800)
 using chain[OF - - s46 r47] **by** (simp add: power2-eq-square power3-eq-cube)
 have r48: interval-lebesgue-integrable lborel (ereal (17 / 2)) (ereal (320 /
 37)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$) $\wedge$
 interval-lebesgue-integral lborel (ereal (17 / 2)) (ereal (320 / 37)) ($\lambda x. R\text{-}r\ x$ /
 /

$$x^{\wedge} 3) = (- (179 / 40) / (320 / 37) - (- 58) / (2 * (320 / 37)^{\wedge} 2)) -$$

$$(- (179 / 40) / (17 /$$

$$2) - (- 58) / (2 * (17 / 2)^{\wedge} 2))$$
by (rule piece-rat) (simp-all add: inner-piece-48)
 have s48: interval-lebesgue-integrable lborel (ereal 3) (ereal (320 / 37)) ($\lambda x. R\text{-}r\ x$ /
 x /

$$x^{\wedge} 3) \wedge$$
 interval-lebesgue-integral lborel (ereal 3) (ereal (320 / 37)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$)
 =
 (1386762179 / 9029529600)
 using chain[OF - - s47 r48] **by** (simp add: power2-eq-square power3-eq-cube)
 have r49: interval-lebesgue-integrable lborel (ereal (320 / 37)) (ereal (200 /
 23)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$) $\wedge$
 interval-lebesgue-integral lborel (ereal (320 / 37)) (ereal (200 / 23)) ($\lambda x. R\text{-}r\ x$ /
 x /

$$x^{\wedge} 3) = (- (91 / 10) / (200 / 23) - (- 98) / (2 * (200 / 23)^{\wedge} 2)) -$$

$$(- (91 / 10) / (320 /$$

$$37) - (- 98) / (2 * (320 / 37)^{\wedge} 2))$$
by (rule piece-rat) (simp-all add: inner-piece-49)
 have s49: interval-lebesgue-integrable lborel (ereal 3) (ereal (200 / 23)) ($\lambda x. R\text{-}r\ x$ /
 x /

$$x^{\wedge} 3) \wedge$$
 interval-lebesgue-integral lborel (ereal 3) (ereal (200 / 23)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$)
 =
 (4294822027 / 28217280000)
 using chain[OF - - s48 r49] **by** (simp add: power2-eq-square power3-eq-cube)
 have r50: interval-lebesgue-integrable lborel (ereal (200 / 23)) (ereal (380 /
 43)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$) $\wedge$
 interval-lebesgue-integral lborel (ereal (200 / 23)) (ereal (380 / 43)) ($\lambda x. R\text{-}r\ x$ /
 x /

$$x^{\wedge} 3) = (- (159 / 20) / (380 / 43) - (- 88) / (2 * (380 / 43)^{\wedge} 2)) -$$

$$(- (159 / 20) /$$

$$(200 / 23) - (- 88) / (2 * (200 / 23)^{\wedge} 2))$$
by (rule piece-rat) (simp-all add: inner-piece-50)

```

have s50: interval-lebesgue-integrable lborel (ereal 3) (ereal (380 / 43)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (380 / 43)) (λx. R-r x / x ^
3) =
  (15907712813 / 107225664000)
using chain[OF - - s49 r50] by (simp add: power2-eq-square power3-eq-cube)
have r51: interval-lebesgue-integrable lborel (ereal (380 / 43)) (ereal 9) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (380 / 43)) (ereal 9) (λx. R-r x / x ^
3) =
  (-(159 / 20) / 9 - (- 88) / (2 * 9 ^ 2)) - (-(159 / 20) / (380 / 43)
- (- 88) / (2 *
  (380 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-51)
have s51: interval-lebesgue-integrable lborel (ereal 3) (ereal 9) (λx. R-r x /
x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal 9) (λx. R-r x / x ^ 3) =
  (22007571589 / 152373312000)
using chain[OF - - s50 r51] by (simp add: power2-eq-square power3-eq-cube)
have r52: interval-lebesgue-integrable lborel (ereal 9) (ereal (210 / 23)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 9) (ereal (210 / 23)) (λx. R-r x / x ^
3) =
  (-(71 / 10) / (210 / 23) - (- 97) / (2 * (210 / 23) ^ 2)) - (-(71 /
10) / 9 - (- 97) /
  (2 * 9 ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-52)
have s52: interval-lebesgue-integrable lborel (ereal 3) (ereal (210 / 23)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (210 / 23)) (λx. R-r x / x ^
3) =
  (16439636107 / 118512576000)
using chain[OF - - s51 r52] by (simp add: power2-eq-square power3-eq-cube)
have r53: interval-lebesgue-integrable lborel (ereal (210 / 23)) (ereal (340 /
37)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (210 / 23)) (ereal (340 / 37)) (λx. R-r
x /
  x ^ 3) = (-(119 / 20) / (340 / 37) - (- 173 / 2) / (2 * (340 / 37) ^
2)) - (-(119 / 20) /
  (210 / 23) - (- 173 / 2) / (2 * (210 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-53)
have s53: interval-lebesgue-integrable lborel (ereal 3) (ereal (340 / 37)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (340 / 37)) (λx. R-r x / x ^

```

$3) =$
 $(622550859 / 4568512000)$
using *chain*[*OF* - - *s52 r53*] **by** (*simp add: power2-eq-square power3-eq-cube*)
have *r54: interval-lebesgue-integrable lborel* (*ereal* ($340 / 37$)) (*ereal* ($400 / 43$)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (*ereal* ($340 / 37$)) (*ereal* ($400 / 43$)) ($\lambda x. R\text{-}r\ x /$
 $x^{\wedge} 3$) = $(- (39 / 5) / (400 / 43) - (- 207 / 2) / (2 * (400 / 43)^{\wedge} 2))$
 $- (- (39 / 5) /$
 $(340 / 37) - (- 207 / 2) / (2 * (340 / 37)^{\wedge} 2))$
by (*rule piece-rat*) (*simp-all add: inner-piece-54*)
have *s54: interval-lebesgue-integrable lborel* (*ereal* 3) (*ereal* ($400 / 43$)) ($\lambda x. R\text{-}r\ x /$
 $x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (*ereal* 3) (*ereal* ($400 / 43$)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$)
 $3) =$
 $(354131967 / 2687360000)$
using *chain*[*OF* - - *s53 r54*] **by** (*simp add: power2-eq-square power3-eq-cube*)
have *r55: interval-lebesgue-integrable lborel* (*ereal* ($400 / 43$)) (*ereal* ($19 / 2$)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (*ereal* ($400 / 43$)) (*ereal* ($19 / 2$)) ($\lambda x. R\text{-}r\ x /$
 $x^{\wedge} 3$) = $(- (183 / 40) / (19 / 2) - (- 147 / 2) / (2 * (19 / 2)^{\wedge} 2)) -$
 $(- (183 / 40) /$
 $(400 / 43) - (- 147 / 2) / (2 * (400 / 43)^{\wedge} 2))$
by (*rule piece-rat*) (*simp-all add: inner-piece-55*)
have *s55: interval-lebesgue-integrable lborel* (*ereal* 3) (*ereal* ($19 / 2$)) ($\lambda x. R\text{-}r\ x /$
 $x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (*ereal* 3) (*ereal* ($19 / 2$)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$)
 $=$
 $(635799579 / 5105984000)$
using *chain*[*OF* - - *s54 r55*] **by** (*simp add: power2-eq-square power3-eq-cube*)
have *r56: interval-lebesgue-integrable lborel* (*ereal* ($19 / 2$)) (*ereal* ($220 / 23$)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (*ereal* ($19 / 2$)) (*ereal* ($220 / 23$)) ($\lambda x. R\text{-}r\ x /$
 $x^{\wedge} 3$) = $(- (223 / 40) / (220 / 23) - (- 83) / (2 * (220 / 23)^{\wedge} 2)) -$
 $(- (223 / 40) / (19 / 2) -$
 $(- 83) / (2 * (19 / 2)^{\wedge} 2))$
by (*rule piece-rat*) (*simp-all add: inner-piece-56*)
have *s56: interval-lebesgue-integrable lborel* (*ereal* 3) (*ereal* ($220 / 23$)) ($\lambda x. R\text{-}r\ x /$
 $x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (*ereal* 3) (*ereal* ($220 / 23$)) ($\lambda x. R\text{-}r\ x / x^{\wedge} 3$)
 $3) =$
 $(3975944721 / 32517056000)$
using *chain*[*OF* - - *s55 r56*] **by** (*simp add: power2-eq-square power3-eq-cube*)
have *r57: interval-lebesgue-integrable lborel* (*ereal* ($220 / 23$)) (*ereal* ($360 /$

```

37)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (220 / 23)) (ereal (360 / 37)) (λx. R-r
x /
x ^ 3) = (- (177 / 40) / (360 / 37) - (- 72) / (2 * (360 / 37) ^ 2)) -
(- (177 / 40) /
(220 / 23) - (- 72) / (2 * (220 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-57)
have s57: interval-lebesgue-integrable lborel (ereal 3) (ereal (360 / 37)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (360 / 37)) (λx. R-r x / x ^
3) =
(3110142259 / 26604864000)
using chain[OF - - s56 r57] by (simp add: power2-eq-square power3-eq-cube)
have r58: interval-lebesgue-integrable lborel (ereal (360 / 37)) (ereal (420 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (360 / 37)) (ereal (420 / 43)) (λx. R-r
x /
x ^ 3) = (- (181 / 20) / (420 / 43) - (- 117) / (2 * (420 / 43) ^ 2))
- (- (181 / 20) /
(360 / 37) - (- 117) / (2 * (360 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-58)
have s58: interval-lebesgue-integrable lborel (ereal 3) (ereal (420 / 43)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (420 / 43)) (λx. R-r x / x ^
3) =
(150869962351 / 1303638336000)
using chain[OF - - s57 r58] by (simp add: power2-eq-square power3-eq-cube)
have r59: interval-lebesgue-integrable lborel (ereal (420 / 43)) (ereal 10) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (420 / 43)) (ereal 10) (λx. R-r x / x ^
3) =
(- (181 / 20) / 10 - (- 117) / (2 * 10 ^ 2)) - (- (181 / 20) / (420 /
43) - (- 117) / (2 *
(420 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-59)
have s59: interval-lebesgue-integrable lborel (ereal 3) (ereal 10) (λx. R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal 10) (λx. R-r x / x ^ 3) =
(2902291759 / 26604864000)
using chain[OF - - s58 r59] by (simp add: power2-eq-square power3-eq-cube)
have r60: interval-lebesgue-integrable lborel (ereal 10) (ereal (440 / 43)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 10) (ereal (440 / 43)) (λx. R-r x / x ^
3) =
(- (69 / 10) / (440 / 43) - (- 114) / (2 * (440 / 43) ^ 2)) - (- (69 /

```

```

10) / 10 - (- 114) /
(2 * 10 ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-60)
  have s60: interval-lebesgue-integrable lborel (ereal 3) (ereal (440 / 43)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (440 / 43)) (λx. R-r x / x ^
3) =
      (29018325629 / 292653504000)
  using chain[OF - - s59 r60] by (simp add: power2-eq-square power3-eq-cube)
  have r61: interval-lebesgue-integrable lborel (ereal (440 / 43)) (ereal (380 /
37)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (440 / 43)) (ereal (380 / 37)) (λx. R-r
x /
  x ^ 3) = (- (147 / 40) / (380 / 37) - (- 81) / (2 * (380 / 37) ^ 2)) -
(- (147 / 40) /
  (440 / 43) - (- 81) / (2 * (440 / 43) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-61)
  have s61: interval-lebesgue-integrable lborel (ereal 3) (ereal (380 / 37)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (380 / 37)) (λx. R-r x / x ^
3) =
      (49355911441 / 505492416000)
  using chain[OF - - s60 r61] by (simp add: power2-eq-square power3-eq-cube)
  have r62: interval-lebesgue-integrable lborel (ereal (380 / 37)) (ereal (240 /
23)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (380 / 37)) (ereal (240 / 23)) (λx. R-r
x /
  x ^ 3) = (- (221 / 40) / (240 / 23) - (- 100) / (2 * (240 / 23) ^ 2))
- (- (221 / 40) /
  (380 / 37) - (- 100) / (2 * (380 / 37) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-62)
  have s62: interval-lebesgue-integrable lborel (ereal 3) (ereal (240 / 23)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (240 / 23)) (λx. R-r x / x ^
3) =
      (809599753 / 8868288000)
  using chain[OF - - s61 r62] by (simp add: power2-eq-square power3-eq-cube)
  have r63: interval-lebesgue-integrable lborel (ereal (240 / 23)) (ereal (21 /
2)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (240 / 23)) (ereal (21 / 2)) (λx. R-r x
/
  x ^ 3) = (- (35 / 8) / (21 / 2) - (- 88) / (2 * (21 / 2) ^ 2)) - (- (35
/ 8) / (240 / 23) -
  (- 88) / (2 * (240 / 23) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-63)
  have s63: interval-lebesgue-integrable lborel (ereal 3) (ereal (21 / 2)) (λx. R-r x

```

```

/
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (21 / 2)) (λx. R-r x / x ^ 3)
=
  (38627440277 / 434546112000)
  using chain[OF - - s62 r63] by (simp add: power2-eq-square power3-eq-cube)
  have r64: interval-lebesgue-integrable lborel (ereal (21 / 2)) (ereal (460 /
    43)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (21 / 2)) (ereal (460 / 43)) (λx. R-r x
/
  x ^ 3) = (- (43 / 8) / (460 / 43) - (- 197 / 2) / (2 * (460 / 43) ^ 2))
- (- (43 / 8) /
  (21 / 2) - (- 197 / 2) / (2 * (21 / 2) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-64)
  have s64: interval-lebesgue-integrable lborel (ereal 3) (ereal (460 / 43)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (460 / 43)) (λx. R-r x / x ^
3) =
  (8077887148897 / 98517811392000)
  using chain[OF - - s63 r64] by (simp add: power2-eq-square power3-eq-cube)
  have r65: interval-lebesgue-integrable lborel (ereal (460 / 43)) (ereal (400 /
    37)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (460 / 43)) (ereal (400 / 37)) (λx. R-r
x /
  x ^ 3) = (- (43 / 8) / (400 / 37) - (- 197 / 2) / (2 * (400 / 37) ^ 2))
- (- (43 / 8) /
  (460 / 43) - (- 197 / 2) / (2 * (460 / 43) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-65)
  have s65: interval-lebesgue-integrable lborel (ereal 3) (ereal (400 / 37)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (400 / 37)) (λx. R-r x / x ^
3) =
  (145806399481 / 1862340480000)
  using chain[OF - - s64 r65] by (simp add: power2-eq-square power3-eq-cube)
  have r66: interval-lebesgue-integrable lborel (ereal (400 / 37)) (ereal (250 /
    23)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (400 / 37)) (ereal (250 / 23)) (λx. R-r
x /
  x ^ 3) = (- 10 / (250 / 23) - (- 297 / 2) / (2 * (250 / 23) ^ 2)) - (-
10 / (400 / 37) -
  (- 297 / 2) / (2 * (400 / 37) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-66)
  have s66: interval-lebesgue-integrable lborel (ereal 3) (ereal (250 / 23)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (250 / 23)) (λx. R-r x / x ^
3) =

```



```

(1779523554587 / 23279256000000)
using chain[OF - - s65 r66] by (simp add: power2-eq-square power3-eq-cube)
have r67: interval-lebesgue-integrable lborel (ereal (250 / 23)) (ereal 11) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (250 / 23)) (ereal 11) (λx. R-r x / x ^
3) =
(- (177 / 20) / 11 - (- 136) / (2 * 11 ^ 2)) - (- (177 / 20) / (250 /
23) - (- 136) / (2 *
(250 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-67)
have s67: interval-lebesgue-integrable lborel (ereal 3) (ereal 11) (λx. R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal 11) (λx. R-r x / x ^ 3) =
(742899929809 / 10242872640000)
using chain[OF - - s66 r67] by (simp add: power2-eq-square power3-eq-cube)
have r68: interval-lebesgue-integrable lborel (ereal 11) (ereal (480 / 43)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 11) (ereal (480 / 43)) (λx. R-r x / x ^
3) =
(- 8 / (480 / 43) - (- 147) / (2 * (480 / 43) ^ 2)) - (- 8 / 11 - (-
147) / (2 * 11 ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-68)
have s68: interval-lebesgue-integrable lborel (ereal 3) (ereal (480 / 43)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (480 / 43)) (λx. R-r x / x ^
3) =
(244140840701 / 3724680960000)
using chain[OF - - s67 r68] by (simp add: power2-eq-square power3-eq-cube)
have r69: interval-lebesgue-integrable lborel (ereal (480 / 43)) (ereal (260 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (480 / 43)) (ereal (260 / 23)) (λx. R-r
x /
x ^ 3) = (- (191 / 40) / (260 / 23) - (- 111) / (2 * (260 / 23) ^ 2))
- (- (191 / 40) /
(480 / 43) - (- 111) / (2 * (480 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-69)
have s69: interval-lebesgue-integrable lborel (ereal 3) (ereal (260 / 23)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (260 / 23)) (λx. R-r x / x ^
3) =
(45257113067 / 756575820000)
using chain[OF - - s68 r69] by (simp add: power2-eq-square power3-eq-cube)
have r70: interval-lebesgue-integrable lborel (ereal (260 / 23)) (ereal (420 /
37)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (260 / 23)) (ereal (420 / 37)) (λx. R-r

```

$x /$
 $x \wedge 3 = (- (29 / 8) / (420 / 37) - (- 98) / (2 * (420 / 37) \wedge 2)) - (-$
 $(29 / 8) / (260 /$
 $23) - (- 98) / (2 * (260 / 23) \wedge 2))$
by (rule piece-rat) (simp-all add: inner-piece-70)
have s70: interval-lebesgue-integrable lborel (ereal 3) (ereal (420 / 37)) ($\lambda x. R-r$
 $x /$
 $x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (420 / 37)) ($\lambda x. R-r x / x \wedge$
 $3) =$
 $(3374166109 / 58198140000)$
using chain[OF - - s69 r70] **by** (simp add: power2-eq-square power3-eq-cube)
have r71: interval-lebesgue-integrable lborel (ereal (420 / 37)) (ereal (23 /
 $2)) (\lambda x. R-r x / x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal (420 / 37)) (ereal (23 / 2)) ($\lambda x. R-r x$
 $/$
 $x \wedge 3) = (- (219 / 40) / (23 / 2) - (- 119) / (2 * (23 / 2) \wedge 2)) - (-$
 $(219 / 40) / (420 /$
 $37) - (- 119) / (2 * (420 / 37) \wedge 2))$
by (rule piece-rat) (simp-all add: inner-piece-71)
have s71: interval-lebesgue-integrable lborel (ereal 3) (ereal (23 / 2)) ($\lambda x. R-r x$
 $/$
 $x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (23 / 2)) ($\lambda x. R-r x / x \wedge 3)$
 $=$
 $(1611729740561 / 30786816060000)$
using chain[OF - - s70 r71] **by** (simp add: power2-eq-square power3-eq-cube)
have r72: interval-lebesgue-integrable lborel (ereal (23 / 2)) (ereal (500 /
 $43)) (\lambda x. R-r x / x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal (23 / 2)) (ereal (500 / 43)) ($\lambda x. R-r x$
 $/$
 $x \wedge 3) = (- (259 / 40) / (500 / 43) - (- 261 / 2) / (2 * (500 / 43) \wedge$
 $2)) - (- (259 / 40) /$
 $(23 / 2) - (- 261 / 2) / (2 * (23 / 2) \wedge 2))$
by (rule piece-rat) (simp-all add: inner-piece-72)
have s72: interval-lebesgue-integrable lborel (ereal 3) (ereal (500 / 43)) ($\lambda x. R-r$
 $x /$
 $x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (500 / 43)) ($\lambda x. R-r x / x \wedge$
 $3) =$
 $(3195807504629 / 66927861000000)$
using chain[OF - - s71 r72] **by** (simp add: power2-eq-square power3-eq-cube)
have r73: interval-lebesgue-integrable lborel (ereal (500 / 43)) (ereal (270 /
 $23)) (\lambda x. R-r x / x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal (500 / 43)) (ereal (270 / 23)) ($\lambda x. R-r$
 $x /$
 $x \wedge 3) = (- (259 / 40) / (270 / 23) - (- 261 / 2) / (2 * (270 / 23) \wedge$
 $2)) - (- (259 / 40) /$
 $(500 / 43) - (- 261 / 2) / (2 * (500 / 43) \wedge 2))$

by (rule piece-rat) (simp-all add: inner-piece-73)
have s73: interval-lebesgue-integrable lborel (ereal 3) (ereal (270 / 23)) ($\lambda x. R\text{-}r$
 $x /$
 $x^{\wedge} 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (270 / 23)) ($\lambda x. R\text{-}r x / x^{\wedge}$
3) =
(529160461763 / 12047014980000)
using chain[OF - - s72 r73] **by** (simp add: power2-eq-square power3-eq-cube)
have r74: interval-lebesgue-integrable lborel (ereal (270 / 23)) (ereal (440 /
37)) ($\lambda x. R\text{-}r x / x^{\wedge} 3) \wedge$
interval-lebesgue-integral lborel (ereal (270 / 23)) (ereal (440 / 37)) ($\lambda x. R\text{-}r$
 $x /$
 $x^{\wedge} 3) = (- (213 / 40) / (440 / 37) - (- 117) / (2 * (440 / 37)^{\wedge} 2))$
 $- (- (213 / 40) /$
 $(270 / 23) - (- 117) / (2 * (270 / 23)^{\wedge} 2))$
by (rule piece-rat) (simp-all add: inner-piece-74)
have s74: interval-lebesgue-integrable lborel (ereal 3) (ereal (440 / 37)) ($\lambda x. R\text{-}r$
 $x /$
 $x^{\wedge} 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (440 / 37)) ($\lambda x. R\text{-}r x / x^{\wedge}$
3) =
(6875944213399 / 176689553040000)
using chain[OF - - s73 r74] **by** (simp add: power2-eq-square power3-eq-cube)
have r75: interval-lebesgue-integrable lborel (ereal (440 / 37)) (ereal 12) ($\lambda x. R\text{-}r$
 $x /$
 $x^{\wedge} 3) \wedge$
interval-lebesgue-integral lborel (ereal (440 / 37)) (ereal 12) ($\lambda x. R\text{-}r x / x^{\wedge}$
3) =
 $(- (199 / 20) / 12 - (- 172) / (2 * 12^{\wedge} 2)) - (- (199 / 20) / (440 /$
37) - $(- 172) / (2 *$
 $(440 / 37)^{\wedge} 2))$
by (rule piece-rat) (simp-all add: inner-piece-75)
have s75: interval-lebesgue-integrable lborel (ereal 3) (ereal 12) ($\lambda x. R\text{-}r x /$
 $x^{\wedge} 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal 12) ($\lambda x. R\text{-}r x / x^{\wedge} 3) =$
(570953551109 / 16062686640000)
using chain[OF - - s74 r75] **by** (simp add: power2-eq-square power3-eq-cube)
have r76: interval-lebesgue-integrable lborel (ereal 12) (ereal (520 / 43)) ($\lambda x. R\text{-}r$
 $x /$
 $x^{\wedge} 3) \wedge$
interval-lebesgue-integral lborel (ereal 12) (ereal (520 / 43)) ($\lambda x. R\text{-}r x / x^{\wedge}$
3) =
 $(- (91 / 10) / (520 / 43) - (- 184) / (2 * (520 / 43)^{\wedge} 2)) - (- (91 /$
10) / $12 - (- 184) /$
 $(2 * 12^{\wedge} 2))$
by (rule piece-rat) (simp-all add: inner-piece-76)
have s76: interval-lebesgue-integrable lborel (ereal 3) (ereal (520 / 43)) ($\lambda x. R\text{-}r$
 $x /$
 $x^{\wedge} 3) \wedge$

```

interval-lebesgue-integral lborel (ereal 3) (ereal (520 / 43)) (λx. R-r x / x ^
3) =
  (6595922886017 / 208814926320000)
  using chain[OF - - s75 r76] by (simp add: power2-eq-square power3-eq-cube)
  have r77: interval-lebesgue-integrable lborel (ereal (520 / 43)) (ereal (280 /
23)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (520 / 43)) (ereal (280 / 23)) (λx. R-r
x /
  x ^ 3) = (- (47 / 8) / (280 / 23) - (- 145) / (2 * (280 / 23) ^ 2)) -
(- (47 / 8) / (520 /
43) - (- 145) / (2 * (520 / 43) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-77)
  have s77: interval-lebesgue-integrable lborel (ereal 3) (ereal (280 / 23)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (280 / 23)) (λx. R-r x / x ^
3) =
  (3176305988663 / 112438806480000)
  using chain[OF - - s76 r77] by (simp add: power2-eq-square power3-eq-cube)
  have r78: interval-lebesgue-integrable lborel (ereal (280 / 23)) (ereal (460 /
37)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (280 / 23)) (ereal (460 / 37)) (λx. R-r
x /
  x ^ 3) = (- (189 / 40) / (460 / 37) - (- 131) / (2 * (460 / 37) ^ 2))
- (- (189 / 40) /
(280 / 23) - (- 131) / (2 * (280 / 23) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-78)
  have s78: interval-lebesgue-integrable lborel (ereal 3) (ereal (460 / 37)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (460 / 37)) (λx. R-r x / x ^
3) =
  (478449408113 / 26388699480000)
  using chain[OF - - s77 r78] by (simp add: power2-eq-square power3-eq-cube)
  have r79: interval-lebesgue-integrable lborel (ereal (460 / 37)) (ereal (25 /
2)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (460 / 37)) (ereal (25 / 2)) (λx. R-r x
/
  x ^ 3) = (- (263 / 40) / (25 / 2) - (- 154) / (2 * (25 / 2) ^ 2)) - (-
(263 / 40) / (460 /
37) - (- 154) / (2 * (460 / 37) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-79)
  have s79: interval-lebesgue-integrable lborel (ereal 3) (ereal (25 / 2)) (λx. R-r x
/
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (25 / 2)) (λx. R-r x / x ^ 3)
=
  (17919605149 / 1147334760000)
  using chain[OF - - s78 r79] by (simp add: power2-eq-square power3-eq-cube)

```

```

have r80: interval-lebesgue-integrable lborel (ereal (25 / 2)) (ereal (540 /
43)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (25 / 2)) (ereal (540 / 43)) (λx. R-r x
/
  x ^ 3) = (− (303 / 40) / (540 / 43) − (− 333 / 2) / (2 * (540 / 43) ^
2)) − (− (303 / 40) /
  (25 / 2) − (− 333 / 2) / (2 * (25 / 2) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-80)
have s80: interval-lebesgue-integrable lborel (ereal 3) (ereal (540 / 43)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (540 / 43)) (λx. R-r x / x ^
3) =
  (11618575517 / 860501070000)
using chain[OF - - s79 r80] by (simp add: power2-eq-square power3-eq-cube)
have r81: interval-lebesgue-integrable lborel (ereal (540 / 43)) (ereal (290 /
23)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (540 / 43)) (ereal (290 / 23)) (λx. R-r
x /
  x ^ 3) = (− (303 / 40) / (290 / 23) − (− 333 / 2) / (2 * (290 / 23) ^
2)) − (− (303 / 40) /
  (540 / 43) − (− 333 / 2) / (2 * (540 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-81)
have s81: interval-lebesgue-integrable lborel (ereal 3) (ereal (290 / 23)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (290 / 23)) (λx. R-r x / x ^
3) =
  (11285560024021 / 964908533160000)
using chain[OF - - s80 r81] by (simp add: power2-eq-square power3-eq-cube)
have r82: interval-lebesgue-integrable lborel (ereal (290 / 23)) (ereal (480 /
37)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (290 / 23)) (ereal (480 / 37)) (λx. R-r
x /
  x ^ 3) = (− (257 / 40) / (480 / 37) − (− 152) / (2 * (480 / 37) ^ 2))
− (− (257 / 40) /
  (290 / 23) − (− 152) / (2 * (290 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-82)
have s82: interval-lebesgue-integrable lborel (ereal 3) (ereal (480 / 37)) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (480 / 37)) (λx. R-r x / x ^
3) =
  (− 3872453569 / 8318177010000)
using chain[OF - - s81 r82] by (simp add: power2-eq-square power3-eq-cube)
have r83: interval-lebesgue-integrable lborel (ereal (480 / 37)) (ereal 13) (λx. R-r
x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (480 / 37)) (ereal 13) (λx. R-r x / x ^

```

$3) =$
 $(- (221 / 20) / 13 - (- 212) / (2 * 13 ^ 2)) - (- (221 / 20) / (480 /$
 $37) - (- 212) / (2 *$
 $(480 / 37) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-83)
have s83: interval-lebesgue-integrable lborel (ereal 3) (ereal 13) ($\lambda x. R-r x /$
 $x ^ 3) \wedge$
 $interval-lebesgue-integral lborel (ereal 3) (ereal 13) (\lambda x. R-r x / x ^ 3) =$
 $(- 4535988527729 / 3460361636160000)$
using chain[OF - - s82 r83] **by** (simp add: power2-eq-square power3-eq-cube)
have r84: interval-lebesgue-integrable lborel (ereal 13) (ereal (560 / 43)) ($\lambda x. R-r x /$
 $x ^ 3) \wedge$
 $interval-lebesgue-integral lborel (ereal 13) (ereal (560 / 43)) (\lambda x. R-r x / x ^$
 $3) =$
 $(- (51 / 5) / (560 / 43) - (- 225) / (2 * (560 / 43) ^ 2)) - (- (51 /$
 $5) / 13 - (- 225) /$
 $(2 * 13 ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-84)
have s84: interval-lebesgue-integrable lborel (ereal 3) (ereal (560 / 43)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) \wedge$
 $interval-lebesgue-integral lborel (ereal 3) (ereal (560 / 43)) (\lambda x. R-r x / x ^$
 $3) =$
 $(- 1862730944387 / 815181346980000)$
using chain[OF - - s83 r84] **by** (simp add: power2-eq-square power3-eq-cube)
have r85: interval-lebesgue-integrable lborel (ereal (560 / 43)) (ereal (300 /
 $23)) (\lambda x. R-r x / x ^ 3) \wedge$
 $interval-lebesgue-integral lborel (ereal (560 / 43)) (ereal (300 / 23)) (\lambda x. R-r$
 $x /$
 $x ^ 3) = (- (279 / 40) / (300 / 23) - (- 183) / (2 * (300 / 23) ^ 2))$
 $- (- (279 / 40) /$
 $(560 / 43) - (- 183) / (2 * (560 / 43) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-85)
have s85: interval-lebesgue-integrable lborel (ereal 3) (ereal (300 / 23)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) \wedge$
 $interval-lebesgue-integral lborel (ereal 3) (ereal (300 / 23)) (\lambda x. R-r x / x ^$
 $3) =$
 $(- 1165002666109 / 372654330048000)$
using chain[OF - - s84 r85] **by** (simp add: power2-eq-square power3-eq-cube)
have r86: interval-lebesgue-integrable lborel (ereal (300 / 23)) (ereal (40 /
 $3)) (\lambda x. R-r x / x ^ 3) \wedge$
 $interval-lebesgue-integral lborel (ereal (300 / 23)) (ereal (40 / 3)) (\lambda x. R-r x$
 $/$
 $x ^ 3) = (- (233 / 40) / (40 / 3) - (- 168) / (2 * (40 / 3) ^ 2)) - (-$
 $(233 / 40) / (300 /$
 $23) - (- 168) / (2 * (300 / 23) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-86)

```

have s86: interval-lebesgue-integrable lborel (ereal 3) (ereal (40 / 3)) ( $\lambda x. R\text{-}r\ x$ 
/
 $x \wedge 3$ )  $\wedge$ 
interval-lebesgue-integral lborel (ereal 3) (ereal (40 / 3)) ( $\lambda x. R\text{-}r\ x / x \wedge 3$ )
=
(- 27299219099561 / 1863271650240000)
using chain[OF - - s85 r86] by (simp add: power2-eq-square power3-eq-cube)
have r87: interval-lebesgue-integrable lborel (ereal (40 / 3)) (ereal (310 /
23)) ( $\lambda x. R\text{-}r\ x / x \wedge 3$ )  $\wedge$ 
interval-lebesgue-integral lborel (ereal (40 / 3)) (ereal (310 / 23)) ( $\lambda x. R\text{-}r\ x$ 
/
 $x \wedge 3$ ) = (- (145 / 8) / (310 / 23) - (- 184) / (2 * (310 / 23)  $\wedge$  2)) -
(- (145 / 8) / (40 /
3) - (- 184) / (2 * (40 / 3)  $\wedge$  2))
by (rule piece-rat) (simp-all add: inner-piece-87)
have s87: interval-lebesgue-integrable lborel (ereal 3) (ereal (310 / 23)) ( $\lambda x. R\text{-}r$ 
 $x /$ 
 $x \wedge 3$ )  $\wedge$ 
interval-lebesgue-integral lborel (ereal 3) (ereal (310 / 23)) ( $\lambda x. R\text{-}r\ x / x \wedge$ 
3) =
(- 19881957772141121 / 1790604055880640000)
using chain[OF - - s86 r87] by (simp add: power2-eq-square power3-eq-cube)
have r88: interval-lebesgue-integrable lborel (ereal (310 / 23)) (ereal (580 /
43)) ( $\lambda x. R\text{-}r\ x / x \wedge 3$ )  $\wedge$ 
interval-lebesgue-integral lborel (ereal (310 / 23)) (ereal (580 / 43)) ( $\lambda x. R\text{-}r$ 
 $x /$ 
 $x \wedge 3$ ) = (- (679 / 40) / (580 / 43) - (- 337 / 2) / (2 * (580 / 43)  $\wedge$ 
2)) - (- (679 / 40) /
(310 / 23) - (- 337 / 2) / (2 * (310 / 23)  $\wedge$  2))
by (rule piece-rat) (simp-all add: inner-piece-88)
have s88: interval-lebesgue-integrable lborel (ereal 3) (ereal (580 / 43)) ( $\lambda x. R\text{-}r$ 
 $x /$ 
 $x \wedge 3$ )  $\wedge$ 
interval-lebesgue-integral lborel (ereal 3) (ereal (580 / 43)) ( $\lambda x. R\text{-}r\ x / x \wedge$ 
3) =
(- 18182057508830539 / 1675081213565760000)
using chain[OF - - s87 r88] by (simp add: power2-eq-square power3-eq-cube)
have r89: interval-lebesgue-integrable lborel (ereal (580 / 43)) (ereal (27 /
2)) ( $\lambda x. R\text{-}r\ x / x \wedge 3$ )  $\wedge$ 
interval-lebesgue-integral lborel (ereal (580 / 43)) (ereal (27 / 2)) ( $\lambda x. R\text{-}r\ x$ 
/
 $x \wedge 3$ ) = (- (679 / 40) / (27 / 2) - (- 337 / 2) / (2 * (27 / 2)  $\wedge$  2)) -
(- (679 / 40) /
(580 / 43) - (- 337 / 2) / (2 * (580 / 43)  $\wedge$  2))
by (rule piece-rat) (simp-all add: inner-piece-89)
have s89: interval-lebesgue-integrable lborel (ereal 3) (ereal (27 / 2)) ( $\lambda x. R\text{-}r\ x$ 
/
 $x \wedge 3$ )  $\wedge$ 
interval-lebesgue-integral lborel (ereal 3) (ereal (27 / 2)) ( $\lambda x. R\text{-}r\ x / x \wedge 3$ )

```

```

=
  (− 16481154814890757 / 1559558371250880000)
  using chain[OF - - s88 r89] by (simp add: power2-eq-square power3-eq-cube)
  have r90: interval-lebesgue-integrable lborel (ereal (27 / 2)) (ereal (500 /
    37)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (27 / 2)) (ereal (500 / 37)) (λx. R-r x
  /
    x ^ 3) = (− (719 / 40) / (500 / 37) − (− 182) / (2 * (500 / 37) ^ 2))
  − (− (719 / 40) /
    (27 / 2) − (− 182) / (2 * (27 / 2) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-90)
  have s90: interval-lebesgue-integrable lborel (ereal 3) (ereal (500 / 37)) (λx. R-r
  x /
    x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (500 / 37)) (λx. R-r x / x ^
  3) =
    (− 547368726821237 / 53482797368000000)
    using chain[OF - - s89 r90] by (simp add: power2-eq-square power3-eq-cube)
    have r91: interval-lebesgue-integrable lborel (ereal (500 / 37)) (ereal (320 /
      23)) (λx. R-r x / x ^ 3) ∧
        interval-lebesgue-integral lborel (ereal (500 / 37)) (ereal (320 / 23)) (λx. R-r
      x /
        x ^ 3) = (− (793 / 40) / (320 / 23) − (− 207) / (2 * (320 / 23) ^ 2))
      − (− (793 / 40) /
        (500 / 37) − (− 207) / (2 * (500 / 37) ^ 2))
      by (rule piece-rat) (simp-all add: inner-piece-91)
      have s91: interval-lebesgue-integrable lborel (ereal 3) (ereal (320 / 23)) (λx. R-r
      x /
        x ^ 3) ∧
        interval-lebesgue-integral lborel (ereal 3) (ereal (320 / 23)) (λx. R-r x / x ^
      3) =
        (− 6494602107759 / 34228990315520000)
        using chain[OF - - s90 r91] by (simp add: power2-eq-square power3-eq-cube)
        have r92: interval-lebesgue-integrable lborel (ereal (320 / 23)) (ereal (600 /
          43)) (λx. R-r x / x ^ 3) ∧
            interval-lebesgue-integral lborel (ereal (320 / 23)) (ereal (600 / 43)) (λx. R-r
          x /
            x ^ 3) = (− (747 / 40) / (600 / 43) − (− 191) / (2 * (600 / 43) ^ 2))
          − (− (747 / 40) /
            (320 / 23) − (− 191) / (2 * (320 / 23) ^ 2))
          by (rule piece-rat) (simp-all add: inner-piece-92)
          have s92: interval-lebesgue-integrable lborel (ereal 3) (ereal (600 / 43)) (λx. R-r
          x /
            x ^ 3) ∧
            interval-lebesgue-integral lborel (ereal 3) (ereal (600 / 43)) (λx. R-r x / x ^
          3) =
            (6355291472 / 7521018379875)
            using chain[OF - - s91 r92] by (simp add: power2-eq-square power3-eq-cube)
            have r93: interval-lebesgue-integrable lborel (ereal (600 / 43)) (ereal 14) (λx.

```



```

R-r x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (600 / 43)) (ereal 14) (λx. R-r x / x ^
3) =
  (- (309 / 20) / 14 - (- 146) / (2 * 14 ^ 2)) - (- (309 / 20) / (600 /
43) - (- 146) / (2 *
  (600 / 43) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-93)
have s93: interval-lebesgue-integrable lborel (ereal 3) (ereal 14) (λx. R-r x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal 14) (λx. R-r x / x ^ 3) =
    (3812015306609 / 1871897907880000)
  using chain[OF - - s92 r93] by (simp add: power2-eq-square power3-eq-cube)
have r94: interval-lebesgue-integrable lborel (ereal 14) (ereal (520 / 37)) (λx.
R-r x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 14) (ereal (520 / 37)) (λx. R-r x / x ^
3) =
  (- (73 / 5) / (520 / 37) - (- 160) / (2 * (520 / 37) ^ 2)) - (- (73 /
5) / 14 - (- 160) /
  (2 * 14 ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-94)
have s94: interval-lebesgue-integrable lborel (ereal 3) (ereal (520 / 37)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (520 / 37)) (λx. R-r x / x ^
3) =
    (10129319814931 / 3476381828920000)
  using chain[OF - - s93 r94] by (simp add: power2-eq-square power3-eq-cube)
have r95: interval-lebesgue-integrable lborel (ereal (520 / 37)) (ereal (330 /
23)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (520 / 37)) (ereal (330 / 23)) (λx. R-r
x /
  x ^ 3) = (- (769 / 40) / (330 / 23) - (- 225) / (2 * (330 / 23) ^ 2))
  - (- (769 / 40) /
    (520 / 37) - (- 225) / (2 * (520 / 37) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-95)
have s95: interval-lebesgue-integrable lborel (ereal 3) (ereal (330 / 23)) (λx. R-r
x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (330 / 23)) (λx. R-r x / x ^
3) =
    (138316892681717 / 17649323131440000)
  using chain[OF - - s94 r95] by (simp add: power2-eq-square power3-eq-cube)
have r96: interval-lebesgue-integrable lborel (ereal (330 / 23)) (ereal (620 /
43)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (330 / 23)) (ereal (620 / 43)) (λx. R-r
x /
  x ^ 3) = (- (723 / 40) / (620 / 43) - (- 417 / 2) / (2 * (620 / 43) ^

```

$2)) - (- (723 / 40) /$
 $(330 / 23) - (- 417 / 2) / (2 * (330 / 23) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-96)
have s96: interval-lebesgue-integrable lborel (ereal 3) (ereal (620 / 43)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (620 / 43)) ($\lambda x. R-r x / x ^$
 $3) =$
 $(40973121188137 / 4521727413840000)$
using chain[OF - - s95 r96] **by** (simp add: power2-eq-square power3-eq-cube)
have r97: interval-lebesgue-integrable lborel (ereal (620 / 43)) (ereal (29 /
 $2)) (\lambda x. R-r x / x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal (620 / 43)) (ereal (29 / 2)) ($\lambda x. R-r x$
 $/$
 $x ^ 3) = (- (723 / 40) / (29 / 2) - (- 417 / 2) / (2 * (29 / 2) ^ 2)) -$
 $(- (723 / 40) /$
 $(620 / 43) - (- 417 / 2) / (2 * (620 / 43) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-97)
have s97: interval-lebesgue-integrable lborel (ereal 3) (ereal (29 / 2)) ($\lambda x. R-r x$
 $/$
 $x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (29 / 2)) ($\lambda x. R-r x / x ^ 3)$
 $=$
 $(44349030780233 / 4230003064560000)$
using chain[OF - - s96 r97] **by** (simp add: power2-eq-square power3-eq-cube)
have r98: interval-lebesgue-integrable lborel (ereal (29 / 2)) (ereal (540 /
 $37)) (\lambda x. R-r x / x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal (29 / 2)) (ereal (540 / 37)) ($\lambda x. R-r x$
 $/$
 $x ^ 3) = (- (763 / 40) / (540 / 37) - (- 223) / (2 * (540 / 37) ^ 2))$
 $- (- (763 / 40) /$
 $(29 / 2) - (- 223) / (2 * (29 / 2) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-98)
have s98: interval-lebesgue-integrable lborel (ereal 3) (ereal (540 / 37)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (540 / 37)) ($\lambda x. R-r x / x ^$
 $3) =$
 $(430957159001711 / 35444508437520000)$
using chain[OF - - s97 r98] **by** (simp add: power2-eq-square power3-eq-cube)
have r99: interval-lebesgue-integrable lborel (ereal (540 / 37)) (ereal (340 /
 $23)) (\lambda x. R-r x / x ^ 3) \wedge$
interval-lebesgue-integral lborel (ereal (540 / 37)) (ereal (340 / 23)) ($\lambda x. R-r$
 $x /$
 $x ^ 3) = (- (837 / 40) / (340 / 23) - (- 250) / (2 * (340 / 23) ^ 2))$
 $- (- (837 / 40) /$
 $(540 / 37) - (- 250) / (2 * (540 / 37) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-99)
have s99: interval-lebesgue-integrable lborel (ereal 3) (ereal (340 / 23)) ($\lambda x. R-r$

```

x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (340 / 23)) (λx. R-r x / x ^
3) =
  (347275409950381 / 22316912719920000)
  using chain[OF - - s98 r99] by (simp add: power2-eq-square power3-eq-cube)
  have r100: interval-lebesgue-integrable lborel (ereal (340 / 23)) (ereal (640 /
43)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (340 / 23)) (ereal (640 / 43)) (λx. R-r
x /
  x ^ 3) = (- (791 / 40) / (640 / 43) - (- 233) / (2 * (640 / 43) ^ 2))
- (- (791 / 40) /
  (340 / 23) - (- 233) / (2 * (340 / 23) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-100)
  have s100: interval-lebesgue-integrable lborel (ereal 3) (ereal (640 / 43)) (λx.
R-r x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (640 / 43)) (λx. R-r x / x ^
3) =
  (5857641039199583 / 336066450370560000)
  using chain[OF - - s99 r100] by (simp add: power2-eq-square power3-eq-cube)
  have r101: interval-lebesgue-integrable lborel (ereal (640 / 43)) (ereal 15) (λx.
R-r x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (640 / 43)) (ereal 15) (λx. R-r x / x ^
3) =
  (- (331 / 20) / 15 - (- 185) / (2 * 15 ^ 2)) - (- (331 / 20) / (640 /
43) - (- 185) / (2 *
  (640 / 43) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-101)
  have s101: interval-lebesgue-integrable lborel (ereal 3) (ereal 15) (λx. R-r x /
x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal 15) (λx. R-r x / x ^ 3) =
  (411704436619163 / 21004153148160000)
  using chain[OF - - s100 r101] by (simp add: power2-eq-square power3-eq-cube)
  have r102: interval-lebesgue-integrable lborel (ereal 15) (ereal (560 / 37)) (λx.
R-r x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 15) (ereal (560 / 37)) (λx. R-r x / x ^
3) =
  (- (157 / 10) / (560 / 37) - (- 200) / (2 * (560 / 37) ^ 2)) - (- (157
/ 10) / 15 -
  (- 200) / (2 * 15 ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-102)
  have s102: interval-lebesgue-integrable lborel (ereal 3) (ereal (560 / 37)) (λx.
R-r x /
  x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (560 / 37)) (λx. R-r x / x ^
3) =

```

```

(3094264707702941 / 147029072037120000)
using chain[OF - - s101 r102] by (simp add: power2-eq-square power3-eq-cube)
have r103: interval-lebesgue-integrable lborel (ereal (560 / 37)) (ereal (350 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (560 / 37)) (ereal (350 / 23)) (λx. R-r
x /
x ^ 3) = (- (813 / 40) / (350 / 23) - (- 270) / (2 * (350 / 23) ^ 2))
- (- (813 / 40) /
(560 / 37) - (- 270) / (2 * (560 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-103)
have s103: interval-lebesgue-integrable lborel (ereal 3) (ereal (350 / 23)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (350 / 23)) (λx. R-r x / x ^
3) =
(3227322267154901 / 147029072037120000)
using chain[OF - - s102 r103] by (simp add: power2-eq-square power3-eq-cube)
have r104: interval-lebesgue-integrable lborel (ereal (350 / 23)) (ereal (660 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (350 / 23)) (ereal (660 / 43)) (λx. R-r
x /
x ^ 3) = (- (767 / 40) / (660 / 43) - (- 505 / 2) / (2 * (660 / 43) ^
2)) - (- (767 / 40) /
(350 / 23) - (- 505 / 2) / (2 * (350 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-104)
have s104: interval-lebesgue-integrable lborel (ereal 3) (ereal (660 / 43)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (660 / 43)) (λx. R-r x / x ^
3) =
(59581386442537163 / 2541502530927360000)
using chain[OF - - s103 r104] by (simp add: power2-eq-square power3-eq-cube)
have r105: interval-lebesgue-integrable lborel (ereal (660 / 43)) (ereal (31 /
2)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (660 / 43)) (ereal (31 / 2)) (λx. R-r x
/
x ^ 3) = (- (767 / 40) / (31 / 2) - (- 505 / 2) / (2 * (31 / 2) ^ 2)) -
(- (767 / 40) /
(660 / 43) - (- 505 / 2) / (2 * (660 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-105)
have s105: interval-lebesgue-integrable lborel (ereal 3) (ereal (31 / 2)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (31 / 2)) (λx. R-r x / x ^ 3)
=
(16424908746219293 / 651128747592960000)
using chain[OF - - s104 r105] by (simp add: power2-eq-square power3-eq-cube)
have r106: interval-lebesgue-integrable lborel (ereal (31 / 2)) (ereal (360 /
23)) (λx. R-r x / x ^ 3) ∧

```

```

    interval-lebesgue-integral lborel (ereal (31 / 2)) (ereal (360 / 23)) (λx. R-r x
/
    x ^ 3) = (- (807 / 40) / (360 / 23) - (- 268) / (2 * (360 / 23) ^ 2))
- (- (807 / 40) /
    (31 / 2) - (- 268) / (2 * (31 / 2) ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-106)
    have s106: interval-lebesgue-integrable lborel (ereal 3) (ereal (360 / 23)) (λx.
R-r x /
    x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (360 / 23)) (λx. R-r x / x ^
3) =
    (1706843121663209 / 6301245944480000)
    using chain[OF - - s105 r106] by (simp add: power2-eq-square power3-eq-cube)
    have r107: interval-lebesgue-integrable lborel (ereal (360 / 23)) (ereal (580 /
37)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (360 / 23)) (ereal (580 / 37)) (λx. R-r
x /
    x ^ 3) = (- (761 / 40) / (580 / 37) - (- 250) / (2 * (580 / 37) ^ 2))
- (- (761 / 40) /
    (360 / 23) - (- 250) / (2 * (360 / 23) ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-107)
    have s107: interval-lebesgue-integrable lborel (ereal 3) (ereal (580 / 37)) (λx.
R-r x /
    x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (580 / 37)) (λx. R-r x / x ^
3) =
    (16678296005188087 / 609120441296640000)
    using chain[OF - - s106 r107] by (simp add: power2-eq-square power3-eq-cube)
    have r108: interval-lebesgue-integrable lborel (ereal (580 / 37)) (ereal (680 /
43)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (580 / 37)) (ereal (680 / 43)) (λx. R-r
x /
    x ^ 3) = (- (167 / 8) / (680 / 43) - (- 279) / (2 * (680 / 43) ^ 2)) -
(- (167 / 8) / (580 /
    37) - (- 279) / (2 * (580 / 37) ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-108)
    have s108: interval-lebesgue-integrable lborel (ereal 3) (ereal (680 / 43)) (λx.
R-r x /
    x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (680 / 43)) (λx. R-r x / x ^
3) =
    (10405244655250051 / 357070603518720000)
    using chain[OF - - s107 r108] by (simp add: power2-eq-square power3-eq-cube)
    have r109: interval-lebesgue-integrable lborel (ereal (680 / 43)) (ereal 16) (λx.
R-r x /
    x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (680 / 43)) (ereal 16) (λx. R-r x / x ^
3) =
    (- (353 / 20) / 16 - (- 228) / (2 * 16 ^ 2)) - (- (353 / 20) / (680 /

```

$43) - (-228) / (2 * (680 / 43) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-109)
have s109: interval-lebesgue-integrable lborel (ereal 3) (ereal 16) ($\lambda x. R-r x / x ^ 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal 16) ($\lambda x. R-r x / x ^ 3$) =
(663289007996003 / 21004153148160000)
using chain[OF - - s108 r109] **by** (simp add: power2-eq-square power3-eq-cube)
have r110: interval-lebesgue-integrable lborel (ereal 16) (ereal (370 / 23)) ($\lambda x. R-r x / x ^ 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 16) (ereal (370 / 23)) ($\lambda x. R-r x / x ^ 3$) =
 $(- (84 / 5) / (370 / 23) - (-244) / (2 * (370 / 23) ^ 2)) - (- (84 / 5) / 16 - (-244) / (2 * 16 ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-110)
have s110: interval-lebesgue-integrable lborel (ereal 3) (ereal (370 / 23)) ($\lambda x. R-r x / x ^ 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (370 / 23)) ($\lambda x. R-r x / x ^ 3$) =
(923500395904002107 / 28754685659831040000)
using chain[OF - - s109 r110] **by** (simp add: power2-eq-square power3-eq-cube)
have r111: interval-lebesgue-integrable lborel (ereal (370 / 23)) (ereal (600 / 37)) ($\lambda x. R-r x / x ^ 3$) $\wedge$
interval-lebesgue-integral lborel (ereal (370 / 23)) (ereal (600 / 37)) ($\lambda x. R-r x / x ^ 3$) =
 $(- (313 / 20) / (600 / 37) - (-451 / 2) / (2 * (600 / 37) ^ 2)) - (- (313 / 20) / (370 / 23) - (-451 / 2) / (2 * (370 / 23) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-111)
have s111: interval-lebesgue-integrable lborel (ereal 3) (ereal (600 / 37)) ($\lambda x. R-r x / x ^ 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (600 / 37)) ($\lambda x. R-r x / x ^ 3$) =
(25609594035043943 / 777153666481920000)
using chain[OF - - s110 r111] **by** (simp add: power2-eq-square power3-eq-cube)
have r112: interval-lebesgue-integrable lborel (ereal (600 / 37)) (ereal (700 / 43)) ($\lambda x. R-r x / x ^ 3$) $\wedge$
interval-lebesgue-integral lborel (ereal (600 / 37)) (ereal (700 / 43)) ($\lambda x. R-r x / x ^ 3$) =
 $(- (811 / 40) / (700 / 43) - (-601 / 2) / (2 * (700 / 43) ^ 2)) - (- (811 / 40) / (600 / 37) - (-601 / 2) / (2 * (600 / 37) ^ 2))$
by (rule piece-rat) (simp-all add: inner-piece-112)
have s112: interval-lebesgue-integrable lborel (ereal 3) (ereal (700 / 43)) ($\lambda x. R-r x /$

```

x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (700 / 43)) (λx. R-r x / x ^
3) =
(181572637023959273 / 5440075665373440000)
using chain[OF - - s111 r112] by (simp add: power2-eq-square power3-eq-cube)
have r113: interval-lebesgue-integrable lborel (ereal (700 / 43)) (ereal (33 /
2)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (700 / 43)) (ereal (33 / 2)) (λx. R-r x
/
x ^ 3) = (- (811 / 40) / (33 / 2) - (- 601 / 2) / (2 * (33 / 2) ^ 2)) -
(- (811 / 40) /
(700 / 43) - (- 601 / 2) / (2 * (700 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-113)
have s113: interval-lebesgue-integrable lborel (ereal 3) (ereal (33 / 2)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (33 / 2)) (λx. R-r x / x ^ 3)
=
(3288608185470880631 / 94035593644312320000)
using chain[OF - - s112 r113] by (simp add: power2-eq-square power3-eq-cube)
have r114: interval-lebesgue-integrable lborel (ereal (33 / 2)) (ereal (380 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (33 / 2)) (ereal (380 / 23)) (λx. R-r x
/
x ^ 3) = (- (851 / 40) / (380 / 23) - (- 317) / (2 * (380 / 23) ^ 2)) -
(- (851 / 40) /
(33 / 2) - (- 317) / (2 * (33 / 2) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-114)
have s114: interval-lebesgue-integrable lborel (ereal 3) (ereal (380 / 23)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (380 / 23)) (λx. R-r x / x ^
3) =
(5707206507596333599 / 162425116294721280000)
using chain[OF - - s113 r114] by (simp add: power2-eq-square power3-eq-cube)
have r115: interval-lebesgue-integrable lborel (ereal (380 / 23)) (ereal (50 /
3)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (380 / 23)) (ereal (50 / 3)) (λx. R-r x
/
x ^ 3) = (- (161 / 8) / (50 / 3) - (- 298) / (2 * (50 / 3) ^ 2)) - (-
(161 / 8) / (380 /
23) - (- 298) / (2 * (380 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-115)
have s115: interval-lebesgue-integrable lborel (ereal 3) (ereal (50 / 3)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (50 / 3)) (λx. R-r x / x ^ 3)
=
(310127401715264389 / 8548690331301120000)

```

```

using chain[OF - - s114 r115] by (simp add: power2-eq-square power3-eq-cube)
have r116: interval-lebesgue-integrable lborel (ereal (50 / 3)) (ereal (720 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (50 / 3)) (ereal (720 / 43)) (λx. R-r x
/
x ^ 3) = (- (799 / 40) / (720 / 43) - (- 591 / 2) / (2 * (720 / 43) ^
2)) - (- (799 / 40) /
(50 / 3) - (- 591 / 2) / (2 * (50 / 3) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-116)
have s116: interval-lebesgue-integrable lborel (ereal 3) (ereal (720 / 43)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (720 / 43)) (λx. R-r x / x ^
3) =
(315555919018815731 / 8548690331301120000)
using chain[OF - - s115 r116] by (simp add: power2-eq-square power3-eq-cube)
have r117: interval-lebesgue-integrable lborel (ereal (720 / 43)) (ereal (620 /
37)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (720 / 43)) (ereal (620 / 37)) (λx. R-r
x /
x ^ 3) = (- (67 / 4) / (620 / 37) - (- 483 / 2) / (2 * (620 / 37) ^ 2))
- (- (67 / 4) /
(720 / 43) - (- 483 / 2) / (2 * (720 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-117)
have s117: interval-lebesgue-integrable lborel (ereal 3) (ereal (620 / 37)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (620 / 37)) (λx. R-r x / x ^
3) =
(9809927604509515411 / 265009400270334720000)
using chain[OF - - s116 r117] by (simp add: power2-eq-square power3-eq-cube)
have r118: interval-lebesgue-integrable lborel (ereal (620 / 37)) (ereal (390 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (620 / 37)) (ereal (390 / 23)) (λx. R-r
x /
x ^ 3) = (- (93 / 5) / (390 / 23) - (- 545 / 2) / (2 * (390 / 23) ^ 2))
- (- (93 / 5) /
(620 / 37) - (- 545 / 2) / (2 * (620 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-118)
have s118: interval-lebesgue-integrable lborel (ereal 3) (ereal (390 / 23)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (390 / 23)) (λx. R-r x / x ^
3) =
(4303992980933782153 / 111132974306914560000)
using chain[OF - - s117 r118] by (simp add: power2-eq-square power3-eq-cube)
have r119: interval-lebesgue-integrable lborel (ereal (390 / 23)) (ereal 17) (λx.
R-r x /
x ^ 3) ∧

```



```

    interval-lebesgue-integral lborel (ereal (390 / 23)) (ereal 17) (λx. R-r x / x ^
3) =
    (- (349 / 20) / 17 - (- 253) / (2 * 17 ^ 2)) - (- (349 / 20) / (390 /
23) - (- 253) / (2 *
(390 / 23) ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-119)
    have s119: interval-lebesgue-integrable lborel (ereal 3) (ereal 17) (λx. R-r x /
x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal 17) (λx. R-r x / x ^ 3) =
    (5684162388119036477 / 145327735632119040000)
    using chain[OF - - s118 r119] by (simp add: power2-eq-square power3-eq-cube)
    have r120: interval-lebesgue-integrable lborel (ereal 17) (ereal (740 / 43)) (λx.
R-r x /
x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 17) (ereal (740 / 43)) (λx. R-r x / x ^
3) =
    (- (83 / 5) / (740 / 43) - (- 270) / (2 * (740 / 43) ^ 2)) - (- (83 /
5) / 17 - (- 270) /
(2 * 17 ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-120)
    have s120: interval-lebesgue-integrable lborel (ereal 3) (ereal (740 / 43)) (λx.
R-r x /
x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (740 / 43)) (λx. R-r x / x ^
3) =
    (12555663195326239297 / 316301542258141440000)
    using chain[OF - - s119 r120] by (simp add: power2-eq-square power3-eq-cube)
    have r121: interval-lebesgue-integrable lborel (ereal (740 / 43)) (ereal (640 /
37)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (740 / 43)) (ereal (640 / 37)) (λx. R-r
x /
x ^ 3) = (- (83 / 5) / (640 / 37) - (- 270) / (2 * (640 / 37) ^ 2)) -
(- (83 / 5) / (740 /
43) - (- 270) / (2 * (740 / 43) ^ 2))
    by (rule piece-rat) (simp-all add: inner-piece-121)
    have s121: interval-lebesgue-integrable lborel (ereal 3) (ereal (640 / 37)) (λx.
R-r x /
x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal 3) (ereal (640 / 37)) (λx. R-r x / x ^
3) =
    (2733958174246260623 / 68389522650408960000)
    using chain[OF - - s120 r121] by (simp add: power2-eq-square power3-eq-cube)
    have r122: interval-lebesgue-integrable lborel (ereal (640 / 37)) (ereal (400 /
23)) (λx. R-r x / x ^ 3) ∧
    interval-lebesgue-integral lborel (ereal (640 / 37)) (ereal (400 / 23)) (λx. R-r
x /
x ^ 3) = (- (849 / 40) / (400 / 23) - (- 350) / (2 * (400 / 23) ^ 2)) -
(- (849 / 40) /
(640 / 37) - (- 350) / (2 * (640 / 37) ^ 2))

```

```

    by (rule piece-rat) (simp-all add: inner-piece-122)
  have s122: interval-lebesgue-integrable lborel (ereal 3) (ereal (400 / 23)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (400 / 23)) (λx. R-r x / x ^
3) =
(172268644150069133 / 4274345165650560000)
  using chain[OF - - s121 r122] by (simp add: power2-eq-square power3-eq-cube)
  have r123: interval-lebesgue-integrable lborel (ereal (400 / 23)) (ereal (35 /
2)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (400 / 23)) (ereal (35 / 2)) (λx. R-r x
/
x ^ 3) = (- (803 / 40) / (35 / 2) - (- 330) / (2 * (35 / 2) ^ 2)) - (-
(803 / 40) / (400 /
23) - (- 330) / (2 * (400 / 23) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-123)
  have s123: interval-lebesgue-integrable lborel (ereal 3) (ereal (35 / 2)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (35 / 2)) (λx. R-r x / x ^ 3)
=
(1218264660892033991 / 29920416159553920000)
  using chain[OF - - s122 r123] by (simp add: power2-eq-square power3-eq-cube)
  have r124: interval-lebesgue-integrable lborel (ereal (35 / 2)) (ereal (760 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (35 / 2)) (ereal (760 / 43)) (λx. R-r x
/
x ^ 3) = (- (843 / 40) / (760 / 43) - (- 695 / 2) / (2 * (760 / 43) ^
2)) - (- (843 / 40) /
(35 / 2) - (- 695 / 2) / (2 * (35 / 2) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-124)
  have s124: interval-lebesgue-integrable lborel (ereal 3) (ereal (760 / 43)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (760 / 43)) (λx. R-r x / x ^
3) =
(3366979142827744247 / 81212558147360640000)
  using chain[OF - - s123 r124] by (simp add: power2-eq-square power3-eq-cube)
  have r125: interval-lebesgue-integrable lborel (ereal (760 / 43)) (ereal (410 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (760 / 43)) (ereal (410 / 23)) (λx. R-r
x /
x ^ 3) = (- (357 / 20) / (410 / 23) - (- 581 / 2) / (2 * (410 / 23) ^
2)) - (- (357 / 20) /
(760 / 43) - (- 581 / 2) / (2 * (760 / 43) ^ 2))
  by (rule piece-rat) (simp-all add: inner-piece-125)
  have s125: interval-lebesgue-integrable lborel (ereal 3) (ereal (410 / 23)) (λx.
R-r x /
x ^ 3) ∧

```

```

interval-lebesgue-integral lborel (ereal 3) (ereal (410 / 23)) (λx. R-r x / x ^
3) =
  (303021066443711433353 / 7185174223458591360000)
using chain[OF - - s124 r125] by (simp add: power2-eq-square power3-eq-cube)
have r126: interval-lebesgue-integrable lborel (ereal (410 / 23)) (ereal (660 /
37)) (λx. R-r x / x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (410 / 23)) (ereal (660 / 37)) (λx. R-r
x /
  x ^ 3) = (- (167 / 10) / (660 / 37) - (- 270) / (2 * (660 / 37) ^ 2))
- (- (167 / 10) /
  (410 / 23) - (- 270) / (2 * (410 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-126)
have s126: interval-lebesgue-integrable lborel (ereal 3) (ereal (660 / 37)) (λx.
R-r x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (660 / 37)) (λx. R-r x / x ^
3) =
  (81409371519421251563 / 1927729669708402560000)
using chain[OF - - s125 r126] by (simp add: power2-eq-square power3-eq-cube)
have r127: interval-lebesgue-integrable lborel (ereal (660 / 37)) (ereal 18) (λx.
R-r x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal (660 / 37)) (ereal 18) (λx. R-r x / x ^
3) =
  (- (371 / 20) / 18 - (- 303) / (2 * 18 ^ 2)) - (- (371 / 20) / (660 /
37) - (- 303) / (2 *
  (660 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-127)
have s127: interval-lebesgue-integrable lborel (ereal 3) (ereal 18) (λx. R-r x /
x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal 18) (λx. R-r x / x ^ 3) =
  (2515340133869434411 / 58416050597224320000)
using chain[OF - - s126 r127] by (simp add: power2-eq-square power3-eq-cube)
have r128: interval-lebesgue-integrable lborel (ereal 18) (ereal (780 / 43)) (λx.
R-r x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 18) (ereal (780 / 43)) (λx. R-r x / x ^
3) =
  (- (177 / 10) / (780 / 43) - (- 321) / (2 * (780 / 43) ^ 2)) - (- (177
/ 10) / 18 -
  (- 321) / (2 * 18 ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-128)
have s128: interval-lebesgue-integrable lborel (ereal 3) (ereal (780 / 43)) (λx.
R-r x /
  x ^ 3) ∧
  interval-lebesgue-integral lborel (ereal 3) (ereal (780 / 43)) (λx. R-r x / x ^
3) =
  (98035230807336150829 / 2278225973291748480000)
using chain[OF - - s127 r128] by (simp add: power2-eq-square power3-eq-cube)

```

```

have r129: interval-lebesgue-integrable lborel (ereal (780 / 43)) (ereal (420 /
23)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (780 / 43)) (ereal (420 / 23)) (λx. R-r
x /
x ^ 3) = (− (177 / 10) / (420 / 23) − (− 321) / (2 * (420 / 23) ^ 2))
− (− (177 / 10) /
(780 / 43) − (− 321) / (2 * (780 / 43) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-129)
have s129: interval-lebesgue-integrable lborel (ereal 3) (ereal (420 / 23)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (420 / 23)) (λx. R-r x / x ^
3) =
(52816347568143318631 / 1226737062541710720000)
using chain[OF - - s128 r129] by (simp add: power2-eq-square power3-eq-cube)
have r130: interval-lebesgue-integrable lborel (ereal (420 / 23)) (ereal (680 /
37)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (420 / 23)) (ereal (680 / 37)) (λx. R-r
x /
x ^ 3) = (− (331 / 20) / (680 / 37) − (− 300) / (2 * (680 / 37) ^ 2))
− (− (331 / 20) /
(420 / 23) − (− 300) / (2 * (420 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-130)
have s130: interval-lebesgue-integrable lborel (ereal 3) (ereal (680 / 37)) (λx.
R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (680 / 37)) (λx. R-r x / x ^
3) =
(128449727084362846961 / 2979218580458440320000)
using chain[OF - - s129 r130] by (simp add: power2-eq-square power3-eq-cube)
have r131: interval-lebesgue-integrable lborel (ereal (680 / 37)) (ereal (37 /
2)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (680 / 37)) (ereal (37 / 2)) (λx. R-r x
/
x ^ 3) = (− (847 / 40) / (37 / 2) − (− 385) / (2 * (37 / 2) ^ 2)) − (−
(847 / 40) / (680 /
37) − (− 385) / (2 * (680 / 37) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-131)
have s131: interval-lebesgue-integrable lborel (ereal 3) (ereal (37 / 2)) (λx. R-r
x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal (37 / 2)) (λx. R-r x / x ^ 3)
=
(280252059175511707021 / 6484181616291899520000)
using chain[OF - - s130 r131] by (simp add: power2-eq-square power3-eq-cube)
have r132: interval-lebesgue-integrable lborel (ereal (37 / 2)) (ereal (800 /
43)) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (37 / 2)) (ereal (800 / 43)) (λx. R-r x
/

```

$x \wedge 3) = (- (887 / 40) / (800 / 43) - (- 807 / 2) / (2 * (800 / 43) \wedge 2)) - (- (887 / 40) / (37 / 2) - (- 807 / 2) / (2 * (37 / 2) \wedge 2))$
by (rule piece-rat) (simp-all add: inner-piece-132)
have s132: interval-lebesgue-integrable lborel (ereal 3) (ereal (800 / 43)) ($\lambda x.$ R-r $x / x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (800 / 43)) ($\lambda x.$ R-r $x / x \wedge 3) =$
(30388217670048789739 / 700992607166691840000)
using chain[OF - - s131 r132] **by** (simp add: power2-eq-square power3-eq-cube)
have r133: interval-lebesgue-integrable lborel (ereal (800 / 43)) (ereal (430 / 23)) ($\lambda x.$ R-r $x / x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal (800 / 43)) (ereal (430 / 23)) ($\lambda x.$ R-r $x /$
 $x \wedge 3) = (- (379 / 20) / (430 / 23) - (- 687 / 2) / (2 * (430 / 23) \wedge 2)) - (- (379 / 20) / (800 / 43) - (- 687 / 2) / (2 * (800 / 43) \wedge 2))$
by (rule piece-rat) (simp-all add: inner-piece-133)
have s133: interval-lebesgue-integrable lborel (ereal 3) (ereal (430 / 23)) ($\lambda x.$ R-r $x /$
 $x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (430 / 23)) ($\lambda x.$ R-r $x / x \wedge 3) =$
(14092038464086667615347 / 324033832662803303040000)
using chain[OF - - s132 r133] **by** (simp add: power2-eq-square power3-eq-cube)
have r134: interval-lebesgue-integrable lborel (ereal (430 / 23)) (ereal (700 / 37)) ($\lambda x.$ R-r $x / x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal (430 / 23)) (ereal (700 / 37)) ($\lambda x.$ R-r $x /$
 $x \wedge 3) = (- (89 / 5) / (700 / 37) - (- 322) / (2 * (700 / 37) \wedge 2)) - (- (89 / 5) / (430 / 23) - (- 322) / (2 * (430 / 23) \wedge 2))$
by (rule piece-rat) (simp-all add: inner-piece-134)
have s134: interval-lebesgue-integrable lborel (ereal 3) (ereal (700 / 37)) ($\lambda x.$ R-r $x /$
 $x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (700 / 37)) ($\lambda x.$ R-r $x / x \wedge 3) =$
(330948661396284592537 / 7535670527041937280000)
using chain[OF - - s133 r134] **by** (simp add: power2-eq-square power3-eq-cube)
have r135: interval-lebesgue-integrable lborel (ereal (700 / 37)) (ereal 19) ($\lambda x.$ R-r $x /$
 $x \wedge 3) \wedge$
interval-lebesgue-integral lborel (ereal (700 / 37)) (ereal 19) ($\lambda x.$ R-r $x / x \wedge 3) =$
 $(- (393 / 20) / 19 - (- 357) / (2 * 19 \wedge 2)) - (- (393 / 20) / (700 / 37) - (- 357) / (2 * (700 / 37) \wedge 2))$

by (rule piece-rat) (simp-all add: inner-piece-135)
have s135: interval-lebesgue-integrable lborel (ereal 3) (ereal 19) ($\lambda x. R-r x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal 19) ($\lambda x. R-r x / x^{\wedge} 3$) =
(6314515564686688895611 / 143177740013796808320000)
using chain[OF - - s134 r135] **by** (simp add: power2-eq-square power3-eq-cube)
have r136: interval-lebesgue-integrable lborel (ereal 19) (ereal (820 / 43)) ($\lambda x. R-r x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 19) (ereal (820 / 43)) ($\lambda x. R-r x / x^{\wedge} 3$) =
(- (94 / 5) / (820 / 43) - (- 376) / (2 * (820 / 43)^2)) - (- (94 / 5) / 19 - (- 376) / (2 * 19^2))
by (rule piece-rat) (simp-all add: inner-piece-136)
have s136: interval-lebesgue-integrable lborel (ereal 3) (ereal (820 / 43)) ($\lambda x. R-r x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (820 / 43)) ($\lambda x. R-r x / x^{\wedge} 3$) =
(13569347630600699631329 / 308962491608719428480000)
using chain[OF - - s135 r136] **by** (simp add: power2-eq-square power3-eq-cube)
have r137: interval-lebesgue-integrable lborel (ereal (820 / 43)) (ereal (440 / 23)) ($\lambda x. R-r x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (ereal (820 / 43)) (ereal (440 / 23)) ($\lambda x. R-r x / x^{\wedge} 3$) =
(- (94 / 5) / (440 / 23) - (- 376) / (2 * (440 / 23)^2)) -
(- (94 / 5) / (820 / 43) - (- 376) / (2 * (820 / 43)^2))
by (rule piece-rat) (simp-all add: inner-piece-137)
have s137: interval-lebesgue-integrable lborel (ereal 3) (ereal (440 / 23)) ($\lambda x. R-r x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (440 / 23)) ($\lambda x. R-r x / x^{\wedge} 3$) =
(3628346009801718502259 / 82892375797461310080000)
using chain[OF - - s136 r137] **by** (simp add: power2-eq-square power3-eq-cube)
have r138: interval-lebesgue-integrable lborel (ereal (440 / 23)) (ereal (720 / 37)) ($\lambda x. R-r x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (ereal (440 / 23)) (ereal (720 / 37)) ($\lambda x. R-r x / x^{\wedge} 3$) =
(- (353 / 20) / (720 / 37) - (- 354) / (2 * (720 / 37)^2)) -
(- (353 / 20) / (440 / 23) - (- 354) / (2 * (440 / 23)^2))
by (rule piece-rat) (simp-all add: inner-piece-138)
have s138: interval-lebesgue-integrable lborel (ereal 3) (ereal (720 / 37)) ($\lambda x. R-r x / x^{\wedge} 3$) $\wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (720 / 37)) ($\lambda x. R-r x / x^{\wedge} 3$) =

$3) =$
 $(325200031085559118469 / 7535670527041937280000)$
using chain[OF - - s137 r138] **by** (simp add: power2-eq-square power3-eq-cube)
have r139: interval-lebesgue-integrable lborel (ereal (720 / 37)) (ereal (39 / 2))
 $(\lambda x. R-r x / x^3) \wedge$
interval-lebesgue-integral lborel (ereal (720 / 37)) (ereal (39 / 2)) $(\lambda x. R-r x$
 $/$
 $x^3) = (- (891 / 40) / (39 / 2) - (- 444) / (2 * (39 / 2)^2)) - (-$
 $(891 / 40) / (720 /$
 $37) - (- 444) / (2 * (720 / 37)^2))$
by (rule piece-rat) (simp-all add: inner-piece-139)
have s139: interval-lebesgue-integrable lborel (ereal 3) (ereal (39 / 2)) $(\lambda x. R-r$
 $x /$
 $x^3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (39 / 2)) $(\lambda x. R-r x / x^3)$
 $=$
 $(4222179444545094496897 / 97963716851545184640000)$
using chain[OF - - s138 r139] **by** (simp add: power2-eq-square power3-eq-cube)
have r140: interval-lebesgue-integrable lborel (ereal (39 / 2)) (ereal (840 / 43))
 $(\lambda x. R-r x / x^3) \wedge$
interval-lebesgue-integral lborel (ereal (39 / 2)) (ereal (840 / 43)) $(\lambda x. R-r x$
 $/$
 $x^3) = (- (931 / 40) / (840 / 43) - (- 927 / 2) / (2 * (840 / 43)^2)) - (-$
 $(931 / 40) /$
 $(39 / 2) - (- 927 / 2) / (2 * (39 / 2)^2))$
by (rule piece-rat) (simp-all add: inner-piece-140)
have s140: interval-lebesgue-integrable lborel (ereal 3) (ereal (840 / 43)) $(\lambda x.$
 $R-r x /$
 $x^3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (840 / 43)) $(\lambda x. R-r x / x^3)$
 $=$
 $(2271196345609880684383 / 52749693689293560960000)$
using chain[OF - - s139 r140] **by** (simp add: power2-eq-square power3-eq-cube)
have r141: interval-lebesgue-integrable lborel (ereal (840 / 43)) (ereal (450 / 23))
 $(\lambda x. R-r x / x^3) \wedge$
interval-lebesgue-integral lborel (ereal (840 / 43)) (ereal (450 / 23)) $(\lambda x. R-r$
 $x /$
 $x^3) = (- (401 / 20) / (450 / 23) - (- 801 / 2) / (2 * (450 / 23)^2)) - (-$
 $(401 / 20) /$
 $(840 / 43) - (- 801 / 2) / (2 * (840 / 43)^2))$
by (rule piece-rat) (simp-all add: inner-piece-141)
have s141: interval-lebesgue-integrable lborel (ereal 3) (ereal (450 / 23)) $(\lambda x.$
 $R-r x /$
 $x^3) \wedge$
interval-lebesgue-integral lborel (ereal 3) (ereal (450 / 23)) $(\lambda x. R-r x / x^3)$
 $=$
 $(324195926757243309661 / 7535670527041937280000)$
using chain[OF - - s140 r141] **by** (simp add: power2-eq-square power3-eq-cube)
have r142: interval-lebesgue-integrable lborel (ereal (450 / 23)) (ereal 20) $(\lambda x.$

```

R-r x /
x ^ 3) ∧
interval-lebesgue-integral lborel (ereal (450 / 23)) (ereal 20) (λx. R-r x / x ^
3) =
(- (189 / 10) / 20 - (- 378) / (2 * 20 ^ 2)) - (- (189 / 10) / (450 /
23) - (- 378) / (2 *
(450 / 23) ^ 2))
by (rule piece-rat) (simp-all add: inner-piece-142)
have interval-lebesgue-integrable lborel (ereal 3) (ereal 20) (λx. R-r x / x ^ 3) ∧
interval-lebesgue-integral lborel (ereal 3) (ereal 20) (λx. R-r x / x ^ 3) =
(322437603634266857629 / 7535670527041937280000)
using chain[OF - s141 r142] by (simp add: power2-eq-square power3-eq-cube)
thus ?thesis by simp
qed

```

26.5 Integrability on $[3, \infty)$ and the tails (5.14)-(5.17)

The exceptional set $S = \mathbb{Z} \cup (40/3)\mathbb{Z}$ where the fractional parts jump.

definition $Sx :: \text{real set}$ **where** $Sx = \{x. x \in \mathbb{Z} \vee \text{alpha-r} * x \in \mathbb{Z}\}$

(5.15): $\Phi(x) = (P(x) - 2923/240)/x^2 + \lambda/x + 2C(x)/x^3$.

definition $\Phi :: \text{real} \Rightarrow \text{real}$ **where**

$\Phi x = (Pper x - 2923 / 240) / x^2 + lam-r / x + 2 * Cper x / x^3$

lemma *frac-measurable* [*measurable*]: (*frac* :: *real* $\Rightarrow$ *real*) $\in$ *borel-measurable borel*
proof –

have $e: (\text{frac} :: \text{real} \Rightarrow \text{real}) = (\lambda x. x - \text{real-of-int } \lfloor x \rfloor)$

using *frac-def fun-eq-iff* **by** *blast*

show ?thesis **unfolding** e **using** *borel-measurable-real-floor* **by** *simp*

qed

lemma *es-measurable* [*measurable*]: $es \in \text{borel-measurable borel}$

proof –

have $e: es = (\lambda u. \text{if } u \leq 1 / 2 \text{ then } 1 \text{ else } - 1)$ **by** (*simp add: fun-eq-iff es-def*)

show ?thesis **unfolding** e **by** *measurable*

qed

lemma *d0-measurable* [*measurable*]: $d0 \in \text{borel-measurable borel}$

proof –

have $e: d0 = (\lambda u. \min u (1 - u))$ **by** (*simp add: fun-eq-iff d0-def*)

show ?thesis **unfolding** e **by** *measurable*

qed

lemma *pos-measurable* [*measurable*]: $pos-r \in \text{borel-measurable borel}$

proof –

have $e: pos-r = (\lambda u. \max u 0)$ **by** (*simp add: fun-eq-iff pos-r-def*)

show ?thesis **unfolding** e **by** *measurable*

qed

lemma *R-measurable* [*measurable*]: $R-r \in \text{borel-measurable borel}$
proof –
 have $e: R-r = (\lambda x. x * Fx x + Qc (\text{frac } x) (\text{frac } (\text{alpha-r } * x)) (\text{frac } (2 * Hc-r * x)) (\text{frac } (2 * x)) (\text{frac } (2 * (\text{lam-r } * x))))$
 by (*simp add: fun-eq-iff R-eq*)
 show ?thesis **unfolding** *e Fx-def Qc-def* **by** *measurable*
qed

lemma *Fx-bounds*: $- 37 / 5 \leq Fx x \wedge Fx x \leq 111 / 20$
unfolding *Fx-def lam-r-def* **using** *frac-ge-0[of x] frac-lt-1[of x] frac-ge-0[of alpha-r * x] frac-lt-1[of alpha-r * x]* **by** *linarith*

lemma *Q-bounds*: $- 1 / 2 \leq R-r x - x * Fx x \wedge R-r x - x * Fx x \leq 13 / 8$
unfolding *Fx-def* **by** (*rule R-decomp-bound-all*)

lemma *R-abs-le*: $3 \leq x \implies |R-r x| \leq 9 * x$
proof –
 assume $x: 3 \leq x$
 have $F: - 37 / 5 \leq Fx x \wedge Fx x \leq 111 / 20$ **by** (*rule Fx-bounds*)
 have $- 1 / 2 \leq R-r x - x * Fx x \wedge R-r x - x * Fx x \leq 13 / 8$ **by** (*rule Q-bounds*)
 moreover have $x * (- 37 / 5) \leq x * Fx x$ **using** *F x* **by** (*fastforce intro: mult-left-mono*)
 moreover have $x * Fx x \leq x * (111 / 20)$ **using** *F x* **by** (*auto intro: mult-left-mono*)
 ultimately show ?thesis **using** *x abs-le-iff* **by** *linarith*
qed

lemma *R-div-bound*: $3 \leq x \implies |R-r x / x^3| \leq 9 / x^2$
proof –
 assume $x: 3 \leq x$
 have $p: 0 < x^3$ **using** *x* **by** *simp*
 have $|R-r x / x^3| = |R-r x| / x^3$ **using** *p* **by** (*simp add: abs-divide*)
 moreover have $|R-r x| / x^3 \leq 9 * x / x^3$ **using** *R-abs-le[OF x] p*
 by (*fastforce intro: divide-right-mono*)
 moreover have $9 * x / x^3 = 9 / x^2$ **using** *x* **by** (*simp add: power3-eq-cube power2-eq-square*)
 ultimately show ?thesis **by** *linarith*
qed

lemma *inv-sq-integrable*: *set-integrable lborel {3..} ($\lambda x :: \text{real}. 9 / x^2$)*
proof –
 have $((\lambda x :: \text{real}. 1 / x^2) \text{ has-integral } 1 / (\text{real } (2 - 1) * 3^{(2 - 1)})) \{3..\}$
using *has-integral-inverse-power-to-inf[of 2 3]* **by** *simp*
 hence $i0: (\lambda x :: \text{real}. 1 / x^2) \text{ integrable-on } \{3..\}$
by (*blast intro: has-integral-integrable*)
 have $i: (\lambda x :: \text{real}. 9 * (1 / x^2)) \text{ integrable-on } \{3..\}$
using *integrable-on-cmult-left[OF i0, of 9]* **by** *simp*
 have $a: (\lambda x :: \text{real}. 9 * (1 / x^2)) \text{ absolutely-integrable-on } \{3..\}$

```

    using nonnegative-absolutely-integrable-1[OF i] by simp
  have m: ( $\lambda x::\text{real}.$  indicator {3..} x *R (9 * (1 / x ^ 2)))  $\in$ 
    borel-measurable lborel by measurable
  have integrable lborel ( $\lambda x::\text{real}.$  indicator {3..} x *R (9 * (1 / x ^ 2)))
    using a integrable-completion[OF m] unfolding set-integrable-def by blast
  thus ?thesis unfolding set-integrable-def by simp
qed

```

```

theorem R-div-cube-integrable: set-integrable lborel {3..} ( $\lambda x.$  R-r x / x ^ 3)
proof -
  have m: set-borel-measurable lborel {3..} ( $\lambda x.$  R-r x / x ^ 3)
    unfolding set-borel-measurable-def by measurable
  have ae: AE x in lborel. x  $\in$  {3..}  $\longrightarrow$ 
    norm (R-r x / x ^ 3)  $\leq$  norm (9 / x ^ 2)
  proof (rule AE-I2)
    fix x :: real
    show x  $\in$  {3..}  $\longrightarrow$  norm (R-r x / x ^ 3)  $\leq$  norm (9 / x ^ 2)
      using R-div-bound[of x]
      by simp
  qed
  show ?thesis by (rule set-integrable-bound[OF inv-sq-integrable m ae])
qed

```

Derivatives of the fractional part off the integers.

```

lemma frac-has-derivative:  $x \notin \mathbb{Z} \implies (\text{frac has-real-derivative } 1) (at\ x)$ 
proof -
  assume x:  $x \notin \mathbb{Z}$ 
  have e: frac = ( $\lambda x::\text{real}.$  x - real-of-int [x])
    using frac-def fun-eq-iff by blast
  have fl: (( $\lambda x.$  real-of-int [id x]) has-real-derivative 0) (at x)
    using floor-has-real-derivative[where f = id and x = x] x by auto
  show ?thesis unfolding e using DERIV-diff[OF DERIV-ident fl] by force
qed

```

```

lemma frac-scaled-has-derivative:  $c * x \notin \mathbb{Z} \implies$ 
  (( $\lambda x.$  frac (c * x)) has-real-derivative c) (at x)
proof -
  assume h:  $c * x \notin \mathbb{Z}$ 
  have d: (frac has-real-derivative 1) (at (c * x)) by (rule frac-has-derivative[OF h])
  show ?thesis using DERIV-chain2[OF d DERIV-cmult-Id[of c x]] by force
qed

```

```

lemma Pper-deriv:  $x \notin Sx \implies (Pper \text{ has-real-derivative } (Fx\ x + \text{lam-r})) (at\ x)$ 
proof -
  assume x:  $x \notin Sx$ 
  have n1:  $x \notin \mathbb{Z}$  and n2:  $\alpha\text{-r} * x \notin \mathbb{Z}$  using x
    by (simp-all add: Sx-def)
  have d1: (frac has-real-derivative 1) (at x) by (rule frac-has-derivative[OF n1])

```

have $d2$: $((\lambda x. \text{frac} (\text{alpha-r} * x)) \text{ has-real-derivative } \text{alpha-r}) (at\ x)$
by $(rule\ \text{frac-scaled-has-derivative}[OF\ n2])$
have e : $Pper = (\lambda x. \gamma_4 * \text{frac} (\text{alpha-r} * x) * (1 - \text{frac} (\text{alpha-r} * x)) - \text{lam-r} * \text{frac}\ x * (1 - \text{frac}\ x))$
by $(simp\ add: Pper-def\ fun-eq-iff)$
note $D = \text{DERIV-diff}[OF\ \text{DERIV-mult}[OF\ \text{DERIV-cmult}[\text{where } c = \gamma_4, OF\ d2]\ \text{DERIV-diff}[OF\ \text{DERIV-const}[\text{where } k = 1]\ d2]]$
 $\text{DERIV-mult}[OF\ \text{DERIV-cmult}[\text{where } c = \text{lam-r}, OF\ d1]\ \text{DERIV-diff}[OF\ \text{DERIV-const}[\text{where } k = 1]\ d1]]]$
have v : $\gamma_4 * \text{alpha-r} * (1 - \text{frac} (\text{alpha-r} * x)) + (0 - \text{alpha-r}) * (\gamma_4 * \text{frac} (\text{alpha-r} * x)) - (\text{lam-r} * 1 * (1 - \text{frac}\ x) + (0 - 1) * (\text{lam-r} * \text{frac}\ x)) = Fx\ x + \text{lam-r}$
unfolding $Fx-def\ \text{alpha-r-def}\ \text{lam-r-def}$ **using** algebra-simps **by** argo
show $?thesis$ **unfolding** e **using** $\text{DERIV-cong}[OF\ D\ v]$ **by** blast
qed

lemma $G0-eq$: $G0-r = (\lambda v. v * (1 - v) * (2 * v - 1) / 6)$
by $(simp\ add: fun-eq-iff\ G0-r-def)$

lemma $G0-deriv$:
 $(G0-r \text{ has-real-derivative } (v * (1 - v) - 1 / 6)) (at\ v)$
proof –
note $D = \text{DERIV-cdivide}[OF\ \text{DERIV-mult}[OF\ \text{DERIV-mult}[OF\ \text{DERIV-ident}\ \text{DERIV-diff}[OF\ \text{DERIV-const}[\text{where } k = 1]\ \text{DERIV-ident}]]\ \text{DERIV-diff}[OF\ \text{DERIV-cmult}[\text{where } c = 2, OF\ \text{DERIV-ident}]\ \text{DERIV-const}[\text{where } k = 1]]], \text{where } c = 6]$
have v : $((1 * (1 - v) + (0 - 1) * v) * (2 * v - 1) + (2 * 1 - 0) * (v * (1 - v))) / 6 = v * (1 - v) - 1 / 6$
using algebra-simps **by** argo
show $?thesis$ **unfolding** $G0-eq$ **using** $\text{DERIV-cong}[OF\ D\ v]$ **by** blast
qed

lemma $Cper-deriv$: $x \notin Sx \implies (Cper \text{ has-real-derivative } (Pper\ x - 2923 / 240)) (at\ x)$
proof –
assume x : $x \notin Sx$
have $n1$: $x \notin \mathbb{Z}$ **and** $n2$: $\text{alpha-r} * x \notin \mathbb{Z}$ **using** x
by $(simp-all\ add: Sx-def)$
have $d1$: $(\text{frac}\ \text{has-real-derivative}\ 1) (at\ x)$ **by** $(rule\ \text{frac-has-derivative}[OF\ n1])$
have $d2$: $((\lambda x. \text{frac} (\text{alpha-r} * x)) \text{ has-real-derivative } \text{alpha-r}) (at\ x)$
by $(rule\ \text{frac-scaled-has-derivative}[OF\ n2])$
have $g1$: $((\lambda x. G0-r (\text{frac}\ x)) \text{ has-real-derivative } ((\text{frac}\ x * (1 - \text{frac}\ x) - 1 / 6) * 1)) (at\ x)$

```

    by (rule DERIV-chain2[OF G0-deriv d1])
  have g2: (( $\lambda x. G0-r \text{ (frac (alpha-r * x))}$ ) has-real-derivative (( $\text{frac (alpha-r * x)}$ 
* (1 -
   $\text{frac (alpha-r * x)} - 1 / 6$ ) * alpha-r)) (at x)
    by (rule DERIV-chain2[OF G0-deriv d2])
  have e: Cper = ( $\lambda x. 74 / \text{alpha-r} * G0-r \text{ (frac (alpha-r * x))} - \text{lam-r} * G0-r$ 
( $\text{frac x}$ ))
    by (simp add: fun-eq-iff Cper-def)
  note D = DERIV-diff[OF DERIV-cmult[where c =  $74 / \text{alpha-r}$ , OF g2] DE-
RIV-cmult[where c =
    lam-r, OF g1]]
  have v:  $74 / \text{alpha-r} * ((\text{frac (alpha-r * x)} * (1 - \text{frac (alpha-r * x)}) - 1 / 6)$ 
* alpha-r) -
    lam-r * (( $\text{frac x} * (1 - \text{frac x}) - 1 / 6$ ) * 1) = Pper x -  $2923 / 240$ 
    unfolding Pper-def alpha-r-def lam-r-def using algebra-simps by argo
  show ?thesis unfolding e using DERIV-cong[OF D v] by blast
qed

```

lemma comp-frac-piece:

```

  fixes g :: real  $\Rightarrow$  real and m :: int
  assumes continuous-on {0..1} g g 0 = g 1
  shows continuous-on {of-int m..of-int m + 1} ( $\lambda x. g \text{ (frac x)}$ )
proof -
  have c: continuous-on {of-int m..of-int m + 1} ( $\lambda y. g \text{ (y - of-int m)}$ )
  proof (rule continuous-on-compose2[OF assms(1)])
    show continuous-on {of-int m..of-int m + 1} ( $\lambda y::\text{real}. y - \text{of-int m}$ )
      by (auto intro: continuous-intros)
    show ( $\lambda y::\text{real}. y - \text{of-int m}$ ) ' {of-int m..of-int m + 1}  $\subseteq$  {0..1}
      by (auto simp: image-subset-iff)
  qed
  have e:  $g \text{ (frac y)} = g \text{ (y - of-int m)}$  if y:  $y \in \{\text{of-int m..of-int m} + 1\}$  for y
  proof (cases y = of-int m + 1)
    case True
      have f:  $\text{frac y} = 0$  unfolding True by simp
      show ?thesis unfolding f True using assms(2) by simp
    next
      case False
      have f:  $\text{frac y} = y - \text{of-int m}$  using y False
        by (simp add: frac-def floor-eq-iff)
      show ?thesis unfolding f ..
  qed
  show ?thesis using continuous-on-cong[OF refl e] c by fast
qed

```

lemma comp-frac-isCont:

```

  fixes g :: real  $\Rightarrow$  real
  assumes continuous-on {0..1} g g 0 = g 1
  shows isCont ( $\lambda x. g \text{ (frac x)}$ ) x
proof -

```

```

define  $n$  where  $n = \lfloor x \rfloor$ 
have  $c1$ : continuous-on  $\{of-int\ (n - 1)..of-int\ (n - 1) + 1\}$   $(\lambda x. g\ (frac\ x))$ 
  by (rule comp-frac-piece[OF assms])
have  $c2$ : continuous-on  $\{of-int\ n..of-int\ n + 1\}$   $(\lambda x. g\ (frac\ x))$ 
  by (rule comp-frac-piece[OF assms])
have  $u$ :  $\{of-int\ (n - 1)..of-int\ (n - 1) + 1\} \cup \{of-int\ n..of-int\ n + 1\} =$ 
   $\{of-int\ n - 1..of-int\ n + (1::real)\}$ 
  by fastforce
have  $c$ : continuous-on  $\{of-int\ n - 1..of-int\ n + (1::real)\}$   $(\lambda x. g\ (frac\ x))$ 
  using continuous-on-closed-Un[OF closed-atLeastAtMost closed-atLeastAtMost
c1 c2] unfolding  $u$  .
  have  $of-int\ n - 1 < x$  unfolding  $n$ -def using of-int-floor-le[of x] by linarith
  moreover have  $x < of-int\ n + 1$  unfolding  $n$ -def using real-of-int-floor-add-one-gt[of
x]
    by linarith
  ultimately have  $xi$ :  $x \in interior\ \{of-int\ n - 1..of-int\ n + (1::real)\}$  by simp
  show ?thesis by (rule continuous-on-interior[OF c xi])
qed

```

$g \circ frac$ is continuous when g is continuous on $[0,1]$ with $g\ 0 = g\ 1$ (Lean *ContinuousOn.comp-fract''*).

```

lemma comp-frac-cont:
  fixes  $g :: real \Rightarrow real$ 
  assumes continuous-on  $\{0..1\}$   $g\ g\ 0 = g\ 1$ 
  shows continuous-on UNIV  $(\lambda x. g\ (frac\ x))$ 
proof -
  have isCont  $(\lambda x. g\ (frac\ x))\ x$  for  $x$ 
  proof (cases  $x \in \mathbb{Z}$ )
    case False
    thus ?thesis
      using comp-frac-isCont[OF assms, of x] by blast
  next
    case True
    show ?thesis
      using comp-frac-isCont[OF assms, of x] by blast
  qed
  thus ?thesis by (simp add: continuous-at-imp-continuous-on)
qed

```

```

lemma u-cont: continuous-on UNIV  $(\lambda x::real. frac\ x * (1 - frac\ x))$ 
proof -
  have  $c$ : continuous-on  $\{0..1\}$   $(\lambda v::real. v * (1 - v))$ 
    by (fastforce intro: continuous-intros)
  show ?thesis using comp-frac-cont[OF c] by simp
qed

```

```

lemma u-cont-a: continuous-on UNIV  $(\lambda x. frac\ (alpha-r * x) * (1 - frac\ (alpha-r$ 
 $* x)))$ 
proof -

```

have c : *continuous-on UNIV* $(\lambda x. \text{alpha-r} * x)$ **by** *(auto intro: continuous-intros)*
show *?thesis* **using** *continuous-on-compose2[OF u-cont c]* **by** *blast*
qed

lemma *Pper-cont: continuous-on UNIV Pper*
proof –
have e : $Pper = (\lambda x. 74 * (\text{frac} (\text{alpha-r} * x) * (1 - \text{frac} (\text{alpha-r} * x)))) - \text{lam-r}$
 $*$
 $(\text{frac } x * (1 - \text{frac } x))$
unfolding *fun-eq-iff Pper-def* **by** *(simp add: mult.assoc)*
show *?thesis* **unfolding** e **using** *u-cont u-cont-a* **by** *(fast intro: continuous-intros)*
qed

lemma *G0-cont: continuous-on {0..1} G0-r*
unfolding *G0-eq* **by** *(fastforce intro: continuous-intros)*

lemma *G0-frac-cont: continuous-on UNIV* $(\lambda x. G0-r (\text{frac } x))$
using *comp-frac-cont[OF G0-cont]* **by** *(simp add: G0-r-def)*

lemma *Cper-cont: continuous-on UNIV Cper*
proof –
have c : *continuous-on UNIV* $(\lambda x. \text{alpha-r} * x)$ **by** *(auto intro: continuous-intros)*
have a : *continuous-on UNIV* $(\lambda x. G0-r (\text{frac} (\text{alpha-r} * x)))$
using *continuous-on-compose2[OF G0-frac-cont c]* **by** *blast*
have e : $Cper = (\lambda x. 74 / \text{alpha-r} * G0-r (\text{frac} (\text{alpha-r} * x)) - \text{lam-r} * G0-r$
 $(\text{frac } x))$
by *(simp add: fun-eq-iff Cper-def)*
show *?thesis* **unfolding** e **using** *a G0-frac-cont* **by** *(fast intro: continuous-intros)*
qed

lemma *uq-bounds: $0 \leq (v::\text{real}) \implies v \leq 1 \implies$*
 $0 \leq v * (1 - v) \wedge v * (1 - v) \leq 1 / 4$
proof –
assume v : $0 \leq v \wedge v \leq 1$
have $0 \leq v * (1 - v)$ **using** v **by** *simp*
moreover **have** $v * (1 - v) = 1 / 4 - (v - 1 / 2) ^ 2$
by *(simp add: power2-eq-square algebra-simps)*
moreover **have** $0 \leq (v - 1 / 2) ^ 2$ **by** *simp*
ultimately show *?thesis* **by** *linarith*
qed

lemma *Pper-abs-le: $|Pper\ x - 2923 / 240| \leq 32$*
proof –
have e : $Pper\ x = 74 * (\text{frac} (\text{alpha-r} * x) * (1 - \text{frac} (\text{alpha-r} * x))) - \text{lam-r}$
 $*$ $(\text{frac } x * (1 -$
 $\text{frac } x))$
unfolding *Pper-def* **by** *(simp add: mult.assoc)*
have $0 \leq \text{frac} (\text{alpha-r} * x) * (1 - \text{frac} (\text{alpha-r} * x)) \wedge$
 $\text{frac} (\text{alpha-r} * x) * (1 - \text{frac} (\text{alpha-r} * x)) \leq 1 / 4$

by (rule uq-bounds) (simp-all add: frac-lt-1 less-imp-le)
 moreover have $0 \leq \text{frac } x * (1 - \text{frac } x) \wedge \text{frac } x * (1 - \text{frac } x) \leq 1 / 4$
 by (rule uq-bounds) (simp-all add: frac-lt-1 less-imp-le)
 ultimately show ?thesis unfolding e lam-r-def by linarith
 qed

lemma *Phi-deriv*: $0 < x \implies x \notin Sx \implies$
 (Phi has-real-derivative (Fx x / x ^ 2 - 6 * Cper x / x ^ 4)) (at x)
proof –
 assume x0: $0 < x$ and x: $x \notin Sx$
 have nz: $x \neq 0$ and n2: $x^2 \neq 0$ and n3: $x^3 \neq 0$ using x0 by simp-all
 have e: $\text{Phi} = (\lambda x. (Pper\ x - 2923 / 240) / x^2 + \text{lam-r} / x + 2 * Cper\ x / x^3)$
 by (simp add: fun-eq-iff Phi-def)
 note D = DERIV-add[OF DERIV-add[OF DERIV-divide[OF DERIV-diff[OF Pper-deriv[OF x] DERIV-const[where k = 2923 / 240]] DERIV-pow[of 2] n2] DERIV-divide[OF DERIV-const[where k = lam-r] DERIV-ident nz]] DERIV-divide[OF DERIV-cmult[where c = 2, OF Cper-deriv[OF x] DERIV-pow[of 3] n3]]
 have v: $((Fx\ x + \text{lam-r} - 0) * x^2 - (Pper\ x - 2923 / 240) * (\text{real } 2 * x^2 - \text{Suc } 0))) / ((x^2 * x^2) + (0 * x - \text{lam-r} * 1) / (x * x) + (2 * (Pper\ x - 2923 / 240) * x^3 - 2 * Cper\ x * (\text{real } 3 * x^3 - \text{Suc } 0))) / (x^3 * x^3) = Fx\ x / x^2 - 6 * Cper\ x / x^4$
 using nz by (simp add: field-simps eval-nat-numeral)
 show ?thesis unfolding e using DERIV-cong[OF D v] by blast
 qed

lemma *Phi-cont*: $0 < a \implies \text{continuous-on } \{a..b\} \text{ Phi}$
proof –
 assume a: $0 < a$
 have e: $\text{Phi} = (\lambda x. (Pper\ x - 2923 / 240) / x^2 + \text{lam-r} / x + 2 * Cper\ x / x^3)$
 by (simp add: fun-eq-iff Phi-def)
 have P: continuous-on {a..b} Pper using continuous-on-subset[OF Pper-cont]
 by blast
 have C: continuous-on {a..b} Cper using continuous-on-subset[OF Cper-cont]
 by blast
 have nz1: $\forall x \in \{a..b\}. x \neq 0$ and nz2: $\forall x \in \{a..b\}. x^2 \neq 0$ and nz3: $\forall x \in \{a..b\}. x^3 \neq 0$ using a by auto
 have continuous-on {a..b} $(\lambda x. (Pper\ x - 2923 / 240) / x^2)$ using P nz2
 by (fast intro: continuous-on-divide continuous-on-diff continuous-on-const continuous-on-power continuous-on-id)
 moreover have continuous-on {a..b} $(\lambda x. \text{lam-r} / x)$ using nz1
 by (fast intro: continuous-on-divide continuous-on-const continuous-on-id)
 moreover have continuous-on {a..b} $(\lambda x. 2 * Cper\ x / x^3)$ using C nz3

by (fast intro: continuous-on-divide continuous-on-mult continuous-on-const
continuous-on-power
continuous-on-id)
ultimately show ?thesis unfolding e
by (fast intro: continuous-on-add)
qed

lemma Phi-abs-le: $1 \leq x \implies |Phi\ x| \leq 65 / x$

proof -

assume x: $1 \leq x$
have x0: $0 < x$ using x by simp
have xx: $x \leq x^2$ using x by (simp add: power2-eq-square)
have xxx: $x \leq x^3$ using x power-increasing[of 1 3 x]
by simp
have p2: $0 < x^2$ and p3: $0 < x^3$ using x0 by simp-all
have a1a: $|(Pper\ x - 2923 / 240) / x^2| = |Pper\ x - 2923 / 240| / x^2$
using p2 by (simp add: abs-divide)
have a1b: $|Pper\ x - 2923 / 240| / x^2 \leq 32 / x^2$ using Pper-abs-le[of x]
p2
by (force intro: divide-right-mono)
have a1c: $32 / x^2 \leq 32 / x$ using divide-left-mono[OF xx, of 32] x0 p2 by
simp
have |lam-r / x| = lam-r / x using x0
by (simp add: lam-r-def)
moreover have lam-r / x $\leq 1 / x$ using x0 divide-right-mono[of lam-r 1 x]
by (simp add: lam-r-def)
ultimately have a2: $|lam-r / x| \leq 1 / x$ by linarith
have a3a: $|2 * Cper\ x / x^3| = 2 * |Cper\ x| / x^3$ using p3
by (simp add: abs-divide abs-mult)
have a3b: $2 * |Cper\ x| / x^3 \leq 32 / x^3$ using Cper-bound[of x] p3
by (force intro: divide-right-mono)
have a3c: $32 / x^3 \leq 32 / x$ using divide-left-mono[OF xxx, of 32] x0 p3 by
simp
have t: $|(Pper\ x - 2923 / 240) / x^2 + lam-r / x + 2 * Cper\ x / x^3| \leq$
 $|(Pper\ x - 2923 / 240) / x^2| + |lam-r / x| + |2 * Cper\ x /$
 $x^3|$
by argo
have s: $32 / x + 1 / x + 32 / x = 65 / x$ by (simp add: add-divide-distrib[symmetric])
show ?thesis unfolding Phi-def using t a1a a1b a1c a2 a3a a3b a3c s by linarith
qed

lemma Sx-finite: finite (Sx $\cap$ {M..N})

proof -

have s: Sx $\cap$ {M..N} $\subseteq$
(of-int ' { $\lceil M \rceil$.. $\lfloor N \rfloor$ }) $\cup$ ((λn . of-int n /
alpha-r) ' { $\lceil alpha-r * M \rceil$.. $\lfloor alpha-r * N \rfloor$ })

proof

fix x assume x: $x \in Sx \cap \{M..N\}$
have xm: $M \leq x \leq N$ using x by auto


```

show  $x \in (\text{of-int } \{ \lceil M \rceil .. \lfloor N \rfloor \}) \cup ((\lambda n. \text{of-int } n / \text{alpha-r}) \{ \lceil \text{alpha-r} * M \rceil .. \lfloor \text{alpha-r} * N \rfloor \})$ 
proof (cases  $x \in \mathbb{Z}$ )
  case True
    then obtain  $n$  where  $n: x = \text{of-int } n$  by (auto elim: Ints-cases)
    have  $n \in \{ \lceil M \rceil .. \lfloor N \rfloor \}$  using  $xm$  unfolding  $n$ 
      by (simp add: ceiling-le-iff le-floor-iff)
    thus ?thesis unfolding  $n$  by blast
  next
    case False
    hence  $\text{alpha-r} * x \in \mathbb{Z}$  using  $x$  by (simp add: Sx-def)
    then obtain  $n$  where  $n: \text{alpha-r} * x = \text{of-int } n$  by (auto elim: Ints-cases)
    have  $x: x = \text{of-int } n / \text{alpha-r}$  using  $n$  by (simp add: alpha-r-def field-simps)
    have  $\text{alpha-r} * M \leq \text{alpha-r} * x$  using  $xm$  by (simp add: alpha-r-def)
    moreover have  $\text{alpha-r} * x \leq \text{alpha-r} * N$  using  $xm$  by (simp add: alpha-r-def)
    ultimately have  $n \in \{ \lceil \text{alpha-r} * M \rceil .. \lfloor \text{alpha-r} * N \rfloor \}$ 
      unfolding  $n$  by (simp add: ceiling-le-iff le-floor-iff)
    thus ?thesis unfolding  $x$  by blast
  qed
qed
show ?thesis using finite-subset[OF s] by blast
qed

```

```

lemma set-integrable-bounded:
  fixes  $f :: \text{real} \Rightarrow \text{real}$ 
  assumes  $A: A \in \text{sets } \text{l borel } \text{emeasure } \text{l borel } A < \infty$  and  $f: f \in \text{borel-measurable } \text{l borel}$ 
  and  $b: \bigwedge x. x \in A \implies |f x| \leq Bd$ 
  shows set-integrable l borel A f
proof -
  have  $m: f \in \text{borel-measurable } \text{l borel}$  using  $f$ 
    by simp
  have  $ae: AE\ x\ in\ \text{l borel}. x \in A \longrightarrow \text{norm } (f x) \leq Bd$ 
proof (rule AE-I2)
    fix  $x$  show  $x \in A \longrightarrow \text{norm } (f x) \leq Bd$  using  $b[\text{of } x]$ 
      by simp
  qed
show ?thesis unfolding set-integrable-def
  using integrableI-bounded-set-indicator[OF A(1) m A(2) ae]
  by fast
qed

```

```

lemma interval-integrable-bounded:
  fixes  $f :: \text{real} \Rightarrow \text{real}$ 
  assumes  $a \leq b$   $f \in \text{borel-measurable } \text{l borel}$   $\bigwedge x. x \in \{a..b\} \implies |f x| \leq Bd$ 
  shows interval-lebesgue-integrable l borel (ereal a) (ereal b) f
proof -

```

```

have set-integrable lborel {a<..<b} f
proof (rule set-integrable-bounded[OF - - assms(2)])
  show {a<..<b} ∈ sets lborel
    by simp
  show emeasure lborel {a<..<b} < ∞ using assms(1)
    by simp
  show  $\bigwedge x. x \in \{a<..<b\} \implies |f\ x| \leq Bd$  using assms(3)
    by simp
qed
thus ?thesis unfolding interval-lebesgue-integrable-def using assms(1)
  by simp
qed

```

Measurability, bounds and integrability of the two parts of the tail integrand.

```

lemma Fx-measurable [measurable]: Fx ∈ borel-measurable borel
proof -
  have e: Fx = (λx. 4 * lam-r + 2 * lam-r * frac x - 12 * lam-r * frac (alpha-r
    * x))
    by (simp add: fun-eq-iff Fx-def)
  show ?thesis unfolding e by measurable
qed

```

```

lemma Cper-measurable [measurable]: Cper ∈ borel-measurable borel
using borel-measurable-continuous-onI[OF Cper-cont]
by blast

```

```

lemma abs-div-ge1:
  fixes a d B :: real
  assumes d: 1 ≤ d and a: |a| ≤ B
  shows |a / d| ≤ B
proof -
  have |a / d| = |a| / d using d
    by (simp add: abs-divide)
  moreover have |a| / d ≤ |a|
    using divide-left-mono[of 1 d |a|] d
    by linarith
  ultimately show ?thesis using a by linarith
qed

```

```

lemma g-bound: 3 ≤ x  $\implies |F\ x\ /\ x^2 - 6 * Cper\ x\ /\ x^4| \leq 104$ 
proof -
  assume x: 3 ≤ x
  have x2: 1 ≤ x ^ 2 using x by (simp add: one-le-power)
  have x4: 1 ≤ x ^ 4 using x by (simp add: one-le-power)
  have F: |Fx x| ≤ 8 using Fx-bounds[of x]
    by linarith
  have C: |6 * Cper x| ≤ 96 using Cper-bound[of x]
    by linarith
  have |Fx x /\ x ^ 2| ≤ 8 by (rule abs-div-ge1[OF x2 F])

```

moreover have $|6 * Cper\ x / x^4| \leq 96$ **by** (rule abs-div-ge1[OF x4 C])
ultimately show ?thesis **using** abs-triangle-ineq4[of Fx x / x^2 6 * Cper x / x^4]
by linarith
qed

lemma h-bound: $3 \leq x \implies$
 $|6 * Cper\ x / x^4 + (R-r\ x - x * Fx\ x) / x^3| \leq 98$
proof –
assume $x: 3 \leq x$
have x3: $1 \leq x^3$ **using** x **by** (simp add: one-le-power)
have x4: $1 \leq x^4$ **using** x **by** (simp add: one-le-power)
have Q: $|R-r\ x - x * Fx\ x| \leq 2$ **using** Q-bounds[of x]
by linarith
have C: $|6 * Cper\ x| \leq 96$ **using** Cper-bound[of x]
by linarith
have $|6 * Cper\ x / x^4| \leq 96$ **by** (rule abs-div-ge1[OF x4 C])
moreover have $|(R-r\ x - x * Fx\ x) / x^3| \leq 2$ **by** (rule abs-div-ge1[OF x3 Q])
ultimately show ?thesis **using** abs-triangle-ineq[of 6 * Cper x / x^4 (R-r x - x * Fx x) / x^3]
by argo
qed

lemma g-meas: $(\lambda x. Fx\ x / x^2 - 6 * Cper\ x / x^4) \in \text{borel-measurable borel}$
by measurable
lemma h-meas: $(\lambda x. 6 * Cper\ x / x^4 + (R-r\ x - x * Fx\ x) / x^3) \in$
 $\text{borel-measurable borel}$ **by** measurable

lemma g-integrable:
assumes $M: 3 \leq M$ **and** $MN: M \leq N$
shows interval-lebesgue-integrable lborel (ereal M) (ereal N) $(\lambda x. Fx\ x / x^2 - 6 * Cper\ x / x^4)$
proof (rule interval-integrable-bounded[OF MN g-meas])
fix x **assume** $x \in \{M..N\}$
thus $|Fx\ x / x^2 - 6 * Cper\ x / x^4| \leq 104$ **using** g-bound[of x] M
by fastforce
qed

lemma h-integrable:
assumes $M: 3 \leq M$ **and** $MN: M \leq N$
shows interval-lebesgue-integrable lborel (ereal M) (ereal N) $(\lambda x. 6 * Cper\ x / x^4 + (R-r\ x - x * Fx\ x) / x^3)$
proof (rule interval-integrable-bounded[OF MN h-meas])
fix x **assume** $x \in \{M..N\}$
thus $|6 * Cper\ x / x^4 + (R-r\ x - x * Fx\ x) / x^3| \leq 98$ **using** h-bound[of

$x \mid M$
 by *fastforce*
 qed

lemma *g-set-integrable*:
 assumes $M: 3 \leq M$ and $MN: M \leq N$
 shows *set-integrable lborel* $\{M..N\}$ $(\lambda x. Fx \ x / x^2 - 6 * Cper \ x / x^4)$
proof (*rule set-integrable-bounded[OF - - g-meas]*)
 show $\{M..N\} \in \text{sets lborel}$
 by *simp*
 show *emeasure lborel* $\{M..N\} < \infty$ **using** MN
 by *simp*
 fix x **assume** $x \in \{M..N\}$
 thus $|Fx \ x / x^2 - 6 * Cper \ x / x^4| \leq 104$ **using** *g-bound[of x] M*
 by *fastforce*
 qed

lemma *interval-integral-mono-real*:
 fixes $f \ g :: \text{real} \Rightarrow \text{real}$
 assumes $ab: a \leq b$ and $f: \text{interval-lebesgue-integrable lborel } (ereal \ a) \ (ereal \ b) \ f$
 and $g: \text{interval-lebesgue-integrable lborel } (ereal \ a) \ (ereal \ b) \ g$ and $fg: \bigwedge x. x \in \{a <..<b\} \implies f \ x \leq g \ x$
 shows *interval-lebesgue-integral lborel* $(ereal \ a) \ (ereal \ b) \ f \leq$
interval-lebesgue-integral lborel $(ereal \ a) \ (ereal \ b) \ g$
proof –
 have $i1: \text{set-integrable lborel } \{a <..<b\} \ f$ **using** $ab \ f$ **unfolding** *interval-lebesgue-integrable-def*
 by *simp*
 have $i2: \text{set-integrable lborel } \{a <..<b\} \ g$ **using** $ab \ g$ **unfolding** *interval-lebesgue-integrable-def*
 by *simp*
 show *?thesis* **unfolding** *interval-lebesgue-integral-def* **using** ab *set-integral-mono[OF i1 i2 fg]*
 by *fastforce*
 qed

lemma *pt-upper*: $3 \leq x \implies$
 $6 * Cper \ x / x^4 + (R-r \ x - x * Fx \ x) / x^3 \leq 96 / x^4 + 13 / 8 / x^3$
proof –
 assume $x: 3 \leq x$
 have $p3: 0 < x^3$ and $p4: 0 < x^4$ **using** x **by** *simp-all*
 have $6 * Cper \ x / x^4 \leq 96 / x^4$ **using** *Cper-bound[of x] p4*
 by (*auto intro: divide-right-mono*)
 moreover have $(R-r \ x - x * Fx \ x) / x^3 \leq 13 / 8 / x^3$ **using** *Q-bounds[of x] p3*
 by (*fastforce intro: divide-right-mono*)
 ultimately show *?thesis* **by** *linarith*
 qed

lemma *pt-lower*: $3 \leq x \implies$
 $(- 96) / x^4 + (- 1 / 2) / x^3 \leq 6 * Cper \ x / x^4 + (R-r \ x - x * Fx$

$x) / x^3$
proof –
 assume $x: 3 \leq x$
 have $p3: 0 < x^3$ and $p4: 0 < x^4$ using x by *simp-all*
 have $(-96) / x^4 \leq 6 * Cper\ x / x^4$ using $Cper-bound[of\ x]\ p4$
 by (*fastforce intro: divide-right-mono*)
 moreover have $(-1/2) / x^3 \leq (R-r\ x - x * Fx\ x) / x^3$ using
 $Q-bounds[of\ x]\ p3$
 by (*fastforce intro: divide-right-mono*)
 ultimately show *?thesis* by *linarith*
qed

lemma *inv34-integrable*:
 assumes $M: 0 < M$ and $MN: M \leq N$
 shows *interval-lebesgue-integrable lborel (ereal M) (ereal N) ($\lambda x. a / x^4 + b / x^3$)*
proof (*rule interval-integrable-continuous[OF MN]*)
 have $\forall x \in \{M..N\}. x^4 \neq 0$ and $\forall x \in \{M..N\}. x^3 \neq 0$
 using M by *auto*
 thus *continuous-on* $\{M..N\}$ ($\lambda x. a / x^4 + b / x^3$)
 by (*fast intro: continuous-on-add continuous-on-divide continuous-on-const continuous-on-power continuous-on-id*)
qed

$$(5.15): \int -M^N R/x^3 = \Phi(N) - \Phi(M) + \int -M^N (6C/x^4 + Q/x^3).$$

lemma *tail-identity*:
 assumes $M: 3 \leq M$ and $MN: M \leq N$
 shows *interval-lebesgue-integral lborel (ereal M) (ereal N) ($\lambda x. R-r\ x / x^3$)*
 $= \Phi\ N - \Phi\ M + \text{interval-lebesgue-integral lborel (ereal M) (ereal N)}$
 $(\lambda x. 6 * Cper\ x / x^4 + (R-r\ x - x * Fx\ x) / x^3)$
proof –
 let $?g = \lambda x. Fx\ x / x^2 - 6 * Cper\ x / x^4$
 let $?h = \lambda x. 6 * Cper\ x / x^4 + (R-r\ x - x * Fx\ x) / x^3$
 have *split*: $\bigwedge x. x \in \{M <..< N\} \implies R-r\ x / x^3 = ?g\ x + ?h\ x$
proof –
 fix x assume $x: x \in \{M <..< N\}$
 have $x0: x \neq 0$ using $x\ M$ by *auto*
 have $q: (R-r\ x - x * Fx\ x) / x^3 = R-r\ x / x^3 - Fx\ x / x^2$ using $x0$
 by (*simp add: diff-divide-distrib power2-eq-square power3-eq-cube*)
 show $R-r\ x / x^3 = ?g\ x + ?h\ x$ **unfolding** q
 by *simp*
qed
 have $ig: \text{interval-lebesgue-integrable lborel (ereal M) (ereal N) } ?g$
 by (*rule g-integrable[OF M MN]*)
 have $ih: \text{interval-lebesgue-integrable lborel (ereal M) (ereal N) } ?h$
 by (*rule h-integrable[OF M MN]*)

```

have ftc: interval-lebesgue-integral lborel (ereal M) (ereal N) ?g = Phi N - Phi
M
proof -
  have hk: (?g has-integral (Phi N - Phi M)) {M..N}
  proof (rule fundamental-theorem-of-calculus-interior-strong[OF countable-finite[OF
Sx-finite[of
M N]] MN])
    fix x assume x: x ∈ {M<..have x0: 0 < x and xS: x ∉ Sx using x M by auto
    show (Phi has-vector-derivative ?g x) (at x) using Phi-deriv[OF x0 xS]
    unfolding has-real-derivative-iff-has-vector-derivative .
  next
    show continuous-on {M..N} Phi using Phi-cont[of M N] M
    by linarith
  qed
  have si: set-integrable lborel {M..N} ?g by (rule g-set-integrable[OF M MN])
  have e: interval-lebesgue-integral lborel (ereal M) (ereal N) ?g = integral {M..N}
?g
    by (rule interval-integral-eq-integral[OF MN si])
  show ?thesis unfolding e using integral-unique[OF hk]
    by linarith
  qed
have interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. R-r x / x ^ 3) =
interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. ?g x + ?h x)
  by (rule interval-integral-cong-Ioo[OF MN split])
also have ... = interval-lebesgue-integral lborel (ereal M) (ereal N) ?g +
interval-lebesgue-integral lborel (ereal M) (ereal N) ?h
  by (rule interval-lebesgue-integral-add[OF ig ih])
finally show ?thesis unfolding ftc .
qed

lemma integral-inv34:
  assumes 0 < M M ≤ N
  shows interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. a / x ^ 4 + b /
x ^ 3)
    = (- a / (3 * N ^ 3) - b / (2 * N ^ 2)) - (- a / (3 * M ^ 3) - b /
(2 * M ^ 2))
  proof -
    let ?g = λx. a / x ^ 4 + b / x ^ 3
    let ?F = λx. - a / (3 * x ^ 3) - b / (2 * x ^ 2)
    have ∀ x ∈ {M..N}. x ^ 4 ≠ 0 and ∀ x ∈ {M..N}. x ^ 3 ≠ 0
    using assms(1) by auto
    hence cont: continuous-on {M..N} ?g
    by (fast intro: continuous-on-add continuous-on-divide continuous-on-const con-
tinuous-on-power
    continuous-on-id)
    have D: ∧ x. x ∈ {M..N} ⇒
    (?F has-vector-derivative ?g x) (at x within {M..N})
    proof -

```

```

fix  $x$  assume  $x: x \in \{M..N\}$ 
have  $x0: x \neq 0$  using assms(1)  $x$  by auto
have  $n3: 3 * x^3 \neq 0$  and  $n2: 2 * x^2 \neq 0$  using  $x0$  by simp-all
note  $DD = \text{DERIV-diff}[OF \text{DERIV-divide}[OF \text{DERIV-const}[of - a] \text{DERIV-cmult}[OF \text{DERIV-pow}[of 3]] n3]$ 
 $\text{DERIV-divide}[OF \text{DERIV-const}[of b] \text{DERIV-cmult}[OF \text{DERIV-pow}[of 2]] n2]]$ 
have  $v: (0 * (3 * x^3) - a * (3 * (\text{real } 3 * x^{(3 - \text{Suc } 0))})) / (3 * x^3 * (3 * x^3) - (0 * (2 * x^2) - b * (2 * (\text{real } 2 * x^{(2 - \text{Suc } 0))}))) / (2 * x^2 * (2 * x^2)) = a / x^4 + b / x^3$ 
using  $x0$ 
by (simp add: field-simps eval-nat-numeral)
have ( $?F \text{ has-real-derivative } (a / x^4 + b / x^3)$ ) (at  $x$  within  $\{M..N\}$ )
using  $\text{DERIV-cong}[OF DD v]$ 
by blast
thus ( $?F \text{ has-vector-derivative } ?g x$ ) (at  $x$  within  $\{M..N\}$ )
unfolding has-real-derivative-iff-has-vector-derivative .
qed
have  $mm1: \min M N = M$  using assms(2) by simp
have  $mm2: \max M N = N$  using assms(2) by simp
show  $?thesis$  using interval-integral-FTC-finite[of M N ?g ?F] cont D unfolding
 $mm1 mm2$ 
by presburger
qed

```

The tail bounds, uniformly in $M \geq 3$.

lemma *tail-bounds*:

```

assumes  $M: 3 \leq M$ 
shows  $-\text{Phi } M - 32 / M^3 - 1 / (4 * M^2) \leq$ 
 $\text{set-lebesgue-integral lborel } \{M<..\} (\lambda x. R-r x / x^3)$ 
and  $\text{set-lebesgue-integral lborel } \{M<..\} (\lambda x. R-r x / x^3) \leq$ 
 $-\text{Phi } M + 32 / M^3 + 13 / (16 * M^2)$ 
proof -
let  $?f = \lambda x. R-r x / x^3$ 
have  $fin: \bigwedge N. M \leq N \implies$ 
 $\text{interval-lebesgue-integral lborel } (ereal M) (ereal N) ?f \leq$ 
 $\text{Phi } N - \text{Phi } M + ((- 96 / (3 * N^3) - 13 / 8 / (2 * N^2)) - (- 96 / (3 * M^3) - 13 / 8 / (2 * M^2))) \wedge$ 
 $\text{Phi } N - \text{Phi } M + ((96 / (3 * N^3) - (- 1 / 2) / (2 * N^2)) - (96 / (3 * M^3) - (- 1 / 2) / (2 * M^2))) \leq \text{interval-lebesgue-integral lborel } (ereal M) (ereal N) ?f$ 
proof -
fix  $N$  assume  $MN: M \leq N$ 
have  $M0: 0 < M$  using  $M$  by simp
let  $?h = \lambda x. 6 * Cper x / x^4 + (R-r x - x * Fx x) / x^3$ 

```

```

have T: interval-lebesgue-integral lborel (ereal M) (ereal N) ?f =
  Phi N - Phi M + interval-lebesgue-integral lborel (ereal M) (ereal N) ?h
by (rule tail-identity[OF M MN])
have ih: interval-lebesgue-integrable lborel (ereal M) (ereal N) ?h
by (rule h-integrable[OF M MN])
have interval-lebesgue-integral lborel (ereal M) (ereal N) ?h ≤
  interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. 96 / x ^ 4 + 13 / 8
/
  x ^ 3)
proof (rule interval-integral-mono-real[OF MN ih inv34-integrable[OF M0
MN]])
  fix x assume x ∈ {M <..thus ?h x ≤ 96 / x ^ 4 + 13 / 8 / x ^ 3 using pt-upper[of x] M
  by fastforce
qed
moreover have interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. (-
96) /
  x ^ 4 + (- 1 / 2) / x ^ 3) ≤ interval-lebesgue-integral lborel (ereal M) (ereal
N) ?h
proof (rule interval-integral-mono-real[OF MN inv34-integrable[OF M0 MN]
ih])
  fix x assume x ∈ {M <..thus (- 96) / x ^ 4 + (- 1 / 2) / x ^ 3 ≤ ?h x using pt-lower[of x] M
  by fastforce
qed
moreover have interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. 96 /
x ^ 4 +
  13 / 8 / x ^ 3) = (- 96 / (3 * N ^ 3) - 13 / 8 / (2 * N ^ 2)) - (- 96 /
(3 * M ^ 3) - 13 / 8 /
  (2 * M ^ 2))
by (rule integral-inv34[OF M0 MN])
moreover have interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. (-
96) /
  x ^ 4 + (- 1 / 2) / x ^ 3) = (- (- 96) / (3 * N ^ 3) - (- 1 / 2) / (2 *
N ^ 2)) - (- (- 96) /
  (3 * M ^ 3) - (- 1 / 2) / (2 * M ^ 2))
by (rule integral-inv34[OF M0 MN])
hence interval-lebesgue-integral lborel (ereal M) (ereal N) (λx. (- 96) / x ^ 4
+
  (- 1 / 2) / x ^ 3) = (96 / (3 * N ^ 3) - (- 1 / 2) / (2 * N ^ 2)) - (96
/ (3 * M ^ 3) - (- 1 /
  2) / (2 * M ^ 2))
by simp
ultimately show interval-lebesgue-integral lborel (ereal M) (ereal N) ?f ≤
  Phi N - Phi M + ((- 96 / (3 * N ^ 3) - 13 / 8 / (2 * N ^ 2)) - (- 96
/ (3 * M ^ 3) - 13 / 8 /
  (2 * M ^ 2))) ∧
  Phi N - Phi M + ((96 / (3 * N ^ 3) - (- 1 / 2) / (2 * N ^ 2)) - (96 /
(3 * M ^ 3) - (- 1 /
  2) / (2 * M ^ 2)))

```



```

      2) / (2 * M ^ 2))) ≤ interval-lebesgue-integral lborel (ereal M) (ereal N) ?f
    using T by linarith
  qed
  have lim: ((λN. interval-lebesgue-integral lborel (ereal M) (ereal N) ?f)
    → set-lebesgue-integral lborel {M<..} ?f) at-top
  proof -
    have int: set-integrable lborel {M..} ?f
      using set-integrable-subset[OF R-div-cube-integrable, of {M..}] M
      by auto
    have T: ((λb. set-lebesgue-integral lborel {M..b} ?f) →
      set-lebesgue-integral lborel {M..} ?f) at-top
      using tendsto-set-lebesgue-integral-at-top[OF - int]
      by simp
    have E1: set-lebesgue-integral lborel {M..} ?f = set-lebesgue-integral lborel
      {M<..} ?f
      using set-integral-discrete-difference[where X = {M} and A = {M..} and
      B =
        {M<..} and M = lborel and f = ?f]
      by fastforce
    have E2: eventually (λN. set-lebesgue-integral lborel {M..N} ?f =
      interval-lebesgue-integral lborel (ereal M) (ereal N) ?f) at-top
    proof (rule eventually-at-top-linorderI[of M])
      fix N assume M ≤ N
      thus set-lebesgue-integral lborel {M..N} ?f =
        interval-lebesgue-integral lborel (ereal M) (ereal N) ?f
        using interval-integral-Icc[of M N ?f]
        by argo
    qed
    show ?thesis using tendsto-cong[OF E2] T E1
      by simp
  qed
  have PhiN: ((λN. Phi N) → 0) at-top and N3: ((λN :: real. 1
    / N ^ 3) → 0) at-top and N2: ((λN :: real. 1 /
    N ^ 2) → 0) at-top
  proof -
    have ev: eventually (λN. norm (Phi N) ≤ 65 / N) at-top
    proof (rule eventually-at-top-linorderI[of 1])
      fix N :: real assume 1 ≤ N
      thus norm (Phi N) ≤ 65 / N using Phi-abs-le[of N]
      by simp
    qed
    have l65: ((λN::real. 65 / N) → 0) at-top
      using real-tendsto-divide-at-top[OF tendsto-const filterlim-ident]
      by auto
    show ((λN. Phi N) → 0) at-top using Lim-null-comparison[OF ev l65]
      by simp
    have p3: filterlim (λN::real. N ^ 3) at-top at-top
      using filterlim-pow-at-top[OF - filterlim-ident, of 3]
      by simp

```

```

have p2: filterlim (λN::real. N ^ 2) at-top at-top
  using filterlim-pow-at-top[OF - filterlim-ident, of 2]
  by auto
show ((λN :: real. 1 / N ^ 3) → 0) at-top
  using real-tendsto-divide-at-top[OF tendsto-const p3]
  by blast
show ((λN :: real. 1 / N ^ 2) → 0) at-top
  using real-tendsto-divide-at-top[OF tendsto-const p2]
  by auto
qed
show - Phi M - 32 / M ^ 3 - 1 / (4 * M ^ 2) ≤ set-lebesgue-integral lborel
{M<..} ?f
proof -
  let ?K = 96 / (3 * M ^ 3) - (- 1 / 2) / (2 * M ^ 2)
  have eq: λN::real. 96 / (3 * N ^ 3) - (- 1 / 2) / (2 * N ^ 2) =
    32 * (1 / N ^ 3) + 1 / 4 * (1 / N ^ 2)
  by (simp add: field-simps)
  note G = tendsto-add[OF tendsto-diff[OF PhiN tendsto-const[of Phi M]] tend-
sto-diff[OF
    tendsto-add[OF tendsto-mult[OF tendsto-const[of 32] N3] tendsto-mult[OF
tendsto-const[of 1 /
4] N2]] tendsto-const[of ?K]]]
  have gl: ((λN. Phi N - Phi M + ((96 / (3 * N ^ 3) - (- 1 / 2) / (2 * N ^
2)) -
    ?K)) → 0 - Phi M + ((32 * 0 + 1 / 4 * 0) - ?K)) at-top
  unfolding eq using G
  by auto
  have ev: eventually (λN. Phi N - Phi M + ((96 / (3 * N ^ 3) - (- 1 / 2)
/ (2 *
    N ^ 2)) - ?K) ≤ interval-lebesgue-integral lborel (ereal M) (ereal N) ?f)
at-top
  proof (rule eventually-at-top-linorderI[of M])
    fix N assume M ≤ N
    thus Phi N - Phi M + ((96 / (3 * N ^ 3) - (- 1 / 2) / (2 * N ^ 2)) -
?K) ≤
      interval-lebesgue-integral lborel (ereal M) (ereal N) ?f
    using fin[of N]
    by linarith
  qed
  have 0 - Phi M + ((32 * 0 + 1 / 4 * 0) - ?K) ≤ set-lebesgue-integral lborel
{M<..} ?f
  by (rule tendsto-le[OF trivial-limit-at-top-linorder lim gl ev])
  moreover have 0 - Phi M + ((32 * 0 + 1 / 4 * 0) - ?K) = - Phi M -
32 / M ^ 3 - 1 / (4 * M ^ 2)
  by (simp add: field-simps)
  ultimately show ?thesis by linarith
qed
show set-lebesgue-integral lborel {M<..} ?f ≤ - Phi M + 32 / M ^ 3 + 13 /
(16 * M ^ 2)

```

```

proof –
  let ?K = - 96 / (3 * M ^ 3) - 13 / 8 / (2 * M ^ 2)
  have eq:  $\bigwedge N :: \text{real. } - 96 / (3 * N ^ 3) - 13 / 8 / (2 * N ^ 2) =$ 
     $(- 32) * (1 / N ^ 3) + (- 13 / 16) * (1 / N ^ 2)$ 
    by (simp add: field-simps)
  note G = tendsto-add[OF tendsto-diff[OF PhiN tendsto-const[of Phi M]] tend-
    sto-diff[OF
      tendsto-add[OF tendsto-mult[OF tendsto-const[of - 32] N3] tendsto-mult[OF
        tendsto-const[of
          - 13 / 16] N2]] tendsto-const[of ?K]]]
  have gl: (( $\lambda N. \text{Phi } N - \text{Phi } M + ((- 96 / (3 * N ^ 3) - 13 / 8 / (2 * N ^ 2)) - ?K)$ )
    2)) –
    ?K))  $\longrightarrow 0 - \text{Phi } M + (((- 32) * 0 + (- 13 / 16) * 0) - ?K)$  at-top
    unfolding eq using G
    by simp
  have ev: eventually ( $\lambda N. \text{interval-lebesgue-integral lborel } (ereal M) (ereal N) ?f$ 
    ≤
      Phi N - Phi M + ((- 96 / (3 * N ^ 3) - 13 / 8 / (2 * N ^ 2)) - ?K))
    at-top
  proof (rule eventually-at-top-linorderI[of M])
    fix N assume M ≤ N
    thus interval-lebesgue-integral lborel (ereal M) (ereal N) ?f ≤
      Phi N - Phi M + ((- 96 / (3 * N ^ 3) - 13 / 8 / (2 * N ^ 2)) - ?K)
      using fin[of N]
      by linarith
  qed
  have set-lebesgue-integral lborel {M<..} ?f ≤
    0 - Phi M + (((- 32) * 0 + (- 13 / 16) * 0) - ?K)
    by (rule tendsto-le[OF trivial-limit-at-top-linorder gl lim ev])
  moreover have 0 - Phi M + (((- 32) * 0 + (- 13 / 16) * 0) - ?K) =
    - Phi M + 32 / M ^ 3 + 13 / (16 * M ^ 2)
    by (simp add: field-simps)
  ultimately show ?thesis by linarith
qed
qed

```

lemma Phi-20: $\text{Phi } 20 = (37 / 2 - 2923 / 240) / 400 + 37 / 800$

```

proof –
  have a: alpha-r * 20 = 3 / 2 by (simp add: alpha-r-def)
  have f1: frac (20 :: real) = 0
    by fastforce
  have f1:  $\lfloor 3 / 2 :: \text{real} \rfloor = 1$ 
    by (force simp: floor-eq-iff)
  have f2: frac (3 / 2 :: real) = 1 / 2 unfolding frac-def f1
    by simp
  show ?thesis unfolding Phi-def Pper-def Cper-def G0-r-def a f1 f2
    by (simp add: lam-r-def alpha-r-def)
qed

```

(5.16) (tight: $P(20) = 37/2, C(20) = 0$).

theorem *R-tail-20-bound*: *set-lebesgue-integral lborel {20<..} (λx. R-r x / x ^ 3)*
 $\leq$
 $- 2689 / 48000$
using *tail-bounds(2)[of 20]* **unfolding** *Phi-20* **by** *simp*

(5.17): for $40 \mid M$, $\Phi(M) = -(2923/240)/M^2 + \lambda/M$.

theorem *R-tail-M-bound*:
assumes $40 \text{ dvd } M \ 0 < M$
shows $-\text{lam-r} / \text{real } M + (2923 / 240 - 1 / 4) / \text{real } M^2 - 32 / \text{real } M^3$
 $\leq \text{set-lebesgue-integral lborel \{real } M<..\} (\lambda x. R-r x / x^3)$

proof –
have *M3*: $3 \leq \text{real } M$ **using** *assms* **by** (*auto elim!: dvdE*)
have *fr*: $\text{frac } (\text{real } M) = 0 \ \text{frac } (\text{alpha-r} * \text{real } M) = 0$
proof –
obtain *k* **where** $k: M = 40 * k$ **using** *assms(1)* **by** (*elim dvdE*)
have *e*: $\text{alpha-r} * \text{real } M = \text{real } (3 * k)$ **unfolding** *k* *alpha-r-def* **by** *auto*
show $\text{frac } (\text{real } M) = 0 \ \text{frac } (\text{alpha-r} * \text{real } M) = 0$ **unfolding** *e* **by** (*auto simp: frac-def*)
qed
have *PhiM*: $\Phi(\text{real } M) = -(2923 / 240) / \text{real } M^2 + \text{lam-r} / \text{real } M$
unfolding *Phi-def Pper-def Cper-def G0-r-def fr* **by** *simp*
show *?thesis* **using** *tail-bounds(1)[OF M3]* **unfolding** *PhiM* **by** (*simp add: field-simps*)
qed

26.6 Continuity off a countable set, and boundedness

D-R: the points where some $c \ x$ ($c \in \{1, 2, \alpha, 2\alpha, 2H-c, 2\lambda\}$) is an integer.

definition *DR* :: *real set where*

$DR = (\bigcup c \in \{1, 2, \text{alpha-r}, 2 * \text{alpha-r}, 2 * Hc-r, 2 * \text{lam-r}\}. (\lambda n :: \text{int. of-int } n / c) \text{ ' UNIV})$

lemma *DR-countable*: *countable DR*

unfolding *DR-def* **by** (*intro countable-UN countable-image*) *simp-all*

A function that agrees with a continuous one on an open neighbourhood is continuous there.

lemma *isCont-local*:

fixes $f \ g :: \text{real} \Rightarrow \text{real}$

assumes $\text{open } S \ y \in S \ \bigwedge z. z \in S \implies f \ z = g \ z$ *isCont g y*

shows *isCont f y*

proof –

have *ev0*: *eventually* $(\lambda z. z \in S)$ (*nhds y*)

using *eventually-nhds-in-open[OF assms(1,2)]*

by *presburger*

have *ev*: *eventually* $(\lambda z. f \ z = g \ z)$ (*nhds y*) **using** *eventually-mono[OF ev0]* *assms(3)*

```

    by force
  show ?thesis using isCont-cong[OF ev] assms(4)
    by fastforce
qed

lemma frac-isCont:  $(y :: \text{real}) \notin \mathbb{Z} \implies \text{isCont } \text{frac } y$ 
  using continuous-fraction[of y] by blast

lemma es-isCont:  $u \neq 1 / 2 \implies \text{isCont } es \ u$ 
proof -
  assume u:  $u \neq 1 / 2$ 
  show ?thesis
  proof (cases  $u < 1 / 2$ )
    case True
    show ?thesis
    proof (rule isCont-local[of  $\{.. < 1 / 2\}$  u es  $\lambda-. 1$ ])
      show open  $\{.. < 1 / 2 :: \text{real}\}$ 
        by force
      show  $u \in \{.. < 1 / 2\}$  using True
        by force
      show  $\bigwedge z. z \in \{.. < 1 / 2\} \implies es \ z = 1$  unfolding es-def
        by fastforce
      show isCont  $(\lambda-. 1 :: \text{real}) \ u$ 
        by simp
    qed
  next
    case False
    hence g:  $1 / 2 < u$  using u
      by auto
    show ?thesis
    proof (rule isCont-local[of  $\{1 / 2 <..\}$  u es  $\lambda-. - 1$ ])
      show open  $\{1 / 2 < :: \text{real}\}$ 
        by auto
      show  $u \in \{1 / 2 <..\}$  using g
        by blast
      show  $\bigwedge z. z \in \{1 / 2 <..\} \implies es \ z = - 1$  unfolding es-def
        by force
      show isCont  $(\lambda-. - 1 :: \text{real}) \ u$ 
        by auto
    qed
  qed
qed
qed

lemma d0-cont: isCont d0 u and pos-cont: isCont pos-r u
  unfolding d0-def pos-r-def by (intro continuous-intros)+

lemma isCont-R-form:
  fixes a b c d e :: real  $\Rightarrow$  real and x :: real
  assumes isCont a x isCont b x isCont c x isCont d x isCont e x

```

```

    isCont ( $\lambda x. es (a x)$ )  $x$  isCont ( $\lambda x. es (b x)$ )  $x$ 
  shows isCont ( $\lambda x. x * (4 * lam-r + 2 * lam-r * a x - 12 * lam-r * b x) +$ 
     $Qc (a x) (b x) (c x) (d x) (e x) x$ 
  proof -
    have d1: isCont ( $\lambda x. d0 (a x)$ )  $x$  by (rule isCont-o2[OF assms(1) d0-cont])
    have da: isCont ( $\lambda x. d0 (b x)$ )  $x$  by (rule isCont-o2[OF assms(2) d0-cont])
    have pp0: isCont ( $\lambda x. (c x - d x) / 2$ )  $x$  using assms(3,4)
      by (auto intro: continuous-intros)
    have pp: isCont ( $\lambda x. pos-r ((c x - d x) / 2)$ )  $x$  by (rule isCont-o2[OF pp0
pos-cont])
    show ?thesis unfolding Qc-def
      by (fastforce intro: continuous-intros assms d1 da pp)
  qed

```

Every ingredient of R -eq is continuous at $x \notin D$ -R.

```

theorem R-isCont-off-countable:  $\exists D. countable D \wedge (\forall x. 0 < x \longrightarrow$ 
   $x \notin D \longrightarrow isCont R-r x)$ 
proof (intro exI[of - DR] conjI allI impI DR-countable)
  fix  $x :: real$  assume  $x: 0 < x x \notin DR$ 
  have nI:  $c * x \notin \mathbb{Z}$  if  $c \in \{1, 2, alpha-r, 2 * alpha-r, 2 * Hc-r, 2 * lam-r\}$ 
    for  $c$ 
  proof
    assume  $c * x \in \mathbb{Z}$ 
    then obtain  $n$  where  $n: c * x = of-int n$  by (auto elim: Ints-cases)
    have  $c \neq 0$  using that by (force simp: alpha-r-def Hc-r-def lam-r-def)
    hence  $x = of-int n / c$  using  $n$  by (auto simp: field-simps)
    hence  $x \in DR$  unfolding DR-def using that by fast
    thus False using  $x(2)$  by simp
  qed
  have fc: isCont ( $\lambda x. frac (c * x)$ )  $x$ 
    if  $c \in \{1, 2, alpha-r, 2 * alpha-r, 2 * Hc-r, 2 * lam-r\}$  for  $c$ 
  proof -
    have cm: isCont ( $\lambda x. c * x$ )  $x$  by (auto intro: continuous-intros)
    show ?thesis using isCont-o2[OF cm frac-isCont[OF nI[OF that]]] by pres-
burger
  qed
  have ec: isCont ( $\lambda x. es (frac (c * x))$ )  $x$  if  $c \in \{1, alpha-r\}$  for  $c$ 
  proof -
    have c1:  $c \in \{1, 2, alpha-r, 2 * alpha-r, 2 * Hc-r, 2 * lam-r\}$  using that
      by fastforce
    have c2:  $2 * c \in \{1, 2, alpha-r, 2 * alpha-r, 2 * Hc-r, 2 * lam-r\}$  using that
      by auto
    have h:  $frac (c * x) \neq 1 / 2$ 
    proof
      assume  $h: frac (c * x) = 1 / 2$ 
      have  $e: 2 * c * x = of-int (2 * \lfloor c * x \rfloor + 1)$  using  $h$  unfolding frac-def
        by (auto simp: field-simps)
      have  $2 * c * x \in \mathbb{Z}$  unfolding  $e$  by (rule Ints-of-int)
      thus False using nI[OF c2]
    qed
  qed

```

```

    by meson
  qed
  show ?thesis using isCont-o2[OF fc[OF c1] es-isCont[OF h]]
    by fastforce
  qed
  have e:  $R-r = (\lambda x. x * Fx x + Qc (frac x) (frac (alpha-r * x)) (frac (2 * Hc-r * x)) (frac (2 * x)) (frac (2 * (lam-r * x))))$ 
    by (simp add: fun-eq-iff R-eq)
  have f1: isCont frac x using fc[of 1]
    by fastforce
  have fa: isCont ( $\lambda x. frac (alpha-r * x)$ ) x using fc[of alpha-r]
    by fastforce
  have fH: isCont ( $\lambda x. frac (2 * Hc-r * x)$ ) x using fc[of 2 * Hc-r]
    by fast
  have f2: isCont ( $\lambda x. frac (2 * x)$ ) x using fc[of 2]
    by fastforce
  have fl: isCont ( $\lambda x. frac (2 * (lam-r * x))$ ) x using fc[of 2 * lam-r]
    unfolding mult.assoc
    by fastforce
  have e1: isCont ( $\lambda x. es (frac x)$ ) x using ec[of 1]
    by fastforce
  have ea: isCont ( $\lambda x. es (frac (alpha-r * x))$ ) x using ec[of alpha-r]
    by fastforce
  show isCont R-r x unfolding e Fx-def using isCont-R-form[OF f1 fa fH f2 fl e1 ea]
    by meson
  qed

```

definition $DO :: \text{real set}$ **where** $DO = \{1 / 3, 1 / 2, 1\} \cup \text{range } (\lambda n. 1 / \text{real } n)$

lemma $DO\text{-countable}$: countable DO
 unfolding $DO\text{-def}$ **by** (intro countable-Un countable-image) simp-all

theorem $\text{outer-integrand-isCont-off-countable}$: $\exists D. \text{countable } D \wedge (\forall y. 0 < y \longrightarrow y \notin D \longrightarrow \text{isCont outer-integrand } y)$

proof (intro exI[of - DO] conjI allI impI $DO\text{-countable}$)

fix $y :: \text{real}$ **assume** $y: 0 < y \wedge y \notin DO$

have $R0$: isCont $R0-r$ y

proof -

have $y \neq 1 / 3$ **using** $y(2)$ **unfolding** $DO\text{-def}$

by fast

moreover **have** $y \neq 1 / 2$ **using** $y(2)$ **unfolding** $DO\text{-def}$

by fastforce

moreover **have** $y \neq 1$ **using** $y(2)$ **unfolding** $DO\text{-def}$

by force

ultimately consider $(A) 1 / 3 < y \wedge y < 1 / 2 \mid (B) 1 / 2 < y \wedge y < 1 \mid (C) y \in -\{1 / 3..1\}$

by fastforce

```

thus ?thesis
proof cases
  case A
  show ?thesis
  proof (rule isCont-local[of {1 / 3 <..show open {1 / 3 <..by force
  show y ∈ {1 / 3 <..using A
  by simp
  show ∧z. z ∈ {1 / 3 <..unfolding R0-r-def
  by force
  show isCont (λy. 8 - 9 * y - 8 * alpha-r - 5 * min alpha-r (1 - 2 * y)
- 5 *
    pos-r (1 + alpha-r - 3 * y)) y
  unfolding pos-r-def
  by (fast intro: continuous-intros)
qed
next
  case B
  show ?thesis
  proof (rule isCont-local[of {1 / 2 <..show open {1 / 2 <..by force
  show y ∈ {1 / 2 <..using B
  by auto
  show ∧z. z ∈ {1 / 2 <..unfolding R0-r-def
  by fastforce
  show isCont (λy. 7 * (1 - y) - 6 * min alpha-r (1 - y) - 6 * pos-r (1
+ alpha-r -
    2 * y) + pos-r (1 + 4 * alpha-r - 2 * y)) y
  unfolding pos-r-def
  by (fast intro: continuous-intros)
qed
next
  case C
  show ?thesis
  proof (rule isCont-local[of - {1 / 3..1} y R0-r λ-. 0])

```



```

    show open ( $-\{1 / 3..1 :: \text{real}\}$ )
      by fastforce
    show  $y \in -\{1 / 3..1\}$  using C
      by fast
    show  $\bigwedge z. z \in -\{1 / 3..1\} \implies R0\text{-}r\ z = 0$ 
      unfolding R0-r-def
      by auto
    show isCont ( $\lambda-. 0 :: \text{real}$ ) y
      by auto
  qed
qed
qed
have d: isCont dfun-r y
proof (cases  $1 / 3 < y \wedge y < 1 / 2$ )
  case True
    show ?thesis
    proof (rule isCont-local[of  $\{1 / 3 <..< 1 / 2\}$  y dfun-r  $\lambda y. \text{pos-r } ((1 + 4 * \text{alpha-r} - 3 * y) - \text{pos-r } (1 + \text{alpha-r} - 3 * y))$ ])
      show open  $\{1 / 3 <..< 1 / 2 :: \text{real}\}$ 
        by force
      show  $y \in \{1 / 3 <..< 1 / 2\}$  using True
        by simp
      show  $\bigwedge z. z \in \{1 / 3 <..< 1 / 2\} \implies$ 
        dfun-r z = pos-r  $((1 + 4 * \text{alpha-r} - 3 * z) - \text{pos-r } (1 + \text{alpha-r} - 3 * z))$ 
        unfolding dfun-r-def
        by force
      show isCont ( $\lambda y. \text{pos-r } ((1 + 4 * \text{alpha-r} - 3 * y) - \text{pos-r } (1 + \text{alpha-r} - 3 * y))$ ) y
        unfolding pos-r-def
        by (fast intro: continuous-intros)
    qed
  next
  case False
    have ne1:  $y \neq 1 / 3$  using y(2) unfolding DO-def
      by fast
    have ne2:  $y \neq 1 / 2$  using y(2) unfolding DO-def
      by fastforce
    note ne = ne1 ne2
    show ?thesis
    proof (rule isCont-local[of  $-\{1 / 3..1 / 2\}$  y dfun-r  $\lambda-. 0$ ])
      show open ( $-\{1 / 3..1 / 2 :: \text{real}\}$ )
        by fast
      show  $y \in -\{1 / 3..1 / 2\}$  using False ne
        by force
      show  $\bigwedge z. z \in -\{1 / 3..1 / 2\} \implies \text{dfun-r } z = 0$ 
        unfolding dfun-r-def
        by fastforce
    qed
  qed

```

```

    show isCont ( $\lambda\cdot. 0 :: \text{real}$ ) y
      by auto
  qed
qed
have fl: isCont ( $\lambda y. \text{of-int } (\text{floor } (1 / y)) :: \text{real}$ ) y
proof -
  have nI:  $1 / y \notin \mathbb{Z}$ 
  proof
    assume  $1 / y \in \mathbb{Z}$ 
    then obtain n where  $n: 1 / y = \text{of-int } n$  by (auto elim: Ints-cases)
    have p:  $0 < 1 / y$  using y(1)
      by auto
    have  $0 < n$  using n p
      by presburger
    hence r:  $\text{real } (\text{nat } n) = \text{of-int } n$ 
      by fastforce
    have  $y = 1 / \text{real } (\text{nat } n)$  unfolding r n[symmetric]
      by simp
    hence  $y \in DO$  unfolding DO-def
      by fast
    thus False using y(2)
      by meson
  qed
  have c1: isCont ( $\lambda y. 1 / y$ ) y using y(1)
    by (fast intro: continuous-intros)
  have c2: isCont ( $\text{real-of-int} \circ \text{floor}$ ) ( $1 / y$ ) by (rule continuous-floor[OF nI])
  show ?thesis using isCont-o2[OF c1 c2]
    by (force simp: o-def)
  qed
show isCont outer-integrand y
  unfolding outer-integrand-def pos-r-def using R0 d fl
  by (fast intro: continuous-intros)
qed

theorem outer-integrand-bounded:  $\exists Bd. \forall y \in \{1 / 3..2 * \text{lam-r}\}. |\text{outer-integrand } y| \leq Bd$ 
proof (intro exI[of - 100] ballI)
  fix y :: real assume y:  $y \in \{1 / 3..2 * \text{lam-r}\}$ 
  have R0:  $|\text{R0-r } y| \leq 20$  and d:  $|\text{dfun-r } y| \leq 20$ 
  proof -
    have y1a:  $1 / 3 \leq y$  using y unfolding lam-r-def
      by presburger
    have y1b:  $y \leq 37 / 20$  using y unfolding lam-r-def
      by force
    note y1 = y1a y1b
    show  $|\text{R0-r } y| \leq 20$  using y1 unfolding R0-r-def pos-r-def alpha-r-def
      by argo
    show  $|\text{dfun-r } y| \leq 20$  using y1 unfolding dfun-r-def pos-r-def alpha-r-def
      by fastforce
  qed

```

```

qed
have fl: |of-int (floor (1 / y))| ≤ (4 :: real)
proof -
  have y0: 0 < y using y
  by force
  have a: 1 / y ≤ 3 using y y0
  by (auto simp: field-simps)
  have b: 0 ≤ 1 / y using y0
  by simp
  have c: of-int (floor (1 / y)) ≤ 1 / y by (rule of-int-floor-le)
  have d: 0 ≤ floor (1 / y) using b
  by simp
  show ?thesis using a c d
  by fastforce
qed
have s: |∑ j = 1..5. pos-r (2 * lam-r - real j * y)| ≤ 10
proof -
  have t: 0 ≤ pos-r (2 * lam-r - real j * y) ∧ pos-r (2 * lam-r - real j * y) ≤
2
    if j: j ∈ {1..5::nat} for j
  proof -
    have jy: 0 ≤ real j * y using y
    by auto
    show ?thesis using jy unfolding pos-r-def lam-r-def
    by linarith
  qed
  have (∑ j = 1..5. pos-r (2 * lam-r - real j * y)) ≤ of-nat (card {1..5::nat})
* 2
    using sum-bounded-above[of {1..5::nat} λj. pos-r (2 * lam-r - real j * y) 2]
t
    by blast
  moreover have 0 ≤ (∑ j = 1..5. pos-r (2 * lam-r - real j * y))
  using sum-nonneg[of {1..5::nat} λj. pos-r (2 * lam-r - real j * y)] t
  by blast
  ultimately show ?thesis
  by simp
qed
show |outer-integrand y| ≤ 100
proof -
  have l: |2 * lam-r * of-int (floor (1 / y))| ≤ 8
  using fl[unfolded abs-le-iff] unfolding abs-le-iff lam-r-def
  by linarith
  show ?thesis
  unfolding outer-integrand-def abs-le-iff
  using R0[unfolded abs-le-iff] d[unfolded abs-le-iff] l[unfolded abs-le-iff]
s[unfolded
  abs-le-iff]
  by linarith
qed

```

qed

end

theory *Zeta5-Asymptotics*
imports *Zeta5-Integral-Representation Prime-Sum-Limits Zeta5-Explicit-Integrals*
Zeta5-Uniform-Asymptotics
begin

27 Normalisation and the asymptotics of $Q_{K,M}$: Propositions 5.1 and 5.2

Section 5 of the paper: the standing hypotheses, Legendre's formula and the uniformity estimates, integrality of $Q_{K,M}$ (Proposition 5.1, *QKM-int-poly*), the small-prime sum, the prime sums through the prime number theorem, and the limit (5.21) of $K^{-2} \ln m_{K,M}$ (Proposition 5.2, *log-mKM-limsup*, *log-mKM-limsup-AM*). Lean theories *Section5*, *Prop51*, *Legendre5*, *Prime-Sums*, *PNT*, *Prop52*, *Section5Final*.

27.1 The small-prime sum

lemma *hof-le-lam*: $\text{real } (\text{hof } K) \leq \text{lam-r} * \text{real } K$
using *of-nat-div-le-of-nat*[of 37 * K 40] **by** (*simp add: hof-def lam-r-def*)

Small primes $p \leq K/M$: the contribution of $-L-p$ is $6 \lambda K^2/M + o(K^2)$.

theorem *small-prime-sum-bound*:

assumes $M: 0 < M$ **and** $e: 0 < \varepsilon$

shows $\forall_F K$ *in at-top*.

$(\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\}.$

$\text{real } (6 * \text{hof } K * \text{nat-log } p (5 * K) + \text{hof } K * \text{multiplicity } p (24 :: \text{nat})) * \ln$
 $(\text{real } p))$

$\leq (6 * \text{lam-r} / \text{real } M + \varepsilon) * \text{real } K^2$

proof –

define δ **where** $\delta = \varepsilon / 13$

have $d0: 0 < \delta$ **unfolding** $\delta\text{-def}$ **using** e **by** *simp*

have $Mr: 0 < \text{real } M$ **using** M **by** *simp*

have $\forall_F K :: \text{nat}$ *in at-top*. $\text{primes-theta } (1 / \text{real } M * \text{real } K) - \text{primes-theta}$
 $(0 *$

$\text{real } K) \leq (1 / \text{real } M - 0 + \delta) * \text{real } K$

by (*rule cell-upper*) (*use Mr d0 in simp-all*)

moreover have $\forall_F K :: \text{nat}$ *in at-top*. $2 * \ln (\text{real } (5 * K)) * \text{sqrt } (\text{real } (5 * K)) \leq \delta * \text{real } K$

by (*rule eventually-psi-theta-small[OF d0]*)

moreover have $\forall_F K :: \text{nat}$ *in at-top*. $25 * \ln 24 / \delta \leq \text{real } K$

using *filterlim-real-sequentially* **unfolding** *filterlim-at-top* **by** *simp*

ultimately show *?thesis*

using *eventually-gt-at-top*[of 0 :: nat]

```

proof (eventually-elim)
  case (elim K)
  let ?s = {p. p ≤ K ∧ prime p ∧ p * M ≤ K}
  have fin: finite ?s by simp
  have Kr: 0 < real K using elim(4) by simp
  have split: (∑ p ∈ ?s. real (6 * hof K * nat-log p (5 * K) + hof K *
    multiplicity p (24::nat)) * ln (real p))
    = 6 * real (hof K) * ((∑ p ∈ ?s. ln (real p)) + (∑ p ∈
      ?s. (real (nat-log p (5 * K)) - 1) * ln (real p)))
    + real (hof K) * (∑ p ∈ ?s. real (multiplicity p (24::nat)) * ln (real p))
  proof -
    have pw: real (6 * hof K * nat-log p (5 * K) + hof K * multiplicity p
    (24::nat)) *
      ln (real p) =
      6 * real (hof K) * (ln (real p) + (real (nat-log p (5 * K)) - 1) * ln (real
    p)) +
      real (hof K) * (real (multiplicity p (24::nat)) * ln (real p)) for p
    by (simp add: algebra-simps)
  show ?thesis unfolding pw by (simp add: sum.distrib sum-distrib-left[symmetric])
  qed
  have th: (∑ p ∈ ?s. ln (real p)) = primes-theta (1 / real M * real K)
  proof -
    have nn: 0 ≤ 1 / real M * real K by simp
    have seq: ?s = {p. prime p ∧ real p ≤ 1 / real M * real K}
    proof (intro equalityI subsetI)
      fix p assume p ∈ ?s
      hence p: prime p p * M ≤ K by auto
    have real p * real M ≤ real K using p(2) by (metis of-nat-mult of-nat-le-iff)
      hence real p ≤ 1 / real M * real K using Mr by (simp add: field-simps)
      thus p ∈ {p. prime p ∧ real p ≤ 1 / real M * real K} using p(1) by simp
    next
      fix p assume p ∈ {p. prime p ∧ real p ≤ 1 / real M * real K}
      hence p: prime p real p ≤ 1 / real M * real K by auto
      have real p * real M ≤ real K using p(2) Mr by (simp add: field-simps)
      hence pM: p * M ≤ K by (metis of-nat-mult of-nat-le-iff)
      have p ≤ p * M using M by simp
      hence p ≤ K using pM by (rule le-trans)
      thus p ∈ ?s using p(1) pM by simp
    qed
  show ?thesis unfolding seq theta-eq-sum[OF nn] ..
  qed
  have psi: (∑ p ∈ ?s. (real (nat-log p (5 * K)) - 1) * ln (real p)) ≤
    2 * ln (real (5 * K)) * sqrt (real (5 * K))
  proof -
    have (∑ p ∈ ?s. (real (nat-log p (5 * K)) - 1) * ln (real p)) ≤ (∑ p ∈
      {p. prime p ∧ p ≤ 5 * K}. (real (nat-log p (5 * K)) - 1) * ln (real p))
    proof (rule sum-mono2)
      show finite {p. prime p ∧ p ≤ 5 * K} by simp
      show ?s ⊆ {p. prime p ∧ p ≤ 5 * K} by fastforce
    qed

```

```

    fix p assume p ∈ {p. prime p ∧ p ≤ 5 * K} - ?s
    hence p: prime p p ≤ 5 * K by auto
    have p1: 1 < p using p(1) by (rule prime-gt-1-nat)
    have Suc 1 ≤ floorlog p (5 * K) by (rule floorlog-ge-SucI) (use p p1 in
simp-all)
    hence 1 ≤ nat-log p (5 * K) unfolding nat-log-def by linarith
    moreover have 0 ≤ ln (real p) using p1 by simp
    ultimately show 0 ≤ (real (nat-log p (5 * K)) - 1) * ln (real p) by simp
qed
also have ... ≤ 2 * ln (real (5 * K)) * sqrt (real (5 * K))
    by (rule psi-minus-theta-sum) (use elim(4) in simp)
finally show ?thesis .
qed
have s24: (∑ p ∈ ?s. real (multiplicity p (24::nat)) * ln (real p)) ≤ 25 * ln 24
proof -
    have (∑ p ∈ ?s. real (multiplicity p (24::nat)) * ln (real p)) ≤ (∑ p ∈
    ?s. if p ≤ 24 then ln 24 else 0)
    by (rule sum-mono) (simp add: mult24-log-le)
    also have ... ≤ (∑ p ∈ {..24::nat}. ln 24)
    proof -
        have (∑ p ∈ ?s. if p ≤ 24 then ln 24 else (0::real)) = (∑ p ∈
        ?s ∩ {..24}. ln 24)
        using sum.inter-restrict[OF fin, of λp. ln (24::real) {..24}] by simp
        also have ... ≤ (∑ p ∈ {..24::nat}. ln 24)
        by (rule sum-mono2) (simp-all add: ln-ge-zero)
        finally show ?thesis .
    qed
    finally show ?thesis by (simp add: card-atMost)
qed
have h24: 25 * ln 24 ≤ δ * real K using elim(3) d0
    by (simp add: divide-le-eq mult.commute)
have hh: real (hof K) ≤ lam-r * real K by (rule hof-le-lam)
have hh0: 0 ≤ real (hof K) by simp
have A: (∑ p ∈ ?s. ln (real p)) + (∑ p ∈
    ?s. (real (nat-log p (5 * K)) - 1) * ln (real p)) ≤ (1 / real M + 2 * δ) *
real K
    using elim(1) elim(2) psi unfolding th
proof -
    have primes-theta (0 * real K) = 0 by (simp add: primes-theta-def prime-sum-upto-def)
    hence primes-theta (1 / real M * real K) ≤ (1 / real M + δ) * real K
    using elim(1) by simp
    thus primes-theta (1 / real M * real K) + (∑ p ∈
    ?s. (real (nat-log p (5 * K)) - 1) * ln (real p)) ≤
    (1 / real M + 2 * δ) * real K
    using elim(2) psi by (simp add: algebra-simps)
qed
let ?X = (∑ p ∈ ?s. ln (real p)) + (∑ p ∈
    ?s. (real (nat-log p (5 * K)) - 1) * ln (real p))
let ?Y = ∑ p ∈ ?s. real (multiplicity p (24::nat)) * ln (real p)

```

```

have A0: 0 ≤ (1 / real M + 2 * δ) * real K using Mr d0 Kr by simp
have 6 * real (hof K) * ?X ≤ 6 * real (hof K) * ((1 / real M + 2 * δ) * real
K)
  by (rule mult-left-mono[OF A]) simp
also have ... ≤ 6 * (lam-r * real K) * ((1 / real M + 2 * δ) * real K)
  by (rule mult-right-mono[OF - A0]) (use hh in simp)
finally have e12: 6 * real (hof K) * ?X ≤
  6 * (lam-r * real K) * ((1 / real M + 2 * δ) * real K) .
from s24 h24 have ?Y ≤ δ * real K by linarith
hence real (hof K) * ?Y ≤ real (hof K) * (δ * real K)
  by (rule mult-left-mono[OF - hh0])
also have ... ≤ (lam-r * real K) * (δ * real K)
  by (rule mult-right-mono[OF hh]) (use d0 Kr in simp)
finally have e34: real (hof K) * ?Y ≤ (lam-r * real K) * (δ * real K) .
have q: 6 * (lam-r * real K) * ((1 / real M + 2 * δ) * real K) + (lam-r *
real K) *
  (δ * real K) =
  (6 * lam-r / real M + lam-r * ε) * real K ^ 2
  unfolding δ-def using Mr by (simp add: field-simps power2-eq-square)
have le: lam-r * ε ≤ ε using e by (simp add: lam-r-def)
have K2: 0 ≤ real K ^ 2 by simp
have r: (6 * lam-r / real M + lam-r * ε) * real K ^ 2 ≤
  (6 * lam-r / real M + ε) * real K ^ 2
  by (rule mult-right-mono[OF - K2]) (use le in simp)
show ?case unfolding split using e12 e34 q r by linarith
qed
qed

```

27.2 Standing hypotheses and integrality

The standing hypotheses of Theorem 2.1.

definition *StdHyp* :: nat ⇒ nat ⇒ bool **where**
StdHyp K M ⇔ 40 ≤ M ∧ 40 dvd K ∧ 0 < K ∧
200 * M ^ 2 ≤ K

lemma *mKM-pos*: 0 < mKM K M
unfolding *mKM-def*
by (*fastforce simp: power-int-def prime-gt-0-nat intro: prod-pos*)

A rational polynomial that is p -integral at every prime has integer coefficients.

lemma *int-of-vge*:
assumes $\bigwedge p. \text{prime } p \implies \text{vge } p \ x \ 0$
shows $\exists n::\text{int}. x = \text{of-int } n$
proof (*cases* $x = 0$)
case *True*
thus ?thesis **by** (*intro exI[of - 0]*) *simp*
next

```

case False
obtain a b where ab: quotient-of x = (a, b) by (cases quotient-of x)
have b0: b > 0 using quotient-of-denom-pos[OF ab] .
have b = 1
proof (rule ccontr)
  assume b ≠ 1
  hence nat b ≠ 1 using b0 by simp
  then obtain p where p: prime p p dvd nat b using prime-factor-nat by blast
  have 0 ≤ qmultiplicity (int p) x
    using assms[OF p(1)] False by (simp add: vge-def vp-def)
  hence ¬ int p dvd snd (quotient-of x)
    by (rule qmultiplicity-nonneg-imp-not-dvd-denom) (use prime-gt-1-nat[OF
p(1)] in simp)
  moreover have int p dvd int (nat b)
    using p(2) by (simp only: int-dvd-int-iff)
  ultimately show False using ab b0 by simp
qed
thus ?thesis using quotient-of-div[OF ab] by auto
qed

```

```

lemma intPoly-of-vge:
  assumes  $\bigwedge p. \text{prime } p \implies \text{vpGge } p \ Q \ 0$ 
  shows  $\exists P :: \text{int poly. map-poly of-int } P = Q$ 
proof -
  have int:  $\exists m :: \text{int. Polynomial.coeff } Q \ n = \text{of-int } m \text{ for } n$ 
    by (rule int-of-vge) (use assms in ⟨auto simp: vpGge-def vge-def⟩)
  define f ::  $\text{rat} \Rightarrow \text{int}$  where f c = (SOME m. c = of-int m) for c
  have f0: f 0 = 0
    unfolding f-def by (rule some-equality) auto
  have fc: of-int (f (Polynomial.coeff Q n)) = Polynomial.coeff Q n for n
    unfolding f-def by (rule sym, rule someI-ex[OF int[of n]])
  show ?thesis
    by (intro exI[of - map-poly f Q] poly-eqI) (simp add: coeff-map-poly f0 fc)
qed

```

27.3 Legendre's formula and the uniformity estimates

(5.3), Legendre's formula for $v_p(S-K)$ ($K > 0$; false at $K = 0$).

```

theorem vpS-legendre:
  assumes prime p 40 dvd K 0 < K
  defines B  $\equiv 2 * \text{hof } K + K + 1$ 
  shows  $\text{vpS } K \ p = 2 * \text{int } (\text{hof } K) * (\sum_{a=1..B} \text{int } (K \text{ div } p \wedge a))$ 
     $- 12 * \text{int } (\text{hof } K) * (\sum_{a=1..B} \text{int } (\text{Nof } K \text{ div } p \wedge a))$ 
     $- 2 * (\sum_{i=1.. \text{hof } K - 1} \sum_{a=1..B} \text{int } (2 * i \text{ div } p \wedge a))$ 
     $+ (\text{int } (\text{hof } K) - 1) * \text{int } (\text{multiplicity } p \ (4 :: \text{nat}))$ 
proof -
  define h where h = hof K
  define N where N = Nof K
  have pe: prime-elem p using assms(1) by simp

```



```

have pei: prime-elem (int p) using assms(1) by (simp add: prime-nat-int-transfer)
have 40 ≤ K using assms(2,3) by (auto elim!: dvdE)
hence h1: 1 ≤ h unfolding h-def hof-def by linarith
have NK: N ≤ K unfolding N-def Nof-def by linarith
define A :: nat where A = fact K ^ (2 * h) * 4 ^ (h - 1)
define D :: nat where D = fact N ^ (12 * h) * (∏ i = 1..h - 1. fact (2 * i)
^ 2)
have A0: A ≠ 0 and D0: D ≠ 0 by (simp-all add: A-def D-def)
have SK: SK K = of-nat A / of-nat D by (simp add: SK-def A-def D-def h-def
N-def)
have v: vpS K p = int (multiplicity p A) - int (multiplicity p D)
  unfolding vpS-def vp-def SK using A0 D0 pei
  by (simp add: qmultiplicity-divide qmultiplicity-of-nat multiplicity-int)
have mA: multiplicity p A = 2 * h * multiplicity p (fact K) + (h - 1) *
multiplicity p (4::nat)
  unfolding A-def using pe
  by (simp add: prime-elem-multiplicity-mult-distrib prime-elem-multiplicity-power-distrib)
have mD: multiplicity p D = 12 * h * multiplicity p (fact N)
  + (∑ i = 1..h - 1. 2 * multiplicity p (fact (2 * i)))
proof -
  have multiplicity p (∏ i = 1..h - 1. fact (2 * i) ^ 2) = (∑ i =
  1..h - 1. multiplicity p (fact (2 * i) ^ 2))
  by (rule prime-elem-multiplicity-prod-distrib[OF pe]) auto
  also have ... = (∑ i = 1..h - 1. 2 * multiplicity p (fact (2 * i)))
  by (rule sum.cong) (use pe in ⟨simp-all add:
  prime-elem-multiplicity-power-distrib⟩)
  finally have P: multiplicity p (∏ i = 1..h - 1. fact (2 * i) ^ 2) = (∑ i =
  1..h - 1. 2 * multiplicity p (fact (2 * i))) .
  show ?thesis unfolding D-def using pe
  by (subst prime-elem-multiplicity-mult-distrib) (simp-all add: P[unfolded
One-nat-def]
  prime-elem-multiplicity-power-distrib)
qed
have LK: multiplicity p (fact K) = (∑ a = 1..B. K div p ^ a)
  by (rule legendre-sum[OF assms(1)]) (simp add: B-def)
have LN: multiplicity p (fact N) = (∑ a = 1..B. N div p ^ a)
  by (rule legendre-sum[OF assms(1)]) (use NK in ⟨simp add: B-def⟩)
have Li: multiplicity p (fact (2 * i)) = (∑ a = 1..B. 2 * i div p ^ a)
  if i ∈ {1..h - 1} for i
  by (rule legendre-sum[OF assms(1)]) (use that in ⟨auto simp: B-def h-def⟩)
have mD': multiplicity p D = 12 * h * (∑ a = 1..B. N div p ^ a)
  + (∑ i = 1..h - 1. 2 * (∑ a = 1..B. 2 * i div p ^ a))
  unfolding mD LN by (simp add: Li)
have iA: int (multiplicity p A) = 2 * int h * (∑ a =
  1..B. int (K div p ^ a)) + (int h - 1) * int (multiplicity p (4::nat))
  unfolding mA LK of-nat-add of-nat-mult of-nat-diff[OF h1] of-nat-sum by
simp
have iD: int (multiplicity p D) = 12 * int h * (∑ a = 1..B. int (N div p ^ a))
  + 2 * (∑ i = 1..h - 1. ∑ a = 1..B. int (2 * i div p ^ a))

```

```

    unfolding mD' of-nat-add of-nat-mult of-nat-sum by (simp add: sum-distrib-left)
  show ?thesis unfolding v iA iD h-def N-def by simp
qed

```

27.4 Proposition 5.1

```

lemma vp-prime-powi:
  assumes prime p prime q
  shows vp p (of-nat q powi k) = (if q = p then k else 0)
proof -
  have q0: (of-nat q :: rat) ≠ 0 using assms(2) by (simp add: prime-gt-0-nat)
  have vp p (of-nat q) = (if q = p then 1 else 0)
  using assms by (auto simp: vp-def qmultiplicity-of-nat multiplicity-int prime-multiplicity-other
    multiplicity-prime)
  thus ?thesis using vp-powi-rat[OF assms(1) q0, of k] by simp
qed

```

```

lemma vp-mKM:
  assumes prime p
  shows vp p (mKM K M) = (if p ≤ 2 * hof K then - Lp K M p else 0)
proof -
  let ?Q = {q. q ≤ 2 * hof K ∧ prime q}
  have vp p (mKM K M) = (∑ q ∈ ?Q. vp p (of-nat q powi (- Lp K M q)))
  unfolding mKM-def by (rule vp-prod-rat[OF assms]) (auto simp: prime-gt-0-nat)
  also have ... = (∑ q ∈ ?Q. if q = p then - Lp K M q else 0)
  by (rule sum.cong) (use assms in ⟨auto simp: vp-prime-powi⟩)
  also have ... = (if p ≤ 2 * hof K then - Lp K M p else 0)
  using assms by (simp add: sum.delta)
  finally show ?thesis .
qed

```

```

lemma vpS-eq-0-large:
  assumes prime p 40 dvd K 0 < K 2 * hof K < p
  shows vpS K p = 0
proof -
  define B where B = 2 * hof K + K + 1
  have K40: 40 ≤ K using assms(2,3) by (auto elim!: dvdE)
  obtain k where k: K = 40 * k using assms(2) by (elim dvdE)
  have hk: hof K = 37 * k unfolding k hof-def by simp
  have Kp: K < p using assms(4) hk k by linarith
  have NK: Nof K ≤ K unfolding Nof-def by linarith
  have p1: p ≤ p ^ a if 1 ≤ a for a
    using that prime-gt-1-nat[OF assms(1)] by (simp add: self-le-power)
  have z1: K div p ^ a = 0 if a ∈ {1..B} for a using that p1[of a] Kp
    by (auto intro!: div-less)
  have z2: Nof K div p ^ a = 0 if a ∈ {1..B} for a using that p1[of a] Kp NK
    by (auto intro!: div-less)
  have z3: 2 * i div p ^ a = 0 if i ∈ {1..hof K - 1} a ∈ {1..B} for i a
    using that p1[of a] assms(4) by (auto intro!: div-less)

```

```

have p4:  $\neg p \text{ dvd } 4$ 
proof
  assume p dvd 4
  hence  $p \leq 4$  by (rule dvd-imp-le) simp
  thus False using assms(4) K40 hk k by linarith
qed
have m4: multiplicity p (4::nat) = 0 by (rule not-dvd-imp-multiplicity-0[OF
p4])
show ?thesis
  unfolding vpS-legendre[OF assms(1-3), folded B-def] using z1 z2 z3 m4 by
simp
qed

lemma QKM-vpGge:
  assumes StdHyp K M prime p
  shows vpGge p (QKM K M) 0
proof -
  from assms(1) have M40:  $40 \leq M$  and K40d:  $40 \text{ dvd } K$  and K0:  $0 <$ 
    K and KM:  $200 * M \wedge 2 \leq K$ 
  by (auto simp: StdHyp-def)
  obtain k where k:  $K = 40 * k$  using K40d by (elim dvdE)
  have hk:  $\text{hof } K = 37 * k$  unfolding k hof-def Nof-def by simp-all
  have K2h:  $K \leq 2 * \text{hof } K$  using hk k by simp
  have SKi: vpGge p [:SK K:] (of-int (vpS K p)) unfolding vpS-def by (rule
vpGge-C-vp)
  have F: vpGge p (FK K) (of-int (Lp K M p))
  proof (cases  $p * M \leq K$ )
    case True
      have vpGge p (FK K) ( $- 6 * \text{of-nat } (\text{hof } K) * \text{of-nat } (\text{nat-log } p (5 * K))$ 
         $- \text{of-nat } (\text{hof } K) * \text{of-nat } (\text{multiplicity } p (24::\text{nat}))$ )
        by (rule vpG-F-ge[OF assms(2) K40d])
      thus ?thesis using True by (simp add: Lp-def)
    next
      case nb1: False
      show ?thesis
      proof (cases  $3 * p \leq K$ )
        case True
          have IH: InnerHyp K M p using M40 K40d K0 KM nb1 True by (auto simp:
InnerHyp-def)
          obtain n where n:  $\text{gammaIn } K M p = \text{of-int } n$  using gammaIn-isInt by
blast
          have vpGge p ([:SK K:] * DeltaK K) (of-int (vpS K p) + gammaIn K M p)
            by (rule vpGge-mul[OF assms(2) SKi DeltaK-vpGge-inner[OF assms(2)
IH]])
          thus ?thesis using nb1 True n by (simp add: Lp-def FK-def)
        next
          case nb2: False
          show ?thesis
          proof (cases  $p \leq K$ )

```

```

    case True
    have OH: OuterHyp K p using M40 K40d KM assms(2) nb2 True
    by (intro outer-hyp-of-range) auto
    have vpGge p ([:SK K:] * DeltaK K) (of-int (vpS K p) + of-int (gammaOut
K p))
    by (rule vpGge-mul[OF assms(2) SKi DeltaK-vpGge-outer[OF assms(2)
K40d OH]])
    thus ?thesis using nb1 nb2 True by (simp add: Lp-def FK-def)
  next
  case False
  have vpGge p ([:SK K:] * DeltaK K) (of-int (vpS K p) + 0)
  by (rule vpGge-mul[OF assms(2) SKi DeltaK-vpGge-large-prime[OF
assms(2) K40d]]) (use False
  in simp)
  thus ?thesis using nb1 nb2 False by (simp add: Lp-def FK-def)
qed
qed
qed
have m: vpGge p [:mKM K M:] (of-int (vp p (mKM K M))) by (rule vpGge-C-vp)
have Q: vpGge p (QKM K M) (of-int (vp p (mKM K M)) + of-int (Lp K M p))
  unfolding QKM-def by (rule vpGge-mul[OF assms(2) m F])
have 0 ≤ vp p (mKM K M) + Lp K M p
proof (cases p ≤ 2 * hof K)
  case True
  thus ?thesis by (simp add: vp-mKM[OF assms(2)])
next
  case False
  hence K < p using K2h by linarith
  moreover have p ≤ p * M using M40 by simp
  ultimately have ¬ p * M ≤ K ∧ 3 * p ≤ K ∧ p ≤ K by linarith+
  hence Lp K M p = vpS K p by (simp add: Lp-def)
  also have ... = 0 using False by (intro vpS-eq-0-large[OF assms(2) K40d
K0]) simp
  finally show ?thesis using False by (simp add: vp-mKM[OF assms(2)])
qed
thus ?thesis using Q by (intro vpGge-mono[OF Q]) simp
qed

```

Proposition 5.1: $Q\text{-}\{K, M\} \in \mathbb{Z}[X]$. At each prime, $L\text{-}p(K, M)$ is covered by (3.12), Proposition 4.1 or Proposition 4.3.

theorem *QKM-int-poly*: $\text{StdHyp } K \ M \implies$
 $\exists P :: \text{int poly. map-poly of-int } P = QKM \ K \ M$
by (rule *intPoly-of-vge*) (rule *QKM-vpGge*)

(5.7), inner range: $\gamma\text{-}p\hat{\text{in}} = p\Gamma(K/p) + O\text{-}M(1)$ and $v\text{-}p(S\text{-}K) = pN(K/p) + O\text{-}M(1)$, uniformly over (4.1) (the Lean constant is $1000M^{\wedge}2$).

27.5 The outer range of the uniformity estimate (Lean: Legendre5)

For $n < p^2$ only $a = 1$ contributes to Legendre's sum.

```
lemma sum-div-pow-single:
  assumes prime p 1 ≤ B n < (p :: nat) ^ 2
  shows (∑ a = 1..B. n div p ^ a) = n div p
proof -
  have z: n div p ^ a = 0 if a ∈ {1..B} a ≠ 1 for a
  proof -
    have a2: 2 ≤ a using that by auto
    have p ^ 2 ≤ p ^ a by (rule power-increasing[OF a2]) (use prime-ge-1-nat[OF
assms(1)] in
      simp)
    thus ?thesis using assms(3) by (intro div-less) linarith
  qed
  show ?thesis
    using assms(2) z by (subst sum.remove[of - 1]) auto
qed
```

Under (4.9), Legendre's formula collapses to a single term per factorial:
 $v_p(S-K) = 2h \lfloor K/p \rfloor - 2 \sum_{i=1}^{\lfloor h-1 \rfloor} \lfloor 2i/p \rfloor$.

```
lemma vpS-outer:
  assumes p: prime p and K: 40 dvd K and h: OuterHyp K p
  shows vpS K p = 2 * int (hof K) * int (K div p) - 2 * int (∑ i = 1..hof K -
1. 2 * i div p)
proof -
  have facts: 7 ≤ p p ≤ K K < 3 * p 2 * K < p ^ 2 2 * Nof K < p
  using h unfolding OuterHyp-def by auto
  have K0: 0 < K using facts by linarith
  define B where B = 2 * hof K + K + 1
  have B1: 1 ≤ B unfolding B-def by simp
  have sK: (∑ a = 1..B. K div p ^ a) = K div p
  by (rule sum-div-pow-single[OF p B1]) (use facts in linarith)
  have sN: (∑ a = 1..B. Nof K div p ^ a) = Nof K div p
  proof -
    have Nof K ≤ K unfolding Nof-def by linarith
    thus ?thesis using facts by (intro sum-div-pow-single[OF p B1]) linarith
  qed
  have si: (∑ a = 1..B. 2 * i div p ^ a) = 2 * i div p if i ∈ {1..hof K - 1} for i
  proof -
    have hof K ≤ K unfolding hof-def by linarith
    moreover have 1 ≤ i i ≤ hof K - 1 using that by auto
    ultimately have 2 * i < p ^ 2 using facts(4) by arith
    thus ?thesis by (rule sum-div-pow-single[OF p B1])
  qed
  have N0: Nof K div p = 0 using facts by simp
  have m4: multiplicity p (4::nat) = 0
  proof -
```

```

have  $\neg p \text{ dvd } 4$ 
proof
  assume  $p \text{ dvd } 4$ 
  hence  $p \leq 4$  by (rule dvd-imp-le) simp
  thus False using facts by linarith
qed
thus ?thesis by (rule not-dvd-imp-multiplicity-0)
qed
have leg:  $vpS \ K \ p = 2 * \text{int} \ (hof \ K) * \text{int} \ (K \text{ div } p) - 12 * \text{int} \ (hof \ K) * \text{int} \ (Nof \ K \text{ div } p)$ 
 $- 2 * \text{int} \ (\sum \ i = 1..hof \ K - 1. \ 2 * i \text{ div } p) + (\text{int} \ (hof \ K) - 1) * \text{int} \ (multiplicity \ p \ (4::nat))$ 
proof -
  have L:  $vpS \ K \ p = 2 * \text{int} \ (hof \ K) * (\sum \ a = 1..B. \ \text{int} \ (K \text{ div } p \wedge a))$ 
 $- 12 * \text{int} \ (hof \ K) * (\sum \ a = 1..B. \ \text{int} \ (Nof \ K \text{ div } p \wedge a))$ 
 $- 2 * (\sum \ i = 1..hof \ K - 1. \ \sum \ a = 1..B. \ \text{int} \ (2 * i \text{ div } p \wedge a))$ 
 $+ (\text{int} \ (hof \ K) - 1) * \text{int} \ (multiplicity \ p \ (4::nat))$ 
  unfolding B-def by (rule vpS-legendre[OF p K K0])
  have s1:  $(\sum \ a = 1..B. \ \text{int} \ (K \text{ div } p \wedge a)) = \text{int} \ (K \text{ div } p)$ 
  unfolding of-nat-sum[symmetric] sK ..
  have s2:  $(\sum \ a = 1..B. \ \text{int} \ (Nof \ K \text{ div } p \wedge a)) = \text{int} \ (Nof \ K \text{ div } p)$ 
  unfolding of-nat-sum[symmetric] sN ..
  have s3:  $(\sum \ i = 1..hof \ K - 1. \ \sum \ a = 1..B. \ \text{int} \ (2 * i \text{ div } p \wedge a)) = \text{int} \ (\sum$ 
 $i =$ 
 $1..hof \ K - 1. \ 2 * i \text{ div } p)$ 
  proof -
    have  $(\sum \ i = 1..hof \ K - 1. \ \sum \ a = 1..B. \ \text{int} \ (2 * i \text{ div } p \wedge a)) = (\sum \ i =$ 
 $1..hof \ K - 1. \ \text{int} \ (2 * i \text{ div } p))$ 
    by (rule sum.cong[OF refl]) (simp only: of-nat-sum[symmetric] si)
    thus ?thesis by simp
  qed
  show ?thesis unfolding L s1 s2 s3 ..
qed
show ?thesis unfolding leg N0 m4 by simp
qed

 $\sum \ -\{i=1\} \wedge \{h-1\} \lfloor 2i/p \rfloor = \sum \ -\{j=1\} \wedge \{5\} (h - \lceil jp/2 \rceil) -$  when  $2h < 6p$ :
count the  $j \geq 1$  with  $jp \leq 2i$  and swap the sums.

lemma sum-if-one-card:  $\text{finite } A \implies (\sum \ x \in A. \text{if } P \ x \text{ then } (1::nat) \text{ else } 0) = \text{card} \ \{x \in A. \ P \ x\}$ 
by (simp add: sum.inter-filter[symmetric])

lemma half-ceil-le-iff:  $((m + 1) \text{ div } 2 \leq (i::nat)) = (m \leq 2 * i)$ 
by linarith

lemma suc-pred-sub:  $1 \leq x \implies \text{Suc} \ (h - 1) - x = h - (x::nat)$ 
by arith

lemma half-ceil-ge-1:  $1 \leq (m::nat) \implies 1 \leq (m + 1) \text{ div } 2$ 

```

by *fastforce*

lemma *sum-floor-eq*:

assumes *p*: prime *p* **and** *hh*: $2 * h < 6 * (p :: nat)$

shows $(\sum i = 1..h - 1. 2 * i \text{ div } p) = (\sum j = 1..5. h - (j * p + 1) \text{ div } 2)$

proof –

have *p0*: $0 < p$ **using** *p* **by** (*simp add: prime-gt-0-nat*)

have *cnt*: $2 * i \text{ div } p = (\sum j = 1..5. \text{if } j * p \leq 2 * i \text{ then } 1 \text{ else } 0)$

if $i \in \{1..h - 1\}$ **for** *i*

proof –

have *le5*: $2 * i \text{ div } p \leq 5$

proof –

have $1 \leq i \leq h - 1$ **using** *that* **by** *auto*

hence $2 * i < 6 * p$ **using** *hh* **by** *arith*

hence $2 * i \text{ div } p < 6$ **using** *p0* **by** (*simp add: div-less-iff-less-mult mult.commute*)

thus *?thesis* **by** *simp*

qed

have $\{j \in \{1..5\}. j * p \leq 2 * i\} = \{1..2 * i \text{ div } p\}$

proof –

have *iff*: $\bigwedge j. (j \leq 2 * i \text{ div } p) = (j * p \leq 2 * i)$ **using** *p0*

by (*simp add: less-eq-div-iff-mult-less-eq*)

have *j5*: $\bigwedge j. j * p \leq 2 * i \implies j \leq 5$ **using** *iff le5*

by (*meson order-trans*)

show *?thesis* **using** *iff j5* **by** *auto*

qed

hence $\text{card } \{j \in \{1..5\}. j * p \leq 2 * i\} = 2 * i \text{ div } p$ **by** *simp*

thus *?thesis*

by (*simp add: sum-if-one-card*)

qed

have $(\sum i = 1..h - 1. 2 * i \text{ div } p) = (\sum i = 1..h - 1. \sum j = 1..5. \text{if } j * p \leq 2 * i \text{ then } 1 \text{ else } 0)$

by (*rule sum.cong[OF refl]*) (*simp add: cnt*)

also have $\dots = (\sum j = 1..5. \sum i = 1..h - 1. \text{if } j * p \leq 2 * i \text{ then } 1 \text{ else } 0)$

by (*rule sum.swap*)

also have $\dots = (\sum j = 1..5. h - (j * p + 1) \text{ div } 2)$

proof (*intro sum.cong refl*)

fix *j* :: nat **assume** *j*: $j \in \{1..5\}$

have *jp*: $1 \leq j * p$ **using** *j p0* **by** (*simp add: Suc-le-eq*)

have $\{i \in \{1..h - 1\}. j * p \leq 2 * i\} = \{(j * p + 1) \text{ div } 2..h - 1\}$

proof –

have *lb*: $\bigwedge i. (j * p + 1) \text{ div } 2 \leq i \implies 1 \leq i$

using *half-ceil-ge-1[OF jp]* **by** *linarith*

show *?thesis* **using** *lb* **by** (*auto simp: half-ceil-le-iff[symmetric]*)

qed

thus $(\sum i = 1..h - 1. \text{if } j * p \leq 2 * i \text{ then } 1 \text{ else } 0) = h - (j * p + 1) \text{ div } 2$

proof –

assume *e*: $\{i \in \{1..h - 1\}. j * p \leq 2 * i\} = \{(j * p + 1) \text{ div } 2..h - 1\}$

have $(\sum i = 1..h - 1. \text{if } j * p \leq 2 * i \text{ then } 1 \text{ else } 0) =$

$\text{card } \{(j * p + 1) \text{ div } 2..h - 1\}$

```

    unfolding e[symmetric] by (rule sum-if-one-card) simp
    also have ... = h - (j * p + 1) div 2 unfolding card-atLeastAtMost
    by (rule suc-pred-sub[OF half-ceil-ge-1[OF jp]])
    finally show ?thesis .
  qed
qed
finally show ?thesis .
qed

```

The integer K ($R-0(p/K) - d(p/K)$), written with $N = \text{Nof } K$.

definition $Eout :: nat \Rightarrow nat \Rightarrow int$ **where**

```

  Eout K p = (let N = int (Nof K); K' = int K; p' = int p in
    if K < 2 * p
    then 7 * (K' - p') - 6 * min N (K' - p') - 6 * max (K' + N - 2 * p') 0
    + max (K' + 4 * N - 2 * p') 0
    else 8 * K' - 9 * p' - 8 * N - 5 * min N (K' - 2 * p') - 5 * max (K' +
    N - 3 * p') 0
    - max (K' + 4 * N - 3 * p' - max (K' + N - 3 * p') 0) 0)

```

$p < K$ under (4.9) with $40 \mid K$: $p = K$ would make $40 \mid p$.

lemma *outer-p-less*:

```

  assumes p: prime p and K: 40 dvd K and h: OuterHyp K p
  shows p < K

```

proof –

```

  have f: 7 ≤ p p ≤ K using h unfolding OuterHyp-def by auto
  have p ≠ K

```

proof

```

  assume p = K
  hence 2 dvd p using K by (metis dvd-trans dvd-triv-left mult-2 numeral-Bit0)
  hence (2::nat) = 1 ∨ 2 = p using p unfolding prime-nat-iff by auto
  hence 2 = p by linarith
  thus False using f by linarith

```

qed

```

  thus ?thesis using f by simp

```

qed

lemma *odd-prime-ge-7*: $\text{prime } p \implies 7 \leq (p :: nat) \implies$

```

  ¬ 2 dvd p

```

```

  using prime-odd-nat[of p] by linarith

```

$|- \gamma \cdot p \hat{=} out - E \cdot out| \leq 7$: pure integer arithmetic with $K \bmod p = K - p$
resp. $K - 2p$.

lemma *int-nat-sub*: $\text{int } (a - b) = \max (\text{int } a - \text{int } b) 0$

```

  by (cases b ≤ a) (simp-all add: of-nat-diff)

```

lemma *int-min-nat*: $\text{int } (\min a b) = \min (\text{int } a) (\text{int } b)$

```

  by (simp add: min-def)

```



```

lemma gammaOut-sub-Eout:
  assumes p: prime p and K: 40 dvd K and h: OuterHyp K p
  shows  $|- \text{gammaOut } K \text{ } p - \text{Eout } K \text{ } p| \leq 7$ 
proof -
  have facts:  $7 \leq p \leq K \ K < 3 * p \ 2 * \text{Nof } K < p \ 5 * \text{Nof } K \leq 2 * p - 2$ 
    using h unfolding OuterHyp-def by auto
  have pK:  $p < K$  by (rule outer-p-less[OF p K h])
  show ?thesis
  proof (cases  $K < 2 * p$ )
    case True
    have  $K \text{ div } p = 1$  and m:  $K \text{ mod } p = K - p$ 
    proof -
      show  $d: K \text{ div } p = 1$  by (rule div-nat-eqI) (use facts True in auto)
      show  $K \text{ mod } p = K - p$  using minus-div-mult-eq-mod[of K p] d by simp
    qed
    have n40:  $40 * \text{Nof } K \leq 3 * K \ 3 * K < 40 * \text{Nof } K + 40$  unfolding Nof-def
by linarith+
    have i:  $7 \leq \text{int } p \ \text{int } p < \text{int } K \ \text{int } K < 2 * \text{int } p \ 2 * \text{int } (\text{Nof } K) < \text{int } p$ 
       $40 * \text{int } (\text{Nof } K) \leq 3 * \text{int } K \ 3 * \text{int } K < 40 * \text{int } (\text{Nof } K) + 40$ 
    using facts pK True n40 by simp-all
    show ?thesis
    unfolding gammaOut-def rOut-def Eout-def Let-def if-P[OF True] m
    using i unfolding int-nat-sub int-min-nat of-nat-add of-nat-mult of-nat-1
of-nat-numeral
    min-def max-def abs-if
    by (auto split: if-splits)
  next
  case False
  have  $K \text{ div } p = 2$  and m:  $K \text{ mod } p = K - 2 * p$ 
  proof -
    show  $d: K \text{ div } p = 2$  by (rule div-nat-eqI) (use facts False in auto)
    show  $K \text{ mod } p = K - 2 * p$  using minus-div-mult-eq-mod[of K p] d by (simp
add: mult.commute)
  qed
  have n40:  $40 * \text{Nof } K \leq 3 * K \ 3 * K < 40 * \text{Nof } K + 40$  unfolding Nof-def
by linarith+
  have i:  $7 \leq \text{int } p \ \text{int } p < \text{int } K \ 2 * \text{int } p \leq \text{int } K \ \text{int } K <$ 
     $3 * \text{int } p \ 2 * \text{int } (\text{Nof } K) < \text{int } p$ 
     $40 * \text{int } (\text{Nof } K) \leq 3 * \text{int } K \ 3 * \text{int } K < 40 * \text{int } (\text{Nof } K) + 40$ 
  using facts pK False n40 by simp-all
  show ?thesis
  unfolding gammaOut-def rOut-def Eout-def Let-def if-not-P[OF False] m
  using i unfolding int-nat-sub int-min-nat of-nat-add of-nat-mult of-nat-1
of-nat-numeral
  min-def max-def abs-if
  by (auto split: if-splits)
  qed
qed

```

lemma *hof-real*: $40 \text{ dvd } K \implies \text{real } (\text{hof } K) = \text{lam-r} * \text{real } K$

proof –

assume $40 \text{ dvd } K$

then obtain n **where** $n: K = 40 * n$ **by** (*elim dvdE*)

have $\text{hof } K = 37 * n$ **unfolding** *hof-def* n **by** *auto*

thus *?thesis* **unfolding** n *lam-r-def* **by** *simp*

qed

lemma *Nof-real*: $40 \text{ dvd } K \implies \text{real } (\text{Nof } K) = \text{alpha-r} * \text{real } K$

proof –

assume $40 \text{ dvd } K$

then obtain n **where** $n: K = 40 * n$ **by** (*elim dvdE*)

have $\text{Nof } K = 3 * n$ **unfolding** *Nof-def* n **by** *auto*

thus *?thesis* **unfolding** n *alpha-r-def* **by** *simp*

qed

lemma *hof-le*: $\text{hof } K \leq K$ **and** *Nof-le*: $\text{Nof } K \leq K$ **and** *Nof-lt-hof*: $2 \leq K \implies \text{Nof } K < \text{hof } K$

proof –

show $\text{hof } K \leq K$ **unfolding** *hof-def* **by** *linarith*

show $\text{Nof } K \leq K$ **unfolding** *Nof-def* **by** *linarith*

show $2 \leq K \implies \text{Nof } K < \text{hof } K$ **unfolding** *Nof-def* *hof-def* **by** *linarith*

qed

The real identity behind *E-out*.

lemma *mult-min-nonneg*:

assumes $0 \leq (c::\text{real})$ **shows** $c * \min a b = \min (c * a) (c * b)$

proof (*cases a ≤ b*)

case *True*

hence $c * a \leq c * b$ **using** *assms* **by** (*rule mult-left-mono*)

thus *?thesis* **using** *True* **by** (*simp add: min.absorb1*)

next

case *False*

hence $ba: b \leq a$ **by** *simp*

hence $c * b \leq c * a$ **using** *assms* **by** (*rule mult-left-mono*)

thus *?thesis* **using** *ba* **by** (*simp add: min.absorb2*)

qed

lemma *mult-pos-r-nonneg*:

assumes $0 \leq (c::\text{real})$ **shows** $c * \text{pos-r } a = \max (c * a) 0$

proof (*cases a ≤ 0*)

case *True*

hence $c * a \leq 0$ **using** *assms* **by** (*simp add: mult-nonneg-nonpos*)

thus *?thesis* **using** *True* **unfolding** *pos-r-def* **by** (*simp add: max.absorb2*)

next

case *False*

hence $0 \leq c * a$ **using** *assms* **by** *simp*

thus *?thesis* **using** *False* **unfolding** *pos-r-def* **by** (*simp add: max.absorb1*)

qed

lemma *K-mul-R0-sub-dfun*:

assumes *p*: prime *p* **and** *K*: 40 dvd *K* **and** *h*: OuterHyp *K* *p*

shows $\text{real } K * (R0\text{-}r \text{ (real } p / \text{real } K) - \text{dfun-r (real } p / \text{real } K)) = \text{of-int (Eout } K \text{ } p)$

proof –

have *facts*: $7 \leq p$ $p \leq K$ $K < 3 * p$ **using** *h* **unfolding** *OuterHyp-def* **by** *auto*

have *pK*: $p < K$ **by** (rule *outer-p-less*[*OF* *p* *K* *h*])

have *K0*: $0 < \text{real } K$ **using** *facts* **by** *simp*

define *y* **where** $y = \text{real } p / \text{real } K$

have *Ky*: $\text{real } K * y = \text{real } p$ **unfolding** *y-def* **using** *K0* **by** *simp*

have *N*: $\text{real (Nof } K) = \alpha\text{-r } * \text{real } K$ **by** (rule *Nof-real*[*OF* *K*])

have *Kmin*: $\text{real } K * \min a \ b = \min (\text{real } K * a) (\text{real } K * b)$ **for** *a* *b*

by (rule *mult-min-nonneg*) *simp*

have *Kpos*: $\text{real } K * \text{pos-r } a = \max (\text{real } K * a) \ 0$ **for** *a*

by (rule *mult-pos-r-nonneg*) *simp*

have *pKr*: $\text{real } p < \text{real } K$ **and** *K3*: $\text{real } K < 3 * \text{real } p$ **using** *pK* *facts* **by** *simp-all*

show *?thesis*

proof (cases $K < 2 * p$)

case *True*

hence *y1*: $1 / 2 < y$ $y < 1$ **and** *ny*: $\neg (1 / 3 < y \wedge y < 1 / 2)$

proof –

have $\text{real } K < \text{real } (2 * p)$ **using** *True* **by** (simp only: *of-nat-less-iff*)

hence *K2*: $\text{real } K < 2 * \text{real } p$ **by** *simp*

show *a*: $1 / 2 < y$ **unfolding** *y-def* **using** *K0* *K2* **by** (simp add: *field-simps*)

show $y < 1$ **unfolding** *y-def* **using** *K0* *pKr* **by** (simp add: *field-simps*)

show $\neg (1 / 3 < y \wedge y < 1 / 2)$ **using** *a* **by** (simp add: *not-less less-imp-le*)

qed

have $\text{real } K * (R0\text{-}r \ y - \text{dfun-r } y) =$

$\text{real } K * (7 * (1 - y) - 6 * \min \alpha\text{-r } (1 - y) - 6 * \text{pos-r } (1 +$

$\alpha\text{-r } - 2 * y) +$

$\text{pos-r } (1 + 4 * \alpha\text{-r } - 2 * y))$

unfolding *R0-r-def* *dfun-r-def* **using** *y1* *ny* **by** *simp*

also have $\dots = \text{of-int (Eout } K \text{ } p)$

proof –

have *ga*: $\text{real } K * \alpha\text{-r } = \text{real (Nof } K)$ **using** *N* **by** *simp*

have *g1*: $\text{real } K * (1 - y) = \text{real } K - \text{real } p$ **unfolding** *y-def* **using** *K0*

by (simp add: *field-simps*)

have *g2*: $\text{real } K * (1 + \alpha\text{-r } - 2 * y) = \text{real } K + \text{real (Nof } K) - 2 * \text{real } p$

unfolding *y-def* **using** *K0* *N* **by** (simp add: *field-simps*)

have *g3*: $\text{real } K * (1 + 4 * \alpha\text{-r } - 2 * y) = \text{real } K + 4 * \text{real (Nof } K)$

$- 2 * \text{real } p$

unfolding *y-def* **using** *K0* *N* **by** (simp add: *field-simps*)

have $\text{real } K * (7 * (1 - y) - 6 * \min \alpha\text{-r } (1 - y) - 6 * \text{pos-r } (1 +$

$\alpha\text{-r } - 2 * y) +$

$\text{pos-r } (1 + 4 * \alpha\text{-r } - 2 * y)) =$

$7 * (\text{real } K * (1 - y)) - 6 * (\text{real } K * \min \alpha\text{-r } (1 - y)) - 6 * (\text{real } K * \text{pos-r } (1 +$

$\alpha\text{-r } - 2 * y)) +$

$6 * (\text{real } K * \text{pos-r } (1 + 4 * \alpha\text{-r } - 2 * y))$

```

K * pos-r (1 +
  alpha-r - 2 * y))
+ real K * pos-r (1 + 4 * alpha-r - 2 * y) by (simp add: algebra-simps)
also have ... = 7 * (real K - real p) - 6 * min (real (Nof K)) (real K -
real p)
- 6 * max (real K + real (Nof K) - 2 * real p) 0 + max (real K + 4
* real (Nof K) - 2 *
  real p) 0
unfolding Kmin Kpos g1 g2 g3 ga ..
also have ... = of-int (Eout K p) unfolding Eout-def Let-def if-P[OF True]
by (simp add: of-int-min of-int-max)
finally show ?thesis .
qed
finally show ?thesis unfolding y-def .
next
case False
hence y1: 1 / 3 < y y < 1 / 2
proof -
have odd: ¬ 2 dvd p by (rule odd-prime-ge-7[OF p facts(1)])
have ne: K ≠ 2 * p using K odd by force
have 2 * p < K using False ne by linarith
hence real (2 * p) < real K by (simp only: of-nat-less-iff)
hence K2: 2 * real p < real K by simp
show 1 / 3 < y unfolding y-def using K0 K3 by (simp add: field-simps)
show y < 1 / 2 unfolding y-def using K0 K2 by (simp add: field-simps)
qed
have real K * (R0-r y - dfun-r y) =
  real K * ((8 - 9 * y - 8 * alpha-r - 5 * min alpha-r (1 - 2 * y) - 5
* pos-r (1 +
  alpha-r - 3 * y))
  - pos-r ((1 + 4 * alpha-r - 3 * y) - pos-r (1 + alpha-r - 3
* y)))
unfolding R0-r-def dfun-r-def using y1 by simp
also have ... = of-int (Eout K p)
proof -
have ga: real K * alpha-r = real (Nof K) using N by simp
have Ky': real K * y = real p by (rule Ky)
have h1: real K * (1 - 2 * y) = real K - 2 * real p unfolding y-def using
K0
by (simp add: field-simps)
have h2: real K * (1 + alpha-r - 3 * y) = real K + real (Nof K) - 3 * real
p
unfolding y-def using K0 N by (simp add: field-simps)
have h3: real K * (1 + 4 * alpha-r - 3 * y) = real K + 4 * real (Nof K)
- 3 * real p
unfolding y-def using K0 N by (simp add: field-simps)
have h3': real K * ((1 + 4 * alpha-r - 3 * y) - pos-r (1 + alpha-r - 3 *
y)) =
  real K + 4 * real (Nof K) - 3 * real p - max (real K + real (Nof

```

$K) - 3 * \text{real } p) \ 0$
by (*simp only: right-diff-distrib*[*of real* $K \ 1 + 4 * \text{alpha-r} - 3 * y$] $K\text{pos } h2$ $h3$)
have $\text{real } K * ((8 - 9 * y - 8 * \text{alpha-r} - 5 * \min \text{alpha-r } (1 - 2 * y) - 5 * \text{pos-r } (1 + \text{alpha-r} - 3 * y)) - \text{pos-r } ((1 + 4 * \text{alpha-r} - 3 * y) - \text{pos-r } (1 + \text{alpha-r} - 3 * y))) =$
 $8 * \text{real } K - 9 * (\text{real } K * y) - 8 * (\text{real } K * \text{alpha-r}) - 5 * (\text{real } K * \min \text{alpha-r } (1 - 2 * y)) - 5 * (\text{real } K * \text{pos-r } (1 + \text{alpha-r} - 3 * y)) - \text{real } K * \text{pos-r } ((1 + 4 * \text{alpha-r} - 3 * y) - \text{pos-r } (1 + \text{alpha-r} - 3 * y))$
by (*simp add: algebra-simps*)
also have $\dots = 8 * \text{real } K - 9 * \text{real } p - 8 * \text{real } (\text{Nof } K) - 5 * \min (\text{real } (\text{Nof } K)) (\text{real } K - 2 * \text{real } p) - 5 * \max (\text{real } K + \text{real } (\text{Nof } K) - 3 * \text{real } p) \ 0$
 $- \max (\text{real } K + 4 * \text{real } (\text{Nof } K) - 3 * \text{real } p - \max (\text{real } K + \text{real } (\text{Nof } K) - 3 * \text{real } p) \ 0)$
unfolding $K\text{min } K\text{pos } h1 \ h2 \ h3' \ \text{ga } Ky' \dots$
also have $\dots = \text{of-int } (E\text{out } K \ p) \text{ unfolding } E\text{out-def } Let\text{-def } \text{if-not-}P[OF \text{ False}]$
by (*simp add: of-int-min of-int-max*)
finally show *?thesis* .
qed
finally show *?thesis* **unfolding** $y\text{-def}$.
qed
qed

$-v\text{-}p(S\text{-}K)$ against its limit (5.7, outer): an explicit sum of five bounded terms.

lemma *vpS-sub-limit:*

assumes p : *prime* p **and** K : $40 \text{ dvd } K$ **and** h : *OuterHyp* $K \ p$
shows $-\text{of-int } (vpS \ K \ p) - \text{real } K * (-2 * \text{lam-r} * \text{of-int } (\text{floor } (\text{real } K / \text{real } p)) +$
 $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * (\text{real } p / \text{real } K))))$
 $= \text{of-int } (\sum j = 1..5. 2 * \text{int } (\text{hof } K - (j * p + 1) \text{ div } 2) - \max (2 * \text{int } (\text{hof } K) -$
 $\text{int } j * \text{int } p) \ 0)$

proof $-$

have *facts*: $7 \leq p \ p \leq K \ K < 3 * p$ **using** h **unfolding** *OuterHyp-def* **by** *auto*
have $K0$: $0 < \text{real } K$ **using** *facts* **by** *simp*
have hh : $\text{real } (\text{hof } K) = \text{lam-r} * \text{real } K$ **by** (*rule hof-real*[*OF* K])
have fl : $\text{floor } (\text{real } K / \text{real } p) = \text{int } (K \text{ div } p)$
by (*rule floor-divide-of-nat-eq*)
have trm : $\text{real } K * \text{pos-r } (2 * \text{lam-r} - \text{real } j * (\text{real } p / \text{real } K)) =$

$of_int \ (max \ (2 * int \ (hof \ K) - int \ j * int \ p) \ 0) \ \text{for } j$
proof –
have $t: real \ K * (2 * lam-r - real \ j * (real \ p / real \ K)) = 2 * real \ (hof \ K) -$
 $real \ j * real \ p$
using $K0 \ hh \ \text{by} \ (simp \ add: \ field-simps)$
show $?thesis \ unfolding \ mult-pos-r-nonneg[OF \ of-nat-0-le-iff] \ t \ \text{by} \ (simp \ add:$
 $of-int-max)$
qed
have $hh6: 2 * hof \ K < 6 * p \ \text{using} \ hof-le[of \ K] \ facts \ \text{by} \ linarith$
have $- \ of-int \ (vpS \ K \ p) = - \ (2 * real \ (hof \ K) * real \ (K \ div \ p)) + 2 * (\sum \ j =$
 $1..5. \ real \ (hof \ K - (j * p + 1) \ div \ 2))$
unfolding $vpS-outer[OF \ p \ K \ h] \ sum-floor-eq[OF \ p \ hh6] \ \text{by} \ simp$
moreover **have** $of-int \ (\sum \ j = 1..5. \ 2 * int \ (hof \ K - (j * p + 1) \ div \ 2) - max$
 $(2 *$
 $int \ (hof \ K) - int \ j * int \ p) \ 0)$
 $= 2 * (\sum \ j = 1..5. \ real \ (hof \ K - (j * p + 1) \ div \ 2)) - (\sum \ j =$
 $1..5. \ (of-int \ (max \ (2 * int \ (hof \ K) - int \ j * int \ p) \ 0) :: real))$
by $(simp \ add: \ of-int-sum \ sum-subtractf \ sum-distrib-left)$
moreover **have** $real \ K * (- \ 2 * lam-r * of-int \ (int \ (K \ div \ p))) =$
 $- \ (2 * real \ (hof \ K) * real \ (K \ div \ p))$
by $(simp \ add: \ hh \ algebra-simps)$
ultimately **show** $?thesis$
unfolding $fl \ distrib-left \ sum-distrib-left \ trm \ \text{by} \ linarith$
qed

lemma *half-ceil-abs*:

fixes $h \ m :: nat$

shows $|2 * int \ (h - (m + 1) \ div \ 2) - max \ (2 * int \ h - int \ m) \ 0| \leq 1$

unfolding $abs-if \ max-def \ \text{by} \ presburger$

lemma *two-sub-half-le*:

fixes $h \ m :: nat$

shows $2 * real \ (h - (m + 1) \ div \ 2) \leq max \ (2 * real \ h - real \ m) \ 0$

proof –

have $2 * int \ (h - (m + 1) \ div \ 2) \leq max \ (2 * int \ h - int \ m) \ 0 \ \text{unfolding}$
 $max-def \ \text{by} \ presburger$

hence $real-of-int \ (2 * int \ (h - (m + 1) \ div \ 2)) \leq real-of-int \ (max \ (2 * int \ h -$
 $int \ m) \ 0)$

by $(simp \ only: \ of-int-le-iff)$

thus $?thesis \ \text{by} \ (simp \ add: \ of-int-max)$

qed

lemma *vpS-sub-limit-abs*:

assumes $p: \text{prime } p \ \text{and} \ K: 40 \ dvd \ K \ \text{and} \ h: OuterHyp \ K \ p$

shows $|- \ of-int \ (vpS \ K \ p) - real \ K * (- \ 2 * lam-r * of-int \ (floor \ (real \ K / real$
 $p)) +$

$(\sum \ j = 1..5. \ pos-r \ (2 * lam-r - real \ j * (real \ p / real \ K))))| \leq 5$

proof –

have $one: |2 * int \ (hof \ K - (j * p + 1) \ div \ 2) - max \ (2 * int \ (hof \ K) - int \ j$

```

*
  int p) 0| ≤ 1 for j
  using half-ceil-abs[of hof K j * p] by simp
  have s: |∑ j = 1..5. 2 * int (hof K - (j * p + 1) div 2) - max (2 * int (hof
K) -
    int j * int p) 0| ≤ 5
  proof -
    have |∑ j = 1..5. 2 * int (hof K - (j * p + 1) div 2) - max (2 * int (hof
K) -
      int j * int p) 0|
      ≤ (∑ j = 1..5. |2 * int (hof K - (j * p + 1) div 2) - max (2 * int (hof
K) -
        int j * int p) 0|)
    by (rule sum-abs)
    also have ... ≤ (∑ j ∈ {1..5::nat}. (1::int)) by (rule sum-mono) (rule one)
    also have ... = (5::int) by simp
    finally show ?thesis .
  qed
  show ?thesis unfolding vpS-sub-limit[OF p K h] of-int-abs[symmetric] of-int-le-numeral-iff
    by (rule s)
  qed

```

The outer analogue of (5.7) (constant 7).

theorem *uniformity-outer*:

```

  ∃ Cst. ∀ K p. prime p ⟶ 40 dvd K ⟶
    OuterHyp K p ⟶
      | - of-int (gammaOut K p) - real K * (R0-r (real p / real K) - dfun-r (real
p /
      real K))| ≤ Cst ∧
      | - of-int (vpS K p) - real K * (- 2 * lam-r * of-int (floor (real K / real
p))
      +
        (∑ j = 1..5. pos-r (2 * lam-r - real j * (real p / real K))))| ≤ Cst
  proof (intro exI[of - 7] allI impI conjI)
    fix K p assume p: prime p and K: 40 dvd K and h: OuterHyp K p
    have | - of-int (gammaOut K p) - of-int (Eout K p)| ≤ (7 :: real)
      using gammaOut-sub-Eout[OF p K h]
      by (simp flip: of-int-minus of-int-diff of-int-abs)
    thus | - of-int (gammaOut K p) - real K * (R0-r (real p / real K) - dfun-r
(real p /
      real K))| ≤ 7
      unfolding K-mul-R0-sub-dfun[OF p K h] .
    show | - of-int (vpS K p) - real K * (- 2 * lam-r * of-int (floor (real K / real
p)) +
      (∑ j = 1..5. pos-r (2 * lam-r - real j * (real p / real K))))| ≤ 7
      using vpS-sub-limit-abs[OF p K h] by linarith
  qed

```

27.6 Exact integrals and tails

$$\int_{-1/3}^{1/2} d(y) dy = 9/640.$$

theorem *integral-dfun: interval-lebesgue-integral lborel (ereal (1 / 3)) (ereal (1 / 2)) dfun-r =*
9 / 640
by (rule *integral-dfun-eq*)

(5.12)-(5.14): $R(x) = x F(x) + Q(x)$ with $-1/2 \leq Q(x) \leq 13/8$ (for all $x > 0$).

theorem *R-decomp-bound:*
assumes $3 \leq x$
shows $-1/2 \leq R-r x - x * (4 * lam-r + 2 * lam-r * frac x - 12 * lam-r * frac (alpha-r * x)) \wedge$
 $R-r x - x * (4 * lam-r + 2 * lam-r * frac x - 12 * lam-r * frac (alpha-r * x)) \leq$
 $13 / 8$
by (rule *R-decomp-bound-all*)

The mean of P over a period is $2923/240$.

theorem *Pper-mean: interval-lebesgue-integral lborel (ereal 0) (ereal 40) Pper / 40 = 2923 / 240*
by (rule *Pper-mean-eq*)

$R(x)/x^3$ is integrable on $[3, \infty)$.

theorem *R-integrableOn: set-integrable lborel {3..} (λx. R-r x / x ^ 3)*
by (rule *R-div-cube-integrable*)

27.7 Prime sums via the prime number theorem

The prime number theorem in Chebyshev's form, from the AFP (*Prime-Number-Theorem.v-asymptotic* $\vartheta \sim x$).

lemma *theta-div-tendsto: ((λx. primes-theta x / x) → 1) at-top*
by (rule *pnt-theta*)

Chebyshev's bound $\vartheta(n) \leq n \log 4$ (from `Prime_Distribution_Elementary`).

lemma *sum-log-primes-le: (∑ p ∈ {p. p ≤ n ∧ prime p}. ln (real p)) ≤ ln 4 * real n*

proof –

have *eq: (∑ p ∈ {p. p ≤ n ∧ prime p}. ln (real p)) = primes-theta (real n)*

unfolding *primes-theta-def prime-sum-upto-def* **by** (rule *sum.cong*) *auto*

show *?thesis*

proof (*cases n = 0*)

case *True*

thus *?thesis* **unfolding** *eq* **by** (*simp add: prime-sum-upto-eq-0 primes-theta-def*)

next

case *False*
hence *primes-theta* (*real n*) < *ln 4 * real n* **by** (*intro* *∅-upper-bound*) *simp*
thus *?thesis unfolding eq by simp*
qed
qed

Outer range, variable $y = p/K$: Riemann-sum upper bound for a bounded, almost-everywhere continuous φ (step majorants + the PNT cell estimates).

theorem *outer-prime-sum-limsup*:
assumes $0 < a \leq b \wedge y. y \in \{a..b\} \implies$
 $|\varphi y| \leq Bd$ *countable D*
and $\bigwedge y. y \in \{a < .. < b\} \implies y \notin D \implies$
 $isCont \varphi y 0 < \varepsilon$
shows $\forall_F K$ *in at-top*.
 $(\sum p \in \{p. p \leq nat (floor (b * real K)) \wedge prime p \wedge$
 $a * real K < real p \wedge real p \leq b * real K\}.$
 $\varphi (real p / real K) * ln (real p))$
 $\leq (interval-lebesgue-integral lborel (ereal a) (ereal b) \varphi + \varepsilon) * real K$
using *assms by (rule outer-prime-sum-bound)*

Inner range, variable $x = K/p$.

theorem *inner-prime-sum-limsup*:
assumes $\exists < M \wedge x. x \in \{\exists..real M\} \implies$
 $|f x| \leq Bd$ *countable D*
and $\bigwedge x. x \in \{\exists < .. < real M\} \implies x \notin D \implies$
 $isCont f x 0 < \varepsilon$
shows $\forall_F K$ *in at-top*.
 $(\sum p \in \{p. p \leq K \wedge prime p \wedge K < p * M \wedge$
 $\exists * p \leq K\}. real p * f (real K / real p) * ln (real p))$
 $\leq (interval-lebesgue-integral lborel (ereal \exists) (ereal (real M)) (\lambda x. f x /$
 $x \wedge \exists) + \varepsilon) * real K \wedge 2$
using *assms by (rule inner-prime-sum-bound)*

27.8 Continuity of the limiting functions off a countable set

theorem *R-continuousAt-off*: $\exists D. countable D \wedge (\forall x. 0 < x \implies$
 $x \notin D \implies isCont R-r x)$
by (*rule R-isCont-off-countable*)

theorem *outerIntegrand-continuousAt-off*: $\exists D. countable D \wedge$
 $(\forall y. 0 < y \implies y \notin D \implies isCont outer-integrand y)$
by (*rule outer-integrand-isCont-off-countable*)

theorem *outerIntegrand-bounded*: $\exists Bd. \forall y \in$
 $\{1 / \exists..2 * lam-r\}. |outer-integrand y| \leq Bd$
by (*rule outer-integrand-bounded*)

27.9 Proposition 5.2 and (5.21)

lemma *of-rat-power-int*: *of-rat* (*x powi k*) = (*of-rat x :: 'a :: real-field*) *powi k*
unfolding *power-int-def by (auto simp: of-rat-power of-rat-inverse)*

lemma *ln-power-int*: $0 < x \implies \ln (x \text{ powi } k) = \text{of-int } k * \ln (x :: \text{real})$
unfolding *power-int-def* **by** (*fastforce simp: ln-realpow ln-inverse*)

lemma *log-mKM*:

$\ln (\text{of-rat } (mKM \ K \ M)) = (\sum p \in \{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\}. - \text{of-int } (Lp \ K \ M \ p) * \ln (\text{real } p))$

proof –

have *fin*: $\text{finite } \{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\}$ **by** *simp*

have $\ln (\text{of-rat } (mKM \ K \ M)) = \ln (\prod p \in \{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\}. (\text{real } p) \text{ powi } (- Lp \ K \ M \ p))$

unfolding *mKM-def* **by** (*simp add: of-rat-prod of-rat-power-int*)

also have $\dots = (\sum p \in \{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\}. \ln ((\text{real } p) \text{ powi } (- Lp \ K \ M \ p)))$

by (*intro ln-prod fin*) (*auto simp: power-int-def prime-gt-0-nat*)

also have $\dots = (\sum p \in \{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\}. - \text{of-int } (Lp \ K \ M \ p) * \ln (\text{real } p))$

by (*intro sum.cong refl*) (*simp add: ln-power-int prime-gt-0-nat*)

finally show *?thesis* .

qed

$|R(x)| \leq 18M + 2$ on $[3, M]$, from (5.12)-(5.14).

theorem *R-bounded*: $\exists Bd. \forall x \in \{3..real \ M\}. |R-r \ x| \leq Bd$

proof (*intro exI[of - 18 * real M + 2] ballI*)

fix *x* **assume** *x*: $x \in \{3..real \ M\}$

have *Q*: $-1 / 2 \leq R-r \ x - x * (4 * lam-r + 2 * lam-r * \text{frac } x - 12 * lam-r * \text{frac } (alpha-r * x))$

$R-r \ x - x * (4 * lam-r + 2 * lam-r * \text{frac } x - 12 * lam-r * \text{frac } (alpha-r * x)) \leq$

$13 / 8$

using *R-decomp-bound[of x] x* **by** *auto*

have *f*: $0 \leq \text{frac } x \text{ frac } x < 1 \ 0 \leq \text{frac } (alpha-r * x) \text{ frac } (alpha-r * x) < 1$

by (*auto simp: frac-lt-1*)

have *F*: $|x * (4 * lam-r + 2 * lam-r * \text{frac } x - 12 * lam-r * \text{frac } (alpha-r * x))| \leq 8 * x$

proof –

have *b*: $|4 * lam-r + 2 * lam-r * \text{frac } x - 12 * lam-r * \text{frac } (alpha-r * x)| \leq 8$

unfolding *abs-le-iff lam-r-def* **using** *f* **by** (*intro conjI; linarith*)

have *x0*: $0 \leq x$ **using** *x* **by** *auto*

have $x * |4 * lam-r + 2 * lam-r * \text{frac } x - 12 * lam-r * \text{frac } (alpha-r * x)| \leq x * 8$

by (*rule mult-left-mono[OF b x0]*)

thus *?thesis* **using** *x0* **by** (*simp add: abs-mult mult.commute*)

qed

have *F1*: $x * (4 * lam-r + 2 * lam-r * \text{frac } x - 12 * lam-r * \text{frac } (alpha-r * x)) \leq 8 * x$

$- (x * (4 * lam-r + 2 * lam-r * \text{frac } x - 12 * lam-r * \text{frac } (alpha-r * x))$

```

x))) ≤ 8 * x
  using F unfolding abs-le-iff by auto
  have xm: x ≤ real M using x by auto
  show |R-r x| ≤ 18 * real M + 2 unfolding abs-le-iff using Q F1 xm
    by (intro conjI; linarith)
qed

```

For $K < p \leq 2h$: $-v\text{-}p(S\text{-}K) \leq K \cdot \text{outerIntegrand}(p/K)$ (Legendre again, now $K < p$).

lemma *vpS-large*:

```

  assumes p: prime p and K: 40 dvd K 40 ≤ K and Kp: K < p and p2h: p ≤ 2
    * hof K
  shows - of-int (vpS K p) ≤ real K * outer-integrand (real p / real K)
proof -
  have K0: 0 < K using K by linarith
  define B where B = 2 * hof K + K + 1
  have B1: 1 ≤ B unfolding B-def by simp
  have m4: multiplicity p (4::nat) = 0
proof -
  have ¬ p dvd 4
proof
  assume p dvd 4
  hence p ≤ 4 by (rule dvd-imp-le) simp
  thus False using K Kp by linarith
qed
  thus ?thesis by (rule not-dvd-imp-multiplicity-0)
qed
have leg: vpS K p = - 2 * int (∑ i = 1..hof K - 1. 2 * i div p)
proof -
  have pp: 2 * p ≤ p ^ 2 using mult-le-mono1[OF prime-ge-2-nat[OF p], of p]
    by (simp add: power2-eq-square)
  have hK: hof K ≤ K and NK: Nof K ≤ K by (rule hof-le, rule Nof-le)
  have L: vpS K p = 2 * int (hof K) * (∑ a = 1..B. int (K div p ^ a))
    - 12 * int (hof K) * (∑ a = 1..B. int (Nof K div p ^ a))
    - 2 * (∑ i = 1..hof K - 1. ∑ a = 1..B. int (2 * i div p ^ a))
    + (int (hof K) - 1) * int (multiplicity p (4::nat))
    unfolding B-def by (rule vpS-legendre[OF p K(1) K0])
  have Kp2: K < p ^ 2 using pp Kp by linarith
  have Np2: Nof K < p ^ 2 using pp Kp NK by linarith
  have sK: (∑ a = 1..B. K div p ^ a) = 0
    using sum-div-pow-single[OF p B1 Kp2] Kp by simp
  have sN: (∑ a = 1..B. Nof K div p ^ a) = 0
    using sum-div-pow-single[OF p B1 Np2] Kp NK by simp
  have si: (∑ a = 1..B. 2 * i div p ^ a) = 2 * i div p if i ∈ {1..hof K - 1}
for i
proof -
  have i: 1 ≤ i ≤ hof K - 1 using that by auto
  have 2 * i < p ^ 2 using pp Kp hK i by arith
  thus ?thesis by (rule sum-div-pow-single[OF p B1])

```

qed
have $s1$: $(\sum a = 1..B. \text{int } (K \text{ div } p \wedge a)) = 0$ **unfolding** *of-nat-sum[symmetric]*
sK **by** *simp*
have $s2$: $(\sum a = 1..B. \text{int } (Nof K \text{ div } p \wedge a)) = 0$ **unfolding** *of-nat-sum[symmetric]*
sN
by *simp*
have $s3$: $(\sum i = 1..hof K - 1. \sum a = 1..B. \text{int } (2 * i \text{ div } p \wedge a)) = \text{int } (\sum$
i =
 $1..hof K - 1. 2 * i \text{ div } p)$
proof –
have $(\sum i = 1..hof K - 1. \sum a = 1..B. \text{int } (2 * i \text{ div } p \wedge a)) = (\sum i =$
 $1..hof K - 1. \text{int } (2 * i \text{ div } p))$
by (*rule sum.cong[OF refl]*) (*simp only: of-nat-sum[symmetric] si*)
thus *?thesis* **by** *simp*
qed
show *?thesis* **unfolding** $L s1 s2 s3 m4$ **by** *simp*
qed
have $hh6$: $2 * hof K < 6 * p$ **using** *hof-le[of K] Kp* **by** *linarith*
have hv : $-\text{of-int } (vpS K p) = (\sum j = 1..5. 2 * \text{real } (hof K - (j * p + 1) \text{ div } 2))$
unfolding *leg sum-floor-eq[OF p hh6]* **by** (*simp add: sum-distrib-left sum-negf*)
have Kr : $0 < \text{real } K$ **using** $K0$ **by** *simp*
define y **where** $y = \text{real } p / \text{real } K$
have $y1$: $1 < y$ **unfolding** *y-def* **using** $Kp Kr$ **by** (*simp add: less-divide-eq*)
have $R0$: $R0\text{-r } y = 0$ **and** d : $dfun\text{-r } y = 0$ **and** fl : $\text{floor } (1 / y) = 0$
proof –
have n : $\neg y < 1 / 2 \neg y < 1$ **using** $y1$ **by** *linarith+*
show $R0\text{-r } y = 0$ **unfolding** *R0-r-def* **using** n **by** *simp*
show $dfun\text{-r } y = 0$ **unfolding** *dfun-r-def* **using** n **by** *simp*
have $0 \leq 1 / y$ $1 / y < 1$ **using** $y1$ **by** (*simp-all add: field-simps*)
thus $\text{floor } (1 / y) = 0$ **by** (*simp add: floor-eq-iff*)
qed
have oi : $\text{real } K * \text{outer-integrand } y = (\sum j = 1..5. \text{real } K * \text{pos-r } (2 * \text{lam-r}$
 $-\text{real } j * y))$
unfolding *outer-integrand-def R0 d fl* **by** (*simp add: sum-distrib-left*)
have hh : $\text{real } (hof K) = \text{lam-r} * \text{real } K$ **by** (*rule hof-real[OF K(1)]*)
have trm : $2 * \text{real } (hof K - (j * p + 1) \text{ div } 2) \leq \text{real } K * \text{pos-r } (2 * \text{lam-r} -$
 $\text{real } j * y)$
for j
proof –
have e : $\text{real } K * \text{pos-r } (2 * \text{lam-r} - \text{real } j * y) = \text{max } (2 * \text{real } (hof K) -$
 $\text{real } j * \text{real } p) 0$
proof –
have t : $\text{real } K * (2 * \text{lam-r} - \text{real } j * y) = 2 * \text{real } (hof K) - \text{real } j * \text{real } p$
unfolding *y-def* **using** $Kr hh$ **by** (*simp add: field-simps*)
show *?thesis* **unfolding** *mult-pos-r-nonneg[OF of-nat-0-le-iff]* t **..**
qed
show *?thesis* **unfolding** e
using *two-sub-half-le[of hof K j * p]* **by** *simp*

qed
 show ?thesis unfolding hv y-def[symmetric] oi by (rule sum-mono) (rule trm)
 qed

The main term of $-L\text{-}p(K, M)$.

definition $\text{mainCoef} :: \text{nat} \Rightarrow \text{nat} \Rightarrow \text{nat} \Rightarrow \text{real}$ **where**
 $\text{mainCoef } K \ M \ p =$
 (if $p * M \leq K$ then $\text{real } (6 * \text{hof } K * \text{nat-log } p \ (5 * K) + \text{hof } K * \text{multiplicity } p \ (24::\text{nat}))$
 else if $3 * p \leq K$ then $\text{real } p * R\text{-}r \ (\text{real } K / \text{real } p)$
 else $\text{real } K * \text{outer-integrand } (\text{real } p / \text{real } K)$)

Termwise: $-L\text{-}p \leq \text{mainCoef} + (2 \ C\text{-}1 + 1 + 2 \ C\text{-}2)$ for every prime $p \leq 2h$.

lemma neg-Lp-le :

assumes $M: 40 \leq M$ **and** $C1: 0 \leq C1$ **and** $C2: 0 \leq C2$
and $\text{hin}: \bigwedge K \ p. \text{prime } p \implies \text{InnerHyp } K \ M \ p \implies$
 $|\text{of-rat } (\text{gammaIn } K \ M \ p) - \text{real } p * \text{Gam-r } (\text{real } K / \text{real } p)| \leq C1 \wedge$
 $|\text{of-int } (\text{vpS } K \ p) - \text{real } p * \text{Nfun-r } (\text{real } K / \text{real } p)| \leq C1$
and $\text{hout}: \bigwedge K \ p. \text{prime } p \implies 40 \text{ dvd } K \implies$
 $\text{OuterHyp } K \ p \implies$
 $|\text{of-int } (\text{gammaOut } K \ p) - \text{real } K * (R0\text{-}r \ (\text{real } p / \text{real } K) - \text{dfun-r } (\text{real } p /$
 $\text{real } K))| \leq C2 \wedge$
 $|\text{of-int } (\text{vpS } K \ p) - \text{real } K * (-2 * \text{lam-r} * \text{of-int } (\text{floor } (\text{real } K / \text{real } p)) +$
 $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * (\text{real } p / \text{real } K))))| \leq C2$
and $K: 40 \text{ dvd } K \ 200 * M \wedge 2 \leq K \ 0 < K$ **and** $p: \text{prime } p \ p \leq 2 * \text{hof } K$
shows $-\text{of-int } (Lp \ K \ M \ p) \leq \text{mainCoef } K \ M \ p + (2 * C1 + 1 + 2 * C2)$

proof -

have $K40: 40 \leq K$

proof -

have $M \leq M \wedge 2$ **using** M **by** (intro self-le-power) auto

thus ?thesis **using** $M \ K(2)$ **by** linarith

qed

show ?thesis

proof (cases $p * M \leq K$)

case True

have $-\text{of-int } (Lp \ K \ M \ p) = \text{mainCoef } K \ M \ p$ **unfolding** $Lp\text{-def}$ mainCoef-def
 if-P[OF True] **by** simp

thus ?thesis **using** $C1 \ C2$ **by** linarith

next

case small: False

show ?thesis

proof (cases $3 * p \leq K$)

case True

have $IH: \text{InnerHyp } K \ M \ p$ **unfolding** InnerHyp-def **using** $M \ K \ \text{small} \ \text{True}$
by auto

have $h1: |\text{of-rat } (\text{gammaIn } K \ M \ p) - \text{real } p * \text{Gam-r } (\text{real } K / \text{real } p)| \leq C1$

```

    and h2: |of-int (vpS K p) - real p * Nfun-r (real K / real p)| ≤ C1
    using hin[OF p(1) IH] by auto
    have fl: of-rat (gammaIn K M p) - 1 ≤ (of-int (floor (gammaIn K M p)) ::
real)
    proof -
      have gammaIn K M p < of-int ⌊gammaIn K M p⌋ + 1
      using floor-correct[of gammaIn K M p] by simp
      hence gammaIn K M p - 1 ≤ of-int ⌊gammaIn K M p⌋ by linarith
      hence of-rat (gammaIn K M p - 1) ≤
        (of-rat (of-int ⌊gammaIn K M p⌋) :: real)
      by (simp only: of-rat-less-eq)
      thus ?thesis by (simp add: of-rat-diff)
    qed
    have e: mainCoef K M p = - (real p * Gam-r (real K / real p)) - real p *
Nfun-r (real K /
real p)
    unfolding mainCoef-def R-r-def using small True by (simp add: alge-
bra-simps)
    have l: - of-int (Lp K M p) = - of-int (vpS K p) - (of-int (floor (gammaIn
K M p)) :: real)
    unfolding Lp-def using small True by simp
    show ?thesis
    using e l h1[unfolded abs-le-iff] h2[unfolded abs-le-iff] fl C2 by linarith
  next
    case inner: False
    show ?thesis
    proof (cases p ≤ K)
      case True
      have OH: OuterHyp K p
      unfolding OuterHyp-def
      proof -
        have Kp3: K < 3 * p using inner by simp
        have p7: 7 ≤ p using Kp3 K40 by linarith
        have K * K < (3 * p) * (3 * p) using Kp3 by (intro mult-strict-mono)
      auto
      moreover have 18 * K ≤ K * K using K40 by (intro mult-le-mono1)
    linarith
      moreover have (3 * p) * (3 * p) = 9 * p ^ 2 by (simp add:
power2-eq-square)
      ultimately have sq: 2 * K < p ^ 2 by linarith
      have n1: 2 * (3 * K div 40) < p using Kp3 by linarith
      have n2: 5 * (3 * K div 40) ≤ 2 * p - 2 using Kp3 p7 by linarith
      show 7 ≤ p ∧ p ≤ K ∧ K < 3 * p ∧ 2 * K < p ^ 2 ∧
        2 * Nof K < p ∧ 5 * Nof K ≤ 2 * p - 2
      unfolding Nof-def using p7 True Kp3 sq n1 n2 by blast
    qed
    have h1: |- of-int (gammaOut K p) - real K * (R0-r (real p / real K) -
dfun-r (real p / real K))| ≤ C2
    and h2: |- of-int (vpS K p) - real K * (- 2 * lam-r * of-int (floor (real

```

$K /$
 $\text{real } p)) +$
 $(\sum_{C2} j = 1..5. \text{ pos-r } (2 * \text{ lam-r } - \text{ real } j * (\text{real } p / \text{real } K)))) \leq$
using *hout*[*OF* *p*(1) *K*(1) *OH*] **by** *auto*
have *ft*: *floor* (1 / (real *p* / real *K*)) = *floor* (real *K* / real *p*)
by *simp*
have *e*: *mainCoef* *K* *M* *p* = real *K* * (*R0-r* (real *p* / real *K*) - *dfun-r* (real
p / real *K*))
 $+ \text{real } K * (- 2 * \text{ lam-r } * \text{ of-int } (\text{floor } (\text{real } K / \text{real } p)) + (\sum j =$
 $1..5. \text{ pos-r } (2 * \text{ lam-r } - \text{ real } j * (\text{real } p / \text{real } K))))$
unfolding *mainCoef-def* *outer-integrand-def* *fl* **using** *small inner*
by (*simp add: algebra-simps*)
have *l*: - *of-int* (*Lp* *K* *M* *p*) = - *of-int* (*vpS* *K* *p*) - (*of-int* (*gammaOut* *K*
p) :: real)
unfolding *Lp-def* **using** *small inner True* **by** *simp*
show ?thesis
using *e l h1*[*unfolded abs-le-iff*] *h2*[*unfolded abs-le-iff*] *C1* **by** *linarith*
next
case *False*
have - *of-int* (*vpS* *K* *p*) ≤ real *K* * *outer-integrand* (real *p* / real *K*)
by (*rule vpS-large*[*OF* *p*(1) *K*(1) *K40 - p*(2)]) (*use False in simp*)
moreover **have** - (*of-int* (*Lp* *K* *M* *p*) :: real) = - *of-int* (*vpS* *K* *p*)
unfolding *Lp-def*
using *small inner False* **by** *simp*
moreover **have** *mainCoef* *K* *M* *p* = real *K* * *outer-integrand* (real *p* / real
K)
unfolding *mainCoef-def* **using** *small inner* **by** *simp*
ultimately show ?thesis **using** *C1 C2* **by** *linarith*
qed
qed
qed
qed

lemma *sum-log-P-le*:

$(\sum p \in \{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\}. \ln (\text{real } p)) \leq 2 * \text{real } K * \ln 4$

proof -

have $(\sum p \in \{p. p \leq 2 * \text{hof } K \wedge$
 $\text{prime } p\}. \ln (\text{real } p)) \leq \ln 4 * \text{real } (2 * \text{hof } K)$
by (*rule sum-log-primes-le*)
also have $\dots \leq \ln 4 * (2 * \text{real } K)$
using *hof-le*[*of* *K*] **by** (*intro mult-left-mono ln-ge-zero*) *simp-all*
finally show ?thesis **by** (*simp add: mult.commute*)
qed

The primes $p \leq 2h$ split into the three ranges of (5.1) ($40 \mid K$, $K > 0$, $M \geq 3$), and *mainCoef* takes the corresponding branch on each.

lemma *three-le-mul*: $3 \leq M \implies 3 * q \leq q * (M::\text{nat})$

by (*metis mult.commute mult-le-mono1*)

```

lemma not-mul-le-of-lt:  $3 \leq M \implies K < 3 * q \implies$ 
   $\neg q * M \leq (K::nat)$ 
using three-le-mul[of  $M\ q$ ] by linarith

lemma le-of-three-mul:  $3 * q \leq K \implies q \leq (K::nat)$ 
by linarith

lemma primes-split:
  assumes  $K: 40 \text{ dvd } K \ 0 < K$  and  $M: 3 \leq M$ 
  shows  $\{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\} =$ 
     $\{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\} \cup \{p. p \leq K \wedge$ 
       $\text{prime } p \wedge K < p * M \wedge 3 * p \leq K\}$ 
     $\cup \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$ 
       $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}$ 
  and  $\{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\} \cap \{p. p \leq K \wedge$ 
     $\text{prime } p \wedge K < p * M \wedge 3 * p \leq K\} = \{\}$ 
  and  $(\{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\} \cup \{p. p \leq K \wedge$ 
     $\text{prime } p \wedge K < p * M \wedge 3 * p \leq K\})$ 
     $\cap \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$ 
       $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\} = \{\}$ 
  and  $\bigwedge p. p \in \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\} \implies$ 
     $\text{mainCoef } K\ M\ p = \text{real } (6 * \text{hof } K * \text{nat-log } p\ (5 * K) + \text{hof } K * \text{multiplicity } p\ (24::nat))$ 
  and  $\bigwedge p. p \in \{p. p \leq K \wedge \text{prime } p \wedge K < p * M \wedge$ 
     $3 * p \leq K\} \implies \text{mainCoef } K\ M\ p = \text{real } p * \text{R-r } (\text{real } K / \text{real } p)$ 
  and  $\bigwedge p. p \in \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$ 
     $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\} \implies$ 
     $\text{mainCoef } K\ M\ p = \text{real } K * \text{outer-integrand } (\text{real } p / \text{real } K)$ 
proof -
  have fl:  $\text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) = 2 * \text{hof } K$ 
proof -
  have e:  $2 * \text{lam-r} * \text{real } K = \text{real } (2 * \text{hof } K)$  using hof-real[OF  $K(1)$ ]
  by (simp add: mult.assoc)
  show ?thesis unfolding e by (simp only: floor-of-nat nat-int)
qed
have le:  $\text{real } p \leq 2 * \text{lam-r} * \text{real } K \iff p \leq 2 * \text{hof } K$  for  $p$ 
proof -
  have e:  $2 * \text{lam-r} * \text{real } K = \text{real } (2 * \text{hof } K)$  using hof-real[OF  $K(1)$ ]
  by (simp add: mult.assoc)
  show ?thesis unfolding e of-nat-le-iff ..
qed
have lt:  $1 / 3 * \text{real } K < \text{real } p \iff K < 3 * p$  for  $p$ 
proof
  assume  $1 / 3 * \text{real } K < \text{real } p$ 
  hence  $\text{real } K < \text{real } (3 * p)$  by (simp only: of-nat-mult of-nat-numeral)
  thus  $K < 3 * p$  by (simp only: of-nat-less-iff)
next
  assume  $K < 3 * p$ 

```


hence $\text{real } K < \text{real } (3 * p)$ **by** (*simp only: of-nat-less-iff*)
 hence $\text{real } K < 3 * \text{real } p$ **by** (*simp only: of-nat-mult of-nat-numeral*)
 thus $1 / 3 * \text{real } K < \text{real } p$ **by** *linarith*
qed
 have $hK: K \leq 2 * \text{hof } K$
 using $K(1)$ **unfolding** *hof-def* **by** *fastforce*
 show $\{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\} =$
 $\{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\} \cup \{p. p \leq K \wedge$
 $\text{prime } p \wedge K < p * M \wedge 3 * p \leq K\}$
 $\cup \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$
 $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}$
unfolding *fl le lt* **using** $hK M$
by (*auto simp: not-le dest: le-of-three-mul order-trans[OF three-le-mul[OF M]]*)
intro:
 $\text{order-trans}[OF - hK])$
 show $\{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\} \cap \{p. p \leq K \wedge$
 $\text{prime } p \wedge K < p * M \wedge 3 * p \leq K\} = \{\}$
by *auto*
 show $(\{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\} \cup \{p. p \leq K \wedge$
 $\text{prime } p \wedge K < p * M \wedge 3 * p \leq K\})$
 $\cap \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$
 $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\} = \{\}$
unfolding *fl le lt* **using** M
by (*auto simp: not-mul-le-of-lt[OF M] dest: leD*)
 show $\text{mainCoef } K M p = \text{real } (6 * \text{hof } K * \text{nat-log } p (5 * K) + \text{hof } K *$
multiplicity $p (24::\text{nat}))$
if $p \in \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\}$ **for** p
 using *that* **unfolding** *mainCoef-def* **by** *simp*
 show $\text{mainCoef } K M p = \text{real } p * R\text{-r } (\text{real } K / \text{real } p)$
if $p \in \{p. p \leq K \wedge \text{prime } p \wedge K < p * M \wedge 3 * p \leq K\}$ **for** p
 using *that* **unfolding** *mainCoef-def* **by** *simp*
 show $\text{mainCoef } K M p = \text{real } K * \text{outer-integrand } (\text{real } p / \text{real } K)$
if $p \in \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$
 $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}$ **for** p
 using *that* M **unfolding** *mainCoef-def* *lt*
 using *not-mul-le-of-lt*[$OF M, \text{of } K p$] **by** (*auto dest: leD*)
qed

lemma *filterlim-40*: *filterlim* $(\lambda n :: \text{nat}. 40 * n)$ *at-top* *at-top*
by (*rule filterlim-subseq*) (*simp add: strict-mono-Suc-iff*)

Proposition 5.2 (with $K = 40n$): $\limsup K^{-\wedge 2} \log m\{-K, M\} \leq I\text{-out} +$
 $6\lambda/M + \int -3^{\wedge M} R(x)/x^{\wedge 3} dx.$

theorem *log-mKM-limsup*:
 assumes $M: 40 \leq M$ **and** $e: 0 < \varepsilon$
 shows $\forall_F n$ *in at-top*.
 $\ln (\text{of-rat } (mKM (40 * n) M)) / \text{real } (40 * n)^{\wedge 2} \leq$
 $\text{of-rat } I\text{out} + 6 * \text{lam-r} / \text{real } M$
 $+ \text{interval-lebesgue-integral lborel } (\text{ereal } 3) (\text{ereal } (\text{real } M)) (\lambda x. R\text{-r } x / x^{\wedge 3})$

$\beta) +$
 ε
proof –
define IR **where** $IR = \text{interval-lebesgue-integral lborel } (\text{ereal } \beta) (\text{ereal } (\text{real } M))$
 $(\lambda x.$
 $R\text{-r } x / x \wedge \beta)$
obtain $C1'$ **where** $C1': \bigwedge K p. \text{prime } p \implies \text{InnerHyp } K M p \implies$
 $| \text{of-rat } (\text{gammaIn } K M p) - \text{real } p * \text{Gam-r } (\text{real } K / \text{real } p) | \leq C1' \wedge$
 $| \text{of-int } (\text{vpS } K p) - \text{real } p * \text{Nfun-r } (\text{real } K / \text{real } p) | \leq C1'$
using $\text{gammaIn-uniform-estimate}[OF M]$ **by** blast
obtain $C2'$ **where** $C2': \bigwedge K p. \text{prime } p \implies 40 \text{ dvd } K \implies$
 $\text{OuterHyp } K p \implies$
 $| - \text{of-int } (\text{gammaOut } K p) - \text{real } K * (\text{R0-r } (\text{real } p / \text{real } K) - \text{dfun-r } (\text{real } p /$
 $\text{real } K)) | \leq C2' \wedge$
 $| - \text{of-int } (\text{vpS } K p) - \text{real } K * (- 2 * \text{lam-r} * \text{of-int } (\text{floor } (\text{real } K / \text{real } p)) +$
 $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * (\text{real } p / \text{real } K)))) | \leq C2'$
using uniformity-outer **by** blast
define $C1$ **where** $C1 = \max C1' 0$
define $C2$ **where** $C2 = \max C2' 0$
have $C1: 0 \leq C1$ **and** $C2: 0 \leq C2$ **unfolding** $C1\text{-def } C2\text{-def}$ **by** simp-all
have $\text{hin}: \bigwedge K p. \text{prime } p \implies \text{InnerHyp } K M p \implies$
 $| \text{of-rat } (\text{gammaIn } K M p) - \text{real } p * \text{Gam-r } (\text{real } K / \text{real } p) | \leq C1 \wedge$
 $| \text{of-int } (\text{vpS } K p) - \text{real } p * \text{Nfun-r } (\text{real } K / \text{real } p) | \leq C1$
unfolding $C1\text{-def}$ **using** $C1'$ **by** $(\text{meson max.coboundedI1})$
have $\text{hout}: \bigwedge K p. \text{prime } p \implies 40 \text{ dvd } K \implies$
 $\text{OuterHyp } K p \implies$
 $| - \text{of-int } (\text{gammaOut } K p) - \text{real } K * (\text{R0-r } (\text{real } p / \text{real } K) - \text{dfun-r } (\text{real } p /$
 $\text{real } K)) | \leq C2 \wedge$
 $| - \text{of-int } (\text{vpS } K p) - \text{real } K * (- 2 * \text{lam-r} * \text{of-int } (\text{floor } (\text{real } K / \text{real } p)) +$
 $(\sum j = 1..5. \text{pos-r } (2 * \text{lam-r} - \text{real } j * (\text{real } p / \text{real } K)))) | \leq C2$
unfolding $C2\text{-def}$ **using** $C2'$ **by** $(\text{meson max.coboundedI1})$
define C **where** $C = 2 * C1 + 1 + 2 * C2$
have $C0: 0 \leq C$ **unfolding** $C\text{-def}$ **using** $C1 C2$ **by** linarith
have $e4: 0 < \varepsilon / 4$ **using** e **by** simp
have $M3': 3 \leq M$ **and** $M0: 0 < M$ **using** M **by** simp-all
have $E1: \forall_F K \text{ in at-top.}$
 $(\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\}.$
 $\text{real } (6 * \text{hof } K * \text{nat-log } p (5 * K) + \text{hof } K * \text{multiplicity } p (24::\text{nat})) *$
 $\ln (\text{real } p))$
 $\leq (6 * \text{lam-r} / \text{real } M + \varepsilon / 4) * \text{real } K \wedge 2$
by $(\text{rule small-prime-sum-bound}[OF M0 e4])$
obtain BR **where** $BR: \bigwedge x. x \in \{3.. \text{real } M\} \implies |R\text{-r } x| \leq BR$
using $R\text{-bounded}$ **by** blast
obtain DR **where** $DR: \text{countable } DR \bigwedge x. 0 < x \implies$
 $x \notin DR \implies \text{isCont } R\text{-r } x$

```

using R-continuousAt-off by blast
have E2:  $\forall_F K$  in at-top.
   $(\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge K < p * M \wedge$ 
     $3 * p \leq K\}. \text{real } p * R\text{-r } (\text{real } K / \text{real } p) * \ln (\text{real } p))$ 
   $\leq (IR + \varepsilon / 4) * \text{real } K^2$ 
proof –
  have m3:  $3 < M$  using M by linarith
  have dr:  $\bigwedge x. x \in \{3 < .. < \text{real } M\} \implies x \notin DR \implies$ 
    isCont R-r x using DR(2) by simp
  show ?thesis unfolding IR-def by (rule inner-prime-sum-limsup[OF m3 BR
DR(1) dr e4])
qed
obtain BO where BO:  $\bigwedge y. y \in \{1 / 3..2 * \text{lam-r}\} \implies$ 
   $|\text{outer-integrand } y| \leq BO$ 
using outerIntegrand-bounded by blast
obtain DO where DO: countable DO  $\bigwedge y. 0 < y \implies$ 
   $y \notin DO \implies \text{isCont } \text{outer-integrand } y$ 
using outerIntegrand-continuousAt-off by blast
have E3:  $\forall_F K$  in at-top.
   $(\sum p \in \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$ 
     $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}. \text{outer-integrand } (\text{real } p / \text{real } K) * \ln (\text{real } p))$ 
   $\leq (\text{of-rat } I_{\text{out}} + \varepsilon / 4) * \text{real } K$ 
using outer-prime-sum-limsup[of 1 / 3 2 * lam-r outer-integrand BO DO ε /
4] BO DO e4
unfolding outer-integral-value by (auto simp: lam-r-def)
have E4:  $\forall_F K$  in at-top.  $200 * M^2 \leq K \wedge$ 
   $C * (2 * \text{real } K * \ln 4) \leq \varepsilon / 4 * \text{real } K^2$ 
proof –
  have  $\forall_F K$  in at-top.  $\text{nat } (\text{ceiling } (8 * C * \ln 4 / \varepsilon)) \leq K$ 
    by (rule eventually-ge-at-top)
  moreover have  $\forall_F K$  in at-top.  $200 * M^2 \leq K$ 
    by (rule eventually-ge-at-top)
  ultimately show ?thesis
proof (eventually-elim)
  case (elim K)
  have  $8 * C * \ln 4 / \varepsilon \leq \text{real } K$ 
proof –
  have  $\text{real } (\text{nat } (\text{ceiling } (8 * C * \ln 4 / \varepsilon))) \leq \text{real } K$  using elim(1)
    by (simp only: of-nat-le-iff)
  thus ?thesis using real-nat-ceiling-ge[of 8 * C * ln 4 / ε] by linarith
qed
hence  $8 * C * \ln 4 \leq \varepsilon * \text{real } K$  using e
  by (simp add: divide-le-eq mult.commute)
hence a:  $8 * C * \ln 4 * \text{real } K \leq \varepsilon * \text{real } K * \text{real } K$ 
  by (rule mult-right-mono) simp
have b:  $C * (2 * \text{real } K * \ln 4) = (8 * C * \ln 4 * \text{real } K) / 4$  by (simp add:
algebra-simps)
have c:  $\varepsilon / 4 * \text{real } K^2 = (\varepsilon * \text{real } K * \text{real } K) / 4$ 

```

```

    by (simp add: power2-eq-square algebra-simps)
  show ?case using elim(2) a b c by (intro conjI) linarith+
qed
qed
have all:  $\forall_F K$  in at-top.
  ( $\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\}.$ 
     $\text{real } (6 * \text{hof } K * \text{nat-log } p (5 * K) + \text{hof } K * \text{multiplicity } p (24::\text{nat})) *$ 
 $\ln (\text{real } p))$ 
     $\leq (6 * \text{lam-r} / \text{real } M + \varepsilon / 4) * \text{real } K ^ 2 \wedge$ 
    ( $\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge K < p * M \wedge$ 
       $3 * p \leq K\}.$   $\text{real } p * R\text{-r } (\text{real } K / \text{real } p) * \ln (\text{real } p))$ 
     $\leq (IR + \varepsilon / 4) * \text{real } K ^ 2 \wedge$ 
    ( $\sum p \in \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$ 
       $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}.$ 
       $\text{outer-integrand } (\text{real } p / \text{real } K) * \ln (\text{real } p))$ 
     $\leq (\text{of-rat } I_{\text{out}} + \varepsilon / 4) * \text{real } K \wedge$ 
    ( $200 * M ^ 2 \leq K \wedge C * (2 * \text{real } K * \ln 4) \leq \varepsilon / 4 * \text{real } K ^ 2$ )
  by (intro eventually-conj E1 E2 E3 E4)
show ?thesis
  using eventually-compose-filterlim[OF all filterlim-40] eventually-gt-at-top[of 0
:: nat]
proof (eventually-elim)
  case (elim n)
  define K where  $K = 40 * n$ 
  note F = elim(1)[folded K-def]
  have K40:  $40 \text{ dvd } K$  and K0:  $0 < K$  unfolding K-def using elim(2) by
simp-all
  have h1: ( $\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\}.$ 
     $\text{real } (6 * \text{hof } K * \text{nat-log } p (5 * K) + \text{hof } K * \text{multiplicity } p (24::\text{nat})) *$ 
 $\ln (\text{real } p))$ 
     $\leq (6 * \text{lam-r} / \text{real } M + \varepsilon / 4) * \text{real } K ^ 2$ 
  and h2: ( $\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge K < p * M \wedge$ 
     $3 * p \leq K\}.$   $\text{real } p * R\text{-r } (\text{real } K / \text{real } p) * \ln (\text{real } p))$ 
     $\leq (IR + \varepsilon / 4) * \text{real } K ^ 2$ 
  and h3: ( $\sum p \in \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$ 
     $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}.$ 
     $\text{outer-integrand } (\text{real } p / \text{real } K) * \ln (\text{real } p))$ 
     $\leq (\text{of-rat } I_{\text{out}} + \varepsilon / 4) * \text{real } K$ 
  and hKM:  $200 * M ^ 2 \leq K$  and h4:  $C * (2 * \text{real } K * \ln 4) \leq$ 
 $\varepsilon / 4 * \text{real } K ^ 2$ 
  using F by blast+
  let ?P =  $\{p. p \leq 2 * \text{hof } K \wedge \text{prime } p\}$ 
  have hterm:  $\ln (\text{of-rat } (mKM K M)) \leq (\sum p \in$ 
     $?P. \text{mainCoef } K M p * \ln (\text{real } p)) + C * (\sum p \in ?P. \ln (\text{real } p))$ 
  proof -
    have  $\ln (\text{of-rat } (mKM K M)) = (\sum p \in ?P. - \text{of-int } (Lp K M p) * \ln (\text{real}$ 
 $p))$ 
    by (rule log-mKM)
    also have  $\dots \leq (\sum p \in ?P. (\text{mainCoef } K M p + C) * \ln (\text{real } p))$ 

```

```

proof (rule sum-mono)
  fix  $p$  assume  $p \in ?P$ 
  hence  $pp$ :  $\text{prime } p \rightarrow p \leq 2 * \text{hof } K$  by auto
  have  $- \text{of-int } (Lp \ K \ M \ p) \leq \text{mainCoef } K \ M \ p + C$ 
    unfolding  $C\text{-def}$  by (rule neg-Lp-le[OF M C1 C2 hin hout K40 hKM K0
pp])
  moreover have  $0 \leq \ln (\text{real } p)$ 
    by (rule ln-ge-zero) (use prime-ge-1-nat[OF pp(1)] in simp)
  ultimately show  $- \text{of-int } (Lp \ K \ M \ p) * \ln (\text{real } p) \leq (\text{mainCoef } K \ M \ p +$ 
 $C) * \ln (\text{real } p)$ 
    by (rule mult-right-mono)
  qed
also have  $\dots = (\sum p \in ?P. \text{mainCoef } K \ M \ p * \ln (\text{real } p)) + C * (\sum p \in$ 
 $?P. \ln (\text{real } p))$ 
    by (simp add: sum.distrib sum-distrib-left distrib-right)
  finally show ?thesis .
qed
have hsplit:  $(\sum p \in ?P. \text{mainCoef } K \ M \ p * \ln (\text{real } p)) =$ 
 $(\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\}.$ 
 $\text{real } (6 * \text{hof } K * \text{nat-log } p \ (5 * K) + \text{hof } K * \text{multiplicity } p \ (24::\text{nat}))$ 
 $* \ln (\text{real } p))$ 
   $+ (\sum p \in \{p. p \leq K \wedge \text{prime } p \wedge K < p * M \wedge$ 
 $3 * p \leq K\}. \text{real } p * R\text{-r } (\text{real } K / \text{real } p) * \ln (\text{real } p))$ 
   $+ \text{real } K * (\sum p \in \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge$ 
 $\text{prime } p \wedge 1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}.$ 
 $\text{outer-integrand } (\text{real } p / \text{real } K) * \ln (\text{real } p))$ 
proof  $-$ 
  note  $ps = \text{primes-split}[OF \ K40 \ K0 \ M3]$ 
  let  $?A = \{p. p \leq K \wedge \text{prime } p \wedge p * M \leq K\}$ 
  let  $?B = \{p. p \leq K \wedge \text{prime } p \wedge K < p * M \wedge 3 * p \leq K\}$ 
  let  $?C = \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge$ 
 $1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}$ 
  have  $fA$ : finite  $?A$  by (rule finite-subset[of -  $\{..K\}$ ]) auto
  have  $fB$ : finite  $?B$  by (rule finite-subset[of -  $\{..K\}$ ]) auto
  have  $fC$ : finite  $?C$  by (rule finite-subset[of -  $\{.. \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real}$ 
 $K))\}$ ]) auto
  have  $s$ :  $(\sum p \in ?P. \text{mainCoef } K \ M \ p * \ln (\text{real } p)) =$ 
 $(\sum p \in ?A. \text{mainCoef } K \ M \ p * \ln (\text{real } p)) + (\sum p \in$ 
 $?B. \text{mainCoef } K \ M \ p * \ln (\text{real } p))$ 
 $+ (\sum p \in ?C. \text{mainCoef } K \ M \ p * \ln (\text{real } p))$ 
    unfolding  $ps(1)$  sum.union-disjoint[OF finite-UnI[OF  $fA \ fB$ ]  $fC \ ps(3)$ ]
sum.union-disjoint[OF
 $fA \ fB \ ps(2)$ ] ..
  have  $c1$ :  $(\sum p \in ?A. \text{mainCoef } K \ M \ p * \ln (\text{real } p)) =$ 
 $(\sum p \in ?A. \text{real } (6 * \text{hof } K * \text{nat-log } p \ (5 * K) + \text{hof } K *$ 
 $\text{multiplicity } p \ (24::\text{nat})) * \ln (\text{real } p))$ 
    by (rule sum.cong[OF refl]) (simp add:  $ps(4)$ )
  have  $c2$ :  $(\sum p \in ?B. \text{mainCoef } K \ M \ p * \ln (\text{real } p)) = (\sum p \in$ 
 $?B. \text{real } p * R\text{-r } (\text{real } K / \text{real } p) * \ln (\text{real } p))$ 

```

by (rule sum.cong[OF refl]) (simp add: ps(5))
 have c3: $(\sum p \in ?C. \text{mainCoef } K \ M \ p * \ln (\text{real } p)) = \text{real } K * (\sum p \in ?C. \text{outer-integrand } (\text{real } p / \text{real } K) * \ln (\text{real } p))$
 by (subst sum-distrib-left) (rule sum.cong[OF refl], simp add: ps(6))
 show ?thesis unfolding s c1 c2 c3 ..
 qed
 have hlogP: $(\sum p \in ?P. \ln (\text{real } p)) \leq 2 * \text{real } K * \ln 4$ by (rule sum-log-P-le)
 have Kr: $0 \leq \text{real } K$ by simp
 have h3': $\text{real } K * (\sum p \in \{p. p \leq \text{nat } (\text{floor } (2 * \text{lam-r} * \text{real } K)) \wedge \text{prime } p \wedge 1 / 3 * \text{real } K < \text{real } p \wedge \text{real } p \leq 2 * \text{lam-r} * \text{real } K\}. \text{outer-integrand } (\text{real } p / \text{real } K) * \ln (\text{real } p)) \leq (\text{of-rat } Iout + \varepsilon / 4) * \text{real } K ^ 2$
 using mult-left-mono[OF h3 Kr] by (simp add: power2-eq-square algebra-simps)
 have h5: $C * (\sum p \in ?P. \ln (\text{real } p)) \leq C * (2 * \text{real } K * \ln 4)$
 by (rule mult-left-mono[OF hlogP C0])
 have ln (of-rat (mKM K M)) $\leq (\text{of-rat } Iout + 6 * \text{lam-r} / \text{real } M + IR + \varepsilon)$
 *
 real K ^ 2
 using hterm[unfolded hsplit] h1 h2 h3' h4 h5 by (simp add: algebra-simps)
 moreover have $0 < \text{real } K ^ 2$ using K0 by simp
 ultimately show ?case unfolding IR-def[symmetric] K-def[symmetric] by
 (simp add: divide-le-eq)
 qed
 qed

(5.21): $\limsup K^{-2} \log m\{-K, M\} \leq A-M$ for $M \in 40\mathbb{Z}$, $M \geq 40$: split the integral at 20 and at M and use (5.16)-(5.18).

theorem log-mKM-limsup-AM:

assumes $M: 40 \leq M$ $40 \text{ dvd } M$ and $e: 0 < \varepsilon$

shows $\forall_F n \text{ in at-top. } \ln (\text{of-rat } (mKM (40 * n) M)) / \text{real } (40 * n) ^ 2 \leq \text{of-rat } (AM M) + \varepsilon$

proof –

let $?f = \lambda x. R\text{-r } x / x ^ 3$

have $M': 40 \leq \text{real } M$ using M by simp

have int3: set-integrable lborel $\{3..\}$?f by (rule R-integrableOn)

have sub: set-integrable lborel S ?f if $S \subseteq \{3..\}$ $S \in \text{sets borel}$ for S
 by (rule set-integrable-subset[OF int3]) (use that in auto)

have split1: interval-lebesgue-integral lborel (ereal 3) (ereal (real M)) ?f
 = interval-lebesgue-integral lborel (ereal 3) (ereal 20) ?f +
 interval-lebesgue-integral lborel (ereal 20) (ereal (real M)) ?f

proof –

have mn: $\min (\text{ereal } 3) (\min (\text{ereal } 20) (\text{ereal } (\text{real } M))) = \text{ereal } 3$

$\max (\text{ereal } 3) (\max (\text{ereal } 20) (\text{ereal } (\text{real } M))) = \text{ereal } (\text{real } M)$ using

M'

by (simp-all add: min-def max-def)

have interval-lebesgue-integrable lborel (ereal 3) (ereal (real M)) ?f

unfolding interval-lebesgue-integrable-def einterval-eq using M' by (auto intro!: sub)

thus ?thesis using interval-integral-sum[of ereal 3 ereal 20 ereal (real M) ?f]

```

mn by simp
qed
have split2: interval-lebesgue-integral lborel (ereal 20) (ereal (real M)) ?f =
  set-lebesgue-integral lborel {20<.. $\text{real } M$ } ?f
  by (rule interval-integral-Ioc) (use M' in simp)
have split3: set-lebesgue-integral lborel {20<.. $\text{real } M$ } ?f =
  set-lebesgue-integral lborel {20<.. $\text{real } M$ } ?f + set-lebesgue-integral lborel { $\text{real } M$ <.. $\text{real } M$ <.. $\text{real } M$ } ?f
proof -
  have u: {20<.. $\text{real } M$ } = {20<.. $\text{real } M$ }  $\cup$  { $\text{real } M$ <.. $\text{real } M$ <.. $\text{real } M$ } using M' by auto
  have s1: {20<.. $\text{real } M$ }  $\subseteq$  {3.. $\text{real } M$ } by force
  have s2: { $\text{real } M$ <.. $\text{real } M$ <.. $\text{real } M$ }  $\subseteq$  {3.. $\text{real } M$ } using M' by fastforce
  have d: {20<.. $\text{real } M$ }  $\cap$  { $\text{real } M$ <.. $\text{real } M$ <.. $\text{real } M$ } = {} by force
  show ?thesis unfolding u by (rule set-integral-Un[OF d]) (auto intro!: sub s1 s2)
qed
have h16: set-lebesgue-integral lborel {20<.. $\text{real } M$ } ?f  $\leq$  - 2689 / 48000 by (rule R-tail-20-bound)
have h17: - lam-r /  $\text{real } M$  + (2923 / 240 - 1 / 4) /  $\text{real } M^2$  - 32 /  $\text{real } M^3$   $\leq$ 
  set-lebesgue-integral lborel { $\text{real } M$ <.. $\text{real } M$ <.. $\text{real } M$ } ?f
  by (rule R-tail-M-bound) (use M in auto)
have key: of-rat Iout + 6 * lam-r /  $\text{real } M$  +
  interval-lebesgue-integral lborel (ereal 3) (ereal (real M)) ?f  $\leq$  of-rat (AM M)
proof -
  have nA: Astar = Iout + 322437603634266857629 / 7535670527041937280000
  - 2689 / 48000
  unfolding Astar-def Iout-def by simp
  have AM: of-rat (AM M) = of-rat Astar + 7 * (lam-r /  $\text{real } M$ ) - (2923 /
  240 - 1 / 4) /
     $\text{real } M^2$  + 32 /  $\text{real } M^3$ 
  unfolding AM-def lam-q-def lam-r-def
  by (simp add: of-rat-add of-rat-diff of-rat-divide of-rat-mult of-rat-power)
  have As: (of-rat Astar ::  $\text{real}$ ) = of-rat Iout + 322437603634266857629 /
    7535670527041937280000 - 2689 / 48000
  unfolding nA by (simp add: of-rat-add of-rat-diff of-rat-divide)
  have e6: 6 * lam-r /  $\text{real } M$  = 6 * (lam-r /  $\text{real } M$ ) by simp
  have e1: - lam-r /  $\text{real } M$  = - (lam-r /  $\text{real } M$ ) by simp
  show ?thesis unfolding split1 split2 R-integral-3-20 e6 using h16 h17[unfolded
  e1] split3 AM As
  by linarith
qed
show ?thesis
using log-mKM-limsup[OF M(1) e] by (rule eventually-mono) (use key in linarith)
qed
end

```

```

theory Zeta5-Equilibrium-Measure
  imports Zeta5-Asymptotics HOL-Complex-Analysis.Complex-Analysis
    HOL-Decision-Procs.Approximation
begin

```

28 The comparison measure ϱ and its logarithmic potential

Appendix A of the paper: the measure $\varrho = \sum c_j \omega_{[a_j, b_j]}$ of Table 1 as a single density, its logarithmic potential and energy in closed form ((A.1), (A.2)), the analytic half of the potential inequality (A.9) (the tail $[2, \infty)$), and the bounds on the mass of short intervals used in Section 6. Lean theories *AppendixA*, *Lemma61*, *Lemma61Pot*.

28.1 Table 1

$\sum c_j = 37/40 \cdot 10^{12}$: the mass of ϱ is λ . Trivial (simp/eval).

lemma *table1-sum-c*: $(\sum e \leftarrow \text{table1}. \text{snd } (\text{snd } e)) = 37 * 10^{12} \text{ div } 40$
by (*simp add: table1-def*)

Nesting $0 < a-1-6 < \dots < a-1 < b-1 < \dots < b-1-6 < 2$ and $b-j - a-j > 1/225$. Trivial.

lemma *table1-nested*:
 $\text{sorted-wrt } (\lambda x y. y < x) (\text{map fst table1}) \wedge$
 $\text{sorted-wrt } (<) (\text{map } (\text{fst} \circ \text{snd}) \text{ table1}) \wedge$
 $(\forall e \in \text{set table1}. 0 < \text{fst } e \wedge \text{fst } e < \text{fst } (\text{snd } e) \wedge$
 $\text{fst } (\text{snd } e) < 2 * 10^{12} \wedge$
 $10^{12} < 225 * (\text{fst } (\text{snd } e) - \text{fst } e))$
by (*simp add: table1-def*)

lemma *sum-list-dens-measurable* [*measurable*]:
 $(\lambda t. \sum e \leftarrow xs. tC e * \text{arcsine-dens } (tA e) (tB e) t) \in$
 $\text{borel-measurable borel}$
by (*induction xs*) *auto*

lemma *sum-list-map-zero*: $(\bigwedge e. e \in \text{set } xs \implies f e = 0) \implies$
 $(\sum e \leftarrow xs. f e) = (0 :: \text{real})$
by (*induction xs*) *auto*

lemma *nn-integral-sum-list-dens*:
assumes $\forall e \in \text{set } xs. 0 \leq tC e \wedge tA e < tB e$
shows $(\int^+ t. \text{ennreal } (\sum e \leftarrow xs. tC e * \text{arcsine-dens } (tA e) (tB e) t) \partial \text{lborel})$
 $= \text{ennreal } (\sum e \leftarrow xs. tC e)$
using *assms*


```

proof (induction xs)
  case Nil
  thus ?case by simp
next
  case (Cons e xs)
  have c:  $0 \leq tC\ e\ tA\ e < tB\ e$  using Cons.prem by auto
  have r0:  $0 \leq (\sum e \leftarrow xs. tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t)$  for t
    using Cons.prem by (intro sum-list-nonneg) (auto intro!: mult-nonneg-nonneg
      arcsine-dens-nonneg)
  have s0:  $0 \leq (\sum e \leftarrow xs. tC\ e)$  using Cons.prem
    by (intro sum-list-nonneg) auto
  have ( $\int^+ t. \text{ennreal}\ (\sum e' \leftarrow e \# xs. tC\ e' * \text{arcsine-dens}\ (tA\ e')\ (tB\ e')\ t)\ \partial \text{lborel}$ )
    = ( $\int^+ t. \text{ennreal}\ (tC\ e) * \text{ennreal}\ (\text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t)$ 
      +  $\text{ennreal}\ (\sum e \leftarrow xs. tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t)\ \partial \text{lborel}$ )
    using c r0 by (intro nn-integral-cong) (simp add: ennreal-plus[symmetric]
      ennreal-mult arcsine-dens-nonneg)
  also have ... =  $\text{ennreal}\ (tC\ e) * (\int^+ t. \text{ennreal}\ (\text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t)\ \partial \text{lborel})$ 
    + ( $\int^+ t. \text{ennreal}\ (\sum e \leftarrow xs. tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t)\ \partial \text{lborel}$ )
    by (subst nn-integral-add) (auto simp: nn-integral-cmult)
  also have ... =  $\text{ennreal}\ (tC\ e) + \text{ennreal}\ (\sum e \leftarrow xs. tC\ e)$ 
    using Cons by (simp add: arcsine-dens-integral[OF c(2)])
  also have ... =  $\text{ennreal}\ (\sum e' \leftarrow e \# xs. tC\ e')$  using c s0
    by (simp add: ennreal-plus)
  finally show ?case .
qed

```

lemma table1-props: $\forall e \in \text{set table1}. 0 \leq tC\ e \wedge tA\ e < tB\ e \wedge 0 < tA\ e \wedge tB\ e < 2$
by (simp add: table1-def tA-def tB-def tC-def)

lemma table1-sum-tC: $(\sum e \leftarrow \text{table1}. tC\ e) = \text{lam-r}$
by (simp add: table1-def tC-def lam-r-def)

theorem rho-univ: $\text{emeasure rho UNIV} = \text{ennreal lam-r}$

```

proof -
  have  $\text{emeasure rho UNIV} = (\int^+ t. \text{ennreal}\ (\sum e \leftarrow \text{table1}. tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t)\ \partial \text{lborel})$ 
    unfolding rho-def by (subst emeasure-density) auto
  also have ... =  $\text{ennreal lam-r}$ 
    using table1-props by (subst nn-integral-sum-list-dens) (auto simp: table1-sum-tC)
  finally show ?thesis .
qed

```

theorem rho-compl: $\text{emeasure rho} (-\{0 < .. < 2\}) = 0$

```

proof –
  have z: ( $\sum e \leftarrow \text{table1}. tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t = 0$ )
    if  $t: t \notin \{0 < .. < 2\}$  for  $t$ 
  proof (rule sum-list-map-zero)
    fix  $e$  assume  $e: e \in \text{set table1}$ 
    have  $\neg (tA\ e < t \wedge t < tB\ e)$  using table1-props  $e\ t$  by fastforce
    thus  $tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t = 0$  by (auto simp: arcsine-dens-def)
  qed
  have emeasure rho ( $-\{0 < .. < 2\}$ )
    = ( $\int^+ t. \text{ennreal} (\sum e \leftarrow \text{table1}. tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t) * \text{indicator}\ (-\{0 < .. < 2\})\ t\ \partial \text{lborel}$ )
    unfolding rho-def by (subst emeasure-density) auto
  also have  $\dots = (\int^+ t. 0\ \partial \text{lborel})$ 
  proof (subst nn-integral-cong [where  $v = \lambda t. 0$ ])
    fix  $t :: \text{real}$ 
    show  $\text{ennreal} (\sum e \leftarrow \text{table1}. tC\ e * \text{arcsine-dens}\ (tA\ e)\ (tB\ e)\ t) * \text{indicator}\ (-\{0 < .. < 2\})\ t = 0$ 
    by (cases  $t \in \{0 < .. < 2\}$ ) (simp-all add: z)
  qed simp
  finally show ?thesis by simp
qed

```

(A.2): $I(\varrho) = \sum -j (S\text{-}j^{\wedge}2 - S\text{-}\{j-1\}^{\wedge}2) \log((b-j - a-j)/4)$ (nesting makes the j -th potential constant on every earlier interval).

The sixteen arcsine potentials $U\text{-}k$ and their constant values $L\text{-}k$ on their intervals.

```

definition  $Uk :: \text{nat} \Rightarrow \text{real} \Rightarrow \text{real}$  where  $Uk\ k\ t = \text{logPot}\ (\text{arcsine}\ (tA\ (\text{table1}\ !\ k))\ (tB\ (\text{table1}\ !\ k)))\ t$ 
definition  $Lk :: \text{nat} \Rightarrow \text{real}$  where  $Lk\ k = \ln ((tB\ (\text{table1}\ !\ k) - tA\ (\text{table1}\ !\ k)) / 4)$ 
definition  $cK :: \text{nat} \Rightarrow \text{real}$  where  $cK\ k = tC\ (\text{table1}\ !\ k)$ 

```

```

lemma table1-nth:  $k < 16 \implies 0 < tA\ (\text{table1}\ !\ k) \wedge tA\ (\text{table1}\ !\ k) < tB\ (\text{table1}\ !\ k) \wedge tB\ (\text{table1}\ !\ k) < 2 \wedge 0 \leq cK\ k$ 

```

```

proof –
  assume  $k: k < 16$ 
  have  $l: \text{length table1} = 16$  by (simp add: table1-def)
  have  $m: \text{table1}\ !\ k \in \text{set table1}$  using nth-mem[of  $k\ \text{table1}$ ]  $k\ l$  by simp
  show ?thesis using table1-props  $m$  unfolding cK-def by blast
qed

```

```

lemma table1-nested-nth:  $j \leq k \implies k < 16 \implies tA\ (\text{table1}\ !\ k) \leq tA\ (\text{table1}\ !\ j) \wedge tB\ (\text{table1}\ !\ j) \leq tB\ (\text{table1}\ !\ k)$ 

```

```

proof –
  assume  $jk: j \leq k$  and  $k: k < 16$ 
  have  $l: \text{length table1} = 16$  by (simp add: table1-def)
  have  $s1: \text{sorted-wrt}\ (\lambda x\ y. y < x)\ (\text{map}\ \text{fst}\ \text{table1})$ 
    using table1-nested

```

```

    by meson
  have s2: sorted-wrt (<) (map (fst ∘ snd) table1)
    using table1-nested
  by meson
show ?thesis
proof (cases j = k)
  case True
  thus ?thesis
    by simp
next
  case False
  hence jk': j < k using jk
    by linarith
  have fst (table1 ! k) < fst (table1 ! j) using sorted-wrt-nth-less[OF s1 jk'] k l
    using jk'
  by (simp add: l)
  moreover have fst (snd (table1 ! j)) < fst (snd (table1 ! k))
    using sorted-wrt-nth-less[OF s2 jk'] k l
  using jk'
  by (simp add: l)
  ultimately show ?thesis unfolding tA-def tB-def
    by simp
qed
qed

```

lemma *Uk-eq*: $k < 16 \implies Uk\ k\ t = (if\ tA\ (table1\ !\ k) \leq t \wedge t \leq tB\ (table1\ !\ k) \text{ then } Lk\ k$
 $\text{ else } \ln\ ((|t - (tA\ (table1\ !\ k) + tB\ (table1\ !\ k)) / 2| + \text{sqrt}\ ((t - tA\ (table1\ !\ k)) * (t - tB\ (table1\ !\ k)))) / 2))$
unfolding *Uk-def Lk-def* **using** *table1-nth* **by** (*intro arcsine-potential*) *simp*

lemma *Uk-const*: $k < 16 \implies tA\ (table1\ !\ k) \leq t \implies t \leq tB\ (table1\ !\ k) \implies Uk\ k\ t = Lk\ k$
by (*simp add: Uk-eq*)

lemma *Uk-bounds*: $k < 16 \implies 0 < t \implies t < 2 \implies Lk\ k \leq Uk\ k\ t \wedge Uk\ k\ t \leq \ln\ 2$

proof –
 assume *k*: $k < 16$ and *t0*: $0 < t$ and *t2*: $t < 2$
 define *a* where $a = tA\ (table1\ !\ k)$
 define *b* where $b = tB\ (table1\ !\ k)$
 have *ab1*: $0 < a$ **using** *table1-nth*[*OF k*] **unfolding** *a-def b-def*
 by *blast*
 have *ab2*: $a < b$ **using** *table1-nth*[*OF k*] **unfolding** *a-def b-def*
 by *blast*
 have *ab3*: $b < 2$ **using** *table1-nth*[*OF k*] **unfolding** *a-def b-def*
 by *blast*
 note *ab* = *ab1 ab2 ab3*
 have *L*: $Lk\ k = \ln\ ((b - a) / 4)$ **unfolding** *Lk-def a-def b-def*

```

    by meson
  have ba:  $0 < (b - a) / 4$  using ab
    by simp
  have L2:  $Lk\ k \leq \ln\ 2$  unfolding L using ab ba ln-le-cancel-iff[of  $(b - a) / 4$ 
2]
    by simp
  show ?thesis
  proof (cases  $a \leq t \wedge t \leq b$ )
    case True
      have Uk k t = Lk k using Uk-const[OF k] True unfolding a-def b-def
        by blast
      thus ?thesis using L2
        by auto
    next
      case False
        define X where  $X = (|t - (a + b) / 2| + \text{sqrt}((t - a) * (t - b))) / 2$ 
        have U:  $Uk\ k\ t = \ln\ X$  using Uk-eq[OF k, of t] False unfolding X-def a-def
        b-def
          by presburger
        have far:  $(b - a) / 2 \leq |t - (a + b) / 2|$  using False
          by argo
        have near:  $|t - (a + b) / 2| < 2$  using ab t0 t2
          by argo
        have pr:  $(t - a) * (t - b) = (t - (a + b) / 2)^2 - ((b - a) / 2)^2$ 
          by (simp add: power2-eq-square field-simps)
        have p0:  $0 \leq (t - a) * (t - b)$ 
        proof (cases  $t < a$ )
          case True
            thus ?thesis using ab
              by (fastforce intro: mult-nonpos-nonpos)
          next
            case False
              hence  $b < t$  using  $\langle \neg (a \leq t \wedge t \leq b) \rangle$ 
                by linarith
              thus ?thesis using ab
                by (auto intro: mult-nonneg-nonneg)
        qed
        have sq:  $\text{sqrt}((t - a) * (t - b)) \leq |t - (a + b) / 2|$ 
          using real-sqrt-le-mono[of  $(t - a) * (t - b)$   $(t - (a + b) / 2)^2$ ] pr
        real-sqrt-abs
          by simp
        have  $(b - a) / 4 \leq X$  unfolding X-def using far real-sqrt-ge-zero[OF p0]
          by argo
        moreover have  $X \leq 2$  unfolding X-def using sq near
          by simp
        ultimately show ?thesis unfolding U L using ba ln-le-cancel-iff
          by simp
      qed
    qed
  qed

```

lemma *Uk-measurable* [*measurable*]: $k < 16 \implies Uk\ k \in \text{borel-measurable borel}$
proof –
 assume $k: k < 16$
 have $e: Uk\ k = (\lambda t. \text{if } tA\ (table1\ !\ k) \leq t \wedge$
 $t \leq tB\ (table1\ !\ k) \text{ then } Lk\ k \text{ else } \ln\ ((|t - (tA\ (table1\ !\ k) + tB\ (table1\ !\ k))| /$
 $2| + \text{sqrt}\ ((t - tA\ (table1\ !\ k)) * (t - tB\ (table1\ !\ k)))) / 2)) \text{ using } Uk\text{-eq}[OF$
 $k]$
 by (*simp add: fun-eq-iff*)
 show ?thesis **unfolding** e **by** *measurable*
qed

lemma *Uk-integrable*: $j < 16 \implies k < 16 \implies$
 $\text{integrable}\ (\arcsine\ (tA\ (table1\ !\ j))\ (tB\ (table1\ !\ j)))\ (Uk\ k)$
proof –
 assume $j: j < 16$ and $k: k < 16$
 define a **where** $a = tA\ (table1\ !\ j)$
 define b **where** $b = tB\ (table1\ !\ j)$
 have $ab: 0 < a\ a < b\ b < 2$ **using** *table1-nth*[*OF j*] **unfolding** $a\text{-def}\ b\text{-def}$ **by**
auto
 have $pw: \text{norm}\ (Uk\ k\ x) \leq |Lk\ k| + \ln\ 2$ **if** $a < x\ x < b$ **for** x
proof –
 have $x0: 0 < x\ x < 2$ **using** *that ab* **by** *auto*
 have $bb: Lk\ k \leq Uk\ k\ x\ Uk\ k\ x \leq \ln\ 2$ **using** *Uk-bounds*[*OF k x0*] **by** *auto*
 show ?thesis **using** bb *ln-gt-zero*[*of 2::real*] **unfolding** *real-norm-def*
by *linarith*
qed
 have $bd: AE\ x\ \text{in}\ \arcsine\ a\ b. \text{norm}\ (Uk\ k\ x) \leq |Lk\ k| + \ln\ 2$
using *arcsine-AE*[*of a b*] pw
by (*fast elim: AE-mp*)
 have $mU: Uk\ k \in \text{borel-measurable}\ (\arcsine\ a\ b)$ **using** *Uk-measurable*[*OF k*]
unfolding *arcsine-def* **by** *simp*
 show ?thesis **using** *finite-measure.integrable-const-bound*[*OF arcsine-finite*[*OF*
 $ab(2)$] $bd\ mU$]
unfolding $a\text{-def}\ b\text{-def}$.
qed

$\int U\text{-}k\ d\omega\text{-}j = L\text{-}\{max(j,k)\}$: by nesting the outer potential is constant on the inner interval; otherwise swap (Fubini).

lemma *Uk-integral*:
 assumes $j < 16\ k < 16$
 shows $\text{integral}^L\ (\arcsine\ (tA\ (table1\ !\ j))\ (tB\ (table1\ !\ j)))\ (Uk\ k) = Lk\ (max\ j\ k)$
proof (*cases j ≤ k*)
 case *True*
 have $const: AE\ t\ \text{in}\ \arcsine\ (tA\ (table1\ !\ j))\ (tB\ (table1\ !\ j)).\ Uk\ k\ t = Lk\ k$
using *arcsine-AE*[*of tA (table1 ! j) tB (table1 ! j)*] *table1-nested-nth*[*OF True*
 $assms(2)$]
 $Uk\ const$ [*OF assms(2)*]

```

    by (force elim: AE-mp)
  show ?thesis
    using integral-cong-AE[OF - - const] Uk-measurable[OF assms(2)] arcsine-univ[of
  tA (table1 !
    j) tB (table1 ! j)] table1-nth[OF assms(1)] max-absorb2[OF True]
    unfolding arcsine-def
    by (simp add: measure-def)
next
case False
have bj: tA (table1 ! j) < tB (table1 ! j) 0 < tA (table1 ! j) tB (table1 ! j) < 2
  using table1-nth[OF assms(1)] by auto
have bk: tA (table1 ! k) < tB (table1 ! k) 0 < tA (table1 ! k) tB (table1 ! k) <
  2
  using table1-nth[OF assms(2)] by auto
have bnd: Lk k ≤ logPot (arcsine (tA (table1 ! k)) (tB (table1 ! k))) x if 0 < x
  x < 2
  for x
  using Uk-bounds[OF assms(2) that] unfolding Uk-def by simp
have swap: integralL (arcsine (tA (table1 ! j)) (tB (table1 ! j))) (Uk k) =
  integralL (arcsine (tA (table1 ! k)) (tB (table1 ! k))) (Uk j)
  unfolding Uk-def[abs-def] using arcsine-swap[OF bj bk bnd]
  by simp
have kj: k ≤ j using False by simp
have const: AE t in arcsine (tA (table1 ! k)) (tB (table1 ! k)). Uk j t = Lk j
  using arcsine-AE[of tA (table1 ! k) tB (table1 ! k)] table1-nested-nth[OF kj
  assms(1)]
  Uk-const[OF assms(1)]
  by (force elim: AE-mp)
have integralL (arcsine (tA (table1 ! k)) (tB (table1 ! k))) (Uk j) = Lk j
  using integral-cong-AE[OF - - const] Uk-measurable[OF assms(1)] arcsine-univ[of
  tA (table1 !
    k) tB (table1 ! k)] table1-nth[OF assms(2)]
    unfolding arcsine-def
    by (simp add: measure-def)
thus ?thesis using swap max-absorb1[OF kj]
  by simp
qed

```

$U^{\wedge}_{\varrho}(t) = \sum -k \ c-k \ U-k(t)$ and $I(\varrho) = \sum -\{j,k\} \ c-j \ c-k \ L-\{max(j,k)\}$.

lemma *logPot-rho*: $logPot \ rho \ t = (\sum \ k < 16. \ cK \ k * Uk \ k \ t)$

proof –

```

  have nnr0: 0 ≤ (∑ e ← table1. tC e * arcsine-dens (tA e) (tB e) u) for u
    using table1-props
    by (fastforce intro: sum-list-nonneg mult-nonneg-nonneg arcsine-dens-nonneg)
  have nnr: AE u in lborel. 0 ≤ (∑ e ← table1. tC e *
    arcsine-dens (tA e) (tB e) u) using nnr0 by simp
  have logPot rho t = (∫ u. (∑ e ← table1. tC e *
    arcsine-dens (tA e) (tB e) u) * ln |t - u| ∂lborel)
    unfolding logPot-def rho-def

```

```

using nnr0 integral-density[of  $\lambda u. \ln |t -$ 
   $u|$  lborel  $(\lambda u. \sum e \leftarrow \text{table1}. tC\ e *$ 
   $\arcsine\text{-dens}\ (tA\ e)\ (tB\ e)\ u)]\ \text{dens}\text{-}\ln\text{-meas}[of\ t]\ \text{sum}\text{-list}\text{-dens}\text{-measurable}
  nnr
by simp
also have ... =  $(\sum k < 16. cK\ k *$ 
   $(\int u. \arcsine\text{-dens}\ (tA\ (\text{table1}\ !\ k))\ (tB\ (\text{table1}\ !\ k))\ u * \ln |t -$ 
   $u| \partial \text{lborel}))$ 
proof -
  have l:  $\text{length}\ \text{table1} = 16$  by (simp add: table1-def)
  have sl:  $(\sum e \leftarrow \text{table1}. tC\ e * \arcsine\text{-dens}\ (tA\ e)\ (tB\ e)\ u) = (\sum k <$ 
   $16. cK\ k * \arcsine\text{-dens}\ (tA\ (\text{table1}\ !\ k))\ (tB\ (\text{table1}\ !\ k))\ u)$  for u
    using sum-list-sum-nth[of map  $(\lambda e. tC\ e * \arcsine\text{-dens}\ (tA\ e)\ (tB\ e)\ u)$ 
    table1] l
    unfolding cK-def
    by (simp add: atLeast0LessThan)
  have ik: integrable lborel  $(\lambda u. cK\ k *$ 
   $(\arcsine\text{-dens}\ (tA\ (\text{table1}\ !\ k))\ (tB\ (\text{table1}\ !\ k))\ u * \ln |t - u|))$ 
  if  $k \in \{..<16\}$  for k
    using dens-ln-integrable[of tA (table1 ! k) tB (table1 ! k) t] table1-nth[of k]
  that
    by fast
  show ?thesis unfolding sl sum-distrib-right
    using Bochner-Integration.integral-sum[of  $\{..<16\}$  lborel  $\lambda k\ u. cK\ k *$ 
     $(\arcsine\text{-dens}\ (tA\ (\text{table1}\ !\ k))\ (tB\ (\text{table1}\ !\ k))\ u * \ln |t - u|),\ OF\ ik]$ 
    by (fastforce simp: mult.assoc)
  qed
also have ... =  $(\sum k < 16. cK\ k * Uk\ k\ t)$ 
  by (simp add: Uk-def dens-ln-integral)
  finally show ?thesis .
qed

lemma dens-Uk-integrable:
  assumes j:  $j < 16$  and k:  $k < 16$ 
  shows integrable lborel  $(\lambda t. \arcsine\text{-dens}\ (tA\ (\text{table1}\ !\ j))\ (tB\ (\text{table1}\ !\ j))\ t * Uk\ k\ t)$ 
  proof -
    have md:  $(\lambda t. \arcsine\text{-dens}\ (tA\ (\text{table1}\ !\ j))\ (tB\ (\text{table1}\ !\ j))\ t) \in$ 
    borel-measurable borel unfolding arcsine-dens-def by measurable
    have nn:  $AE\ t\ \text{in}\ \text{lborel}. 0 \leq \arcsine\text{-dens}\ (tA\ (\text{table1}\ !\ j))\ (tB\ (\text{table1}\ !\ j))\ t$ 
    by (simp add: arcsine-dens-def)
    show ?thesis using Uk-integrable[OF j k] integrable-density[of Uk k lborel arcsine-dens (tA
    (table1 ! j)) (tB (table1 ! j))] Uk-measurable[OF k] md nn
    unfolding arcsine-def
    by auto
  qed

lemma dens-Uk-integral:$ 
```

assumes $j: j < 16$ **and** $k: k < 16$
shows $(\int t. \text{arcsine-dens } (tA \text{ (table1 ! j)}) (tB \text{ (table1 ! j)}) t * U_k k t \partial \text{lborel}) = Lk \text{ (max j k)}$
proof –
have $md: (\lambda t. \text{arcsine-dens } (tA \text{ (table1 ! j)}) (tB \text{ (table1 ! j)}) t) \in \text{borel-measurable borel}$ **unfolding** arcsine-dens-def **by** measurable
have $nn: AE t \text{ in lborel. } 0 \leq \text{arcsine-dens } (tA \text{ (table1 ! j)}) (tB \text{ (table1 ! j)}) t$ **by** $(\text{simp add: arcsine-dens-def})$
show $?thesis$ **using** $U_k\text{-integral}[OF j k]$ $\text{integral-density[of } U_k k \text{ lborel arcsine-dens (tA (table1 ! j)) (tB (table1 ! j))]}$ $U_k\text{-measurable}[OF k]$ md nn **unfolding** arcsine-def **by** force
qed

lemma $\text{logEnergy-rho: logEnergy rho} = (\sum j < 16. \sum k < 16. cK j * cK k * Lk \text{ (max j k)})$
proof –
have $fe: \text{logPot rho} = (\lambda t. \sum k < 16. cK k * U_k k t)$ **by** $(\text{rule ext, rule logPot-rho})$
have $mf: (\lambda t. \sum k < 16. cK k * U_k k t) \in \text{borel-measurable borel}$ **by** $(\text{force intro: borel-measurable-sum borel-measurable-times } U_k\text{-measurable})$
have $nnr0: 0 \leq (\sum e \leftarrow \text{table1. } tC e * \text{arcsine-dens } (tA e) (tB e) u)$ **for** u **using** table1-props **by** $(\text{fastforce intro: sum-list-nonneg mult-nonneg-nonneg arcsine-dens-nonneg})$
have $nnr: AE u \text{ in lborel. } 0 \leq (\sum e \leftarrow \text{table1. } tC e * \text{arcsine-dens } (tA e) (tB e) u)$ **using** $nnr0$ **by** simp
have $A: \text{logEnergy rho} = (\int t. (\sum e \leftarrow \text{table1. } tC e * \text{arcsine-dens } (tA e) (tB e) t) * (\sum k < 16. cK k * U_k k t) \partial \text{lborel})$ **unfolding** logEnergy-def fe **unfolding** rho-def **using** $\text{integral-density[of } \lambda t. \sum k < 16. cK k * U_k k t \text{ lborel } (\lambda u. \sum e \leftarrow \text{table1. } tC e * \text{arcsine-dens } (tA e) (tB e) u)]}$ $mf \text{ sum-list-dens-measurable nnr}$ **by** fastforce
have $l: \text{length table1} = 16$ **by** $(\text{simp add: table1-def})$
have $sl: (\sum e \leftarrow \text{table1. } tC e * \text{arcsine-dens } (tA e) (tB e) t) = (\sum j < 16. cK j * \text{arcsine-dens } (tA \text{ (table1 ! j)}) (tB \text{ (table1 ! j)}) t)$ **for** t **using** $\text{sum-list-sum-nth[of map } (\lambda e. tC e * \text{arcsine-dens } (tA e) (tB e) t) \text{ table1}]$
 l
unfolding $cK\text{-def}$ **by** $(\text{simp add: atLeast0LessThan})$
have $B: (\sum e \leftarrow \text{table1. } tC e * \text{arcsine-dens } (tA e) (tB e) t) * (\sum k < 16. cK k * U_k k t) = (\sum j < 16. \sum k < 16. cK j * cK k * (\text{arcsine-dens } (tA \text{ (table1 ! j)}) (tB \text{ (table1 ! j)}) t * U_k k t))$
for t
unfolding $sl \text{ sum-product}$ **by** $(\text{simp add: algebra-simps})$
have $C: (\int t. (\sum j < 16. \sum k < 16. cK j * cK k * (\text{arcsine-dens } (tA \text{ (table1 ! j)}) (tB \text{ (table1 ! j)}) t * U_k k t) \partial \text{lborel}) = (\sum j < 16. \sum k < 16. cK j * cK k * Lk \text{ (max j k)})$


```

proof –
  have ij: integrable lborel ( $\lambda t. \sum k < 16. cK j * cK k * (arcsine-dens (tA (table1 ! j)) (tB (table1 ! j)) t * Uk k t)$ )
  if  $j \in \{..<16\}$  for j
  using dens-Uk-integrable that
  by (fast intro: Bochner-Integration.integrable-sum integrable-mult-right)
  have c2: ( $\int t. (\sum k < 16. cK j * cK k * (arcsine-dens (tA (table1 ! j)) (tB (table1 ! j)) t * Uk k t)) \partial lborel$ ) = ( $\sum k < 16. cK j * cK k * Lk (max j k)$ )
  if  $j \in \{..<16\}$  for j
  using Bochner-Integration.integral-sum[of  $\{..<16\}$  lborel  $\lambda k t. cK j * cK k * (arcsine-dens (tA (table1 ! j)) (tB (table1 ! j)) t * Uk k t)$ ] dens-Uk-integrable dens-Uk-integral that
  by auto
  show ?thesis using Bochner-Integration.integral-sum[of  $\{..<16\}$  lborel, OF ij]
c2
  by fastforce
  qed
  show ?thesis unfolding A B C ..
qed

lemma sum-max-regroup:
  fixes c L :: nat  $\Rightarrow$  real
  shows ( $\sum j < n. \sum k < n. c j * c k * L (max j k)$ ) = ( $\sum j < n. ((\sum i \leq j. c i) ^ 2 - (\sum i < j. c i) ^ 2) * L j$ )
proof (induction n)
  case 0
  show ?case by simp
next
  case (Suc n)
  define S where  $S = (\sum i < n. c i)$ 
  have g: ( $\sum j < Suc n. \sum k < Suc n. F j k$ ) = ( $\sum j < n. \sum k < n. F j k$ ) + ( $\sum j < n. F j n$ ) + ( $\sum k < n. F n k$ ) +  $F n n$ 
  for F :: nat  $\Rightarrow$  nat  $\Rightarrow$  real
  by (simp add: sum.distrib)
  have r1a: ( $\sum j < n. c j * c n * L (max j n)$ ) = ( $\sum j < n. c n * L n * c j$ )
  proof (rule sum.cong[OF refl])
    fix j assume  $j \in \{..<n\}$ 
    thus  $c j * c n * L (max j n) = c n * L n * c j$ 
    by simp
  qed
  have r2a: ( $\sum k < n. c n * c k * L (max n k)$ ) = ( $\sum k < n. c n * L n * c k$ )
  proof (rule sum.cong[OF refl])
    fix k assume  $k \in \{..<n\}$ 
    thus  $c n * c k * L (max n k) = c n * L n * c k$ 
    by simp
  qed
  have r3: ( $\sum k < n. c n * L n * c k$ ) =  $c n * L n * S$  unfolding S-def
    by (metis sum-distrib-left)

```

```

have s: (∑ i ≤ n. c i) = S + c n unfolding S-def
  by (simp add: lessThan-Suc-atMost[symmetric])
have (∑ j < Suc n. ∑ k < Suc n. c j * c k * L (max j k)) = (∑ j < n. ∑ k
<
  n. c j * c k * L (max j k)) + 2 * (c n * L n * S) + c n * c n * L n
using g[of λj k. c j * c k * L (max j k)] r1a r2a r3
by linarith
also have ... = (∑ j < n. ((∑ i ≤ j. c i) ^ 2 - (∑ i <
  j. c i) ^ 2) * L j) + ((S + c n) ^ 2 - S ^ 2) * L n
using Suc.IH
by (simp add: power2-eq-square algebra-simps)
also have ... = (∑ j < Suc n. ((∑ i ≤ j. c i) ^ 2 - (∑ i <
  j. c i) ^ 2) * L j)
using s unfolding S-def
by simp
finally show ?case .
qed

```

```

lemma Spart-eq: j ≤ 16 ⇒ Spart j = (∑ i < j. cK i)
proof -
  assume j: j ≤ 16
  have l: length table1 = 16 by (simp add: table1-def)
  have Spart j = (∑ i = 0..<length (map tC (take j table1)). map tC (take j
table1) ! i)
    unfolding Spart-def using sum-list-sum-nth[of map tC (take j table1)]
    by meson
  also have ... = (∑ i < j. cK i)
  proof (rule sum.cong)
    show {0..<length (map tC (take j table1))} = {..<j} using j l
    by auto
    fix i assume i ∈ {..<j}
    thus map tC (take j table1) ! i = cK i using j l unfolding cK-def
    by simp
  qed
  finally show ?thesis .
qed

```

$$(A.2): I(\varrho) = \sum -j (S-j^2 - S-\{j-1\}^2) \log((b-j - a-j)/4).$$

```

theorem energy-A2:
  logEnergy rho = (∑ j < 16. (Spart (j + 1) ^ 2 - Spart j ^ 2) * ln ((tB (table1
! j) -
  tA (table1 ! j)) / 4))
  unfolding logEnergy-rho sum-max-regroup Lk-def[symmetric]
  by (rule sum.cong[OF refl]) (simp add: Spart-eq Lk-def lessThan-Suc-atMost)

```

28.2 Elementary enclosures

Partial sums of the atanh series.

definition atS :: real ⇒ nat ⇒ real **where**

$atS\ z\ m = (\sum\ k < m. 2 * (1 / (2 * real\ k + 1)) * z ^{(2 * k + 1)})$

(A.3): $S\text{-}m(z) \leq \log((1+z)/(1-z)) \leq S\text{-}m(z) + 2z^{2m+1}/((2m+1)(1-z^2))$
 for $0 \leq z < 1$.

lemma *log-series-bounds*:
assumes $0 \leq z < 1$
shows $atS\ z\ m \leq \ln((1+z)/(1-z)) \wedge$
 $\ln((1+z)/(1-z)) \leq atS\ z\ m + 2 * z^{(2 * m + 1)} / ((2 * real\ m + 1) * (1 - z^2))$

proof –
define x **where** $x = (1 + z) / (1 - z)$
define f **where** $f\ n = 2 * z^{(2 * n + 1)} / real\ (2 * n + 1)$ **for** n
have $z1$: $0 < 1 - z$ **using** *assms* **by** *simp*
have $x0$: $0 < x$ **using** *assms* **by** (*simp add: x-def*)
have xz : $(x - 1) / (x + 1) = z$ **using** $z1$ **by** (*simp add: x-def field-simps*)
have S : f *sums* $\ln\ x$ **using** *ln-series-quadratic*[*OF x0*] **unfolding** $xz\ f\text{-def}$.
have sf : *summable* f **using** S **by** (*rule sums-summable*)
have L : $\ln\ x = \text{suminf}\ f$ **using** S **by** (*simp add: sums-iff*)
have $f0$: $0 \leq f\ n$ **for** n **using** *assms* **by** (*simp add: f-def*)
have $atS\text{-eq}$: $atS\ z\ m = (\sum\ k < m. f\ k)$ **unfolding** $atS\text{-def}\ f\text{-def}$ **by** (*simp add: field-simps*)
have low : $(\sum\ k < m. f\ k) \leq \text{suminf}\ f$ **by** (*rule sum-le-suminf*[*OF sf*]) (*simp-all add: f0*)
define c **where** $c = 2 * z^{(2 * m + 1)} / (2 * real\ m + 1)$
define g **where** $g\ n = c * (z^2)^n$ **for** n
have $zz0$: $0 \leq z^2$ **by** *simp*
have zz : $z^2 < 1$ **using** *assms* **by** (*simp add: power-less-one-iff*)
have G : g *sums* $(c * (1 / (1 - z^2)))$
unfolding $g\text{-def}$ **by** (*rule sums-mult*) (*rule geometric-sums, use zz zz0 in simp*)
have fg : $f\ (n + m) \leq g\ n$ **for** n
proof –
have e : $z^{(2 * (n + m) + 1)} = z^{(2 * m + 1)} * (z^2)^n$
by (*simp add: power-add[symmetric] power-mult[symmetric] algebra-simps*)
have $f\ (n + m) = 2 * z^{(2 * m + 1)} * (z^2)^n / real\ (2 * (n + m) + 1)$
unfolding $f\text{-def}\ e$ **by** *simp*
also have $\dots \leq 2 * z^{(2 * m + 1)} * (z^2)^n / (2 * real\ m + 1)$
using *assms* **by** (*intro divide-left-mono*) *auto*
also have $\dots = g\ n$ **by** (*simp add: g-def c-def*)
finally show *?thesis* .
qed
have $tail$: $\text{suminf}\ (\lambda n. f\ (n + m)) = \text{suminf}\ f - (\sum\ k < m. f\ k)$
by (*rule suminf-minus-initial-segment*[*OF sf*])
have $\text{suminf}\ (\lambda n. f\ (n + m)) \leq \text{suminf}\ g$
by (*rule suminf-le*[*OF fg*]) (*use sf G in {auto simp: summable-iff-shift sums-summable*[*OF G*])
also have $\text{suminf}\ g = c * (1 / (1 - z^2))$ **using** G **by** (*simp add: sums-iff*)
finally have up : $\text{suminf}\ f \leq (\sum\ k < m. f\ k) + c * (1 / (1 - z^2))$ **using** $tail$

by *simp*
 have cc: $c * (1 / (1 - z^2)) = 2 * z^{(2 * m + 1)} / ((2 * \text{real } m + 1) * (1 - z^2))$
 by (*simp add: c-def*)
 show ?thesis unfolding *x-def[symmetric]* *L atS-eq* using *low up cc* by *simp*
 qed

(A.10): the enclosures of $I(\varrho)$ and C^* (18 logarithms to 25 digits; in Isabelle either *approximation* or (A.3) evaluated by *simp*). The endpoints combine to exactly $-1366995564511/10^{12}$, so (6.4) rests on the strict enclosures.

The explicit value of $I(\varrho)$ as a combination of sixteen logarithms of rationals.

lemma *logEnergy-explicit*:

logEnergy rho = $(\sum j < 16. (\text{Spart } (j + 1)^2 - \text{Spart } j^2) * \ln ((tB (\text{table1 } ! j) - tA (\text{table1 } ! j)) / 4))$
 by (*rule energy-A2*)

A finite sum over sixteen indices, unfolded.

lemma *sum16*: $(\sum j < 16. f j) = f 0 + f 1 + f 2 + f 3 + f 4 + f 5 + f 6 + f 7 + f 8 + f 9 + f 10 + f 11 + f 12 + f 13 + f 14 + f 15$
 for $f :: \text{nat} \Rightarrow \text{real}$
 by (*simp add: eval-nat-numeral*)

28.3 (A.9): the potential inequality on the half-line

The certified-checker part. Lean defines computable rational upper/lower bounds for $\sqrt{\cdot}$, $\log$, $\arctan$ (soundness lemmas), a checker $\text{chk } l r = (2 \cdot \text{Ubnd } l r - \text{VL } l r \leq \text{Bd})$ with $\text{Bd} = -66451/10^4$, and checks a chain of 1,052 intervals covering $[0, 2]$ by kernel evaluation. In Isabelle the bound functions can be *lb-ln/ub-ln*, *lb-sqrt/ub-sqrt*, *lb-arctan/ub-arctan* from HOL-Decision_Procs.Approximation_Bounds (soundness: *bnds-ln*, *bnds-sqrt*, *bnds-arctan*); the evaluation is by *eval* (oracle) or *code-simp* (kernel).

Closed forms: $\int -0^c \log(t + u^2) du = c \log(t + c^2) - 2c + 2\sqrt{t} \arctan(c/\sqrt{t})$ for $t > 0$, and $2(c \log c - c)$ at $t = 0$.

lemma *Ic-closed*:

assumes $0 < t < c$
 shows $\text{integral } \{0..c\} (\lambda u. \ln(t + u^2)) = c * \ln(t + c^2) - 2 * c + 2 * \sqrt{t} * \arctan(c / \sqrt{t})$
 proof -
 define *F* where $F u = u * \ln(t + u^2) - 2 * u + 2 * \sqrt{t} * \arctan(u / \sqrt{t})$ for *u*
 have *st*: $0 < \sqrt{t}$ using *assms*
 by *auto*
 have *D*: (*F* has-real-derivative $\ln(t + u^2)$) (at *u*) for *u*

```

proof -
  have tu:  $0 < t + u^2$  using assms
    by (smt (verit, ccfv-SIG) not-sum-power2-lt-zero)
  have q1:  $\sqrt{t} * \sqrt{t} = t$  using assms
    by simp
  have q4:  $\sqrt{t} * (2 * \sqrt{t}) = 2 * t$  using q1
    by simp
  have q5:  $\text{inverse} ((t + u^2) / t) = t / (t + u^2)$ 
    by simp
  have q2:  $(u / \sqrt{t})^2 = u^2 / t$  using assms
    by (simp add: power-divide)
  have q3:  $1 + u^2 / t = (t + u^2) / t$  using assms
    by (simp add: field-simps)
  have h1:  $\text{inverse} (1 + (u / \sqrt{t})^2) * \sqrt{t} * (2 * \sqrt{t}) / t =$ 
     $2 * t / (t + u^2)$  using tu assms
    unfolding q2 q3
    using q1
    by (simp add: q4 q5 mult.assoc)
  have h2:  $2 * u * u / (t + u^2) + (2 * t / (t + u^2) - 2) = 0$  using tu
    by (simp add: field-simps power2-eq-square)
  show ?thesis unfolding F-def[abs-def] using tu st assms
    by (intro derivative-eq-intros) (auto, use h1 h2 in presburger)
qed
have Dv:  $\bigwedge u. u \in \{0..c\} \implies$ 
  ( $F$  has-vector-derivative  $\ln(t + u^2)$ ) (at  $u$  within  $\{0..c\}$ )
  using D has-real-derivative-iff-has-vector-derivative has-vector-derivative-at-within
by blast
have (( $\lambda u. \ln(t + u^2)$ ) has-integral ( $F c - F 0$ ))  $\{0..c\}$ 
  using fundamental-theorem-of-calculus[OF - Dv] assms by linarith
hence integral  $\{0..c\} (\lambda u. \ln(t + u^2)) = F c - F 0$  by blast
thus ?thesis unfolding F-def by simp
qed

lemma Ic-zero:
  assumes  $0 < c$ 
  shows integral  $\{0..c\} (\lambda(u::real). \ln(0 + u^2)) = 2 * (c * \ln c - c)$ 
proof -
  define F where  $F u = 2 * (u * \ln u - (u::real))$  for u
  have lim0:  $((\lambda u::real. 2 * (u * \ln u - u)) \longrightarrow 0)$  (at-right 0)
    by real-asymp
  have cont: continuous-on  $\{0..c\}$  F
  proof (rule continuous-on-IccI)
    show  $(F \longrightarrow F 0)$  (at-right 0) using lim0 unfolding F-def
      by simp
    show  $(F \longrightarrow F c)$  (at-left c) using assms unfolding F-def
      by (force intro: tendsto-intros)
  end
  show  $0 < c$  by (rule assms)
  fix x assume  $0 < x < c$ 
  thus  $F -x \rightarrow F x$  unfolding F-def

```

```

    by (fast intro: tendsto-intros)
qed
have Dv:  $\bigwedge x. x \in \{0 < .. < c\} \implies$ 
  (F has-vector-derivative ln (0 + x ^ 2)) (at x)
proof -
  fix x :: real assume x:  $x \in \{0 < .. < c\}$ 
  have x0:  $0 < x$  using x
  by auto
  have (F has-real-derivative 2 * (1 * ln x + x * (1 / x) - 1)) (at x) unfolding
F-def[abs-def]
  using x0
  by (intro derivative-eq-intros) auto
  moreover have  $2 * (1 * \ln x + x * (1 / x) - 1) = \ln (0 + x ^ 2)$  using x0
ln-realpow[of x 2]
  by simp
  ultimately show (F has-vector-derivative ln (0 + x ^ 2)) (at x)
  using has-real-derivative-iff-has-vector-derivative
  by algebra
qed
have (( $\lambda u. \ln (0 + u ^ 2)$ ) has-integral (F c - F 0)) {0..c}
  using fundamental-theorem-of-calculus-interior[OF - cont Dv] assms
  by linarith
hence integral {0..c} ( $\lambda u. \ln (0 + u ^ 2)$ ) = F c - F 0
  by (fast intro: integral-unique)
thus ?thesis unfolding F-def
  by argo
qed

```

lemma Vf-closed:

```

  assumes  $0 < t$ 
  shows Vf  $t = 2 * \pi * \sqrt{t} + (\ln (t + 1) - 2 + 2 * \sqrt{t} * \arctan (1 / \sqrt{t}))$ 
    -  $6 * (3 / 40 * \ln (t + (3 / 40) ^ 2) - 2 * (3 / 40) + 2 * \sqrt{t} * \arctan (3 / 40 / \sqrt{t}))$ 
  unfolding Vf-def using Ic-closed[OF assms, of 1] Ic-closed[OF assms, of 3 / 40] by simp

```

lemma Vf-zero: Vf 0 = - 2 - 6 * (2 * (3 / 40 * ln (3 / 40) - 3 / 40))

```

proof -
  have h1: integral {0..1} ( $\lambda(u::real). \ln (0 + u ^ 2)$ ) = 2 * (1 * ln 1 - 1)
  using Ic-zero by simp
  have h2: integral {0..3 / 40} ( $\lambda(u::real). \ln (0 + u ^ 2)$ ) =
    2 * (3 / 40 * ln (3 / 40) - 3 / 40) by (rule Ic-zero) simp
  show ?thesis unfolding Vf-def h1 h2 by force
qed

```

(6.8) on $[2, \infty)$, by hand: $2U^\rho(t) - V(t) \leq (13/10) \log t + 6\alpha^3/t -$

$2\pi\sqrt{t}$.

lemma *Uk-le-log*: $k < 16 \implies 2 \leq t \implies Uk\ k\ t \leq \ln\ t$

proof –

assume k : $k < 16$ **and** t : $2 \leq t$

define a **where** $a = tA\ (table1\ !\ k)$

define b **where** $b = tB\ (table1\ !\ k)$

have ab : $0 < a < b < 2$ **using** $table1\text{-}nth[OF\ k]$ **unfolding** $a\text{-}def\ b\text{-}def$ **by** *force*+

have nb : $\neg (a \leq t \wedge t \leq b)$ **using** $ab\ t$ **by** *linarith*

have U : $Uk\ k\ t = \ln\ ((|t - (a + b)| / 2) + \sqrt{(t - a) * (t - b)}) / 2$

using $Uk\text{-}eq[OF\ k,\ of\ t]\ nb$ **unfolding** $a\text{-}def\ b\text{-}def$ **by** *presburger*

have sq : $\sqrt{(t - a) * (t - b)} \leq t$

proof –

have $(t - a) * (t - b) \leq t * t$ **using** $ab\ t$ **by** (*simp add: mult-mono*)

hence $\sqrt{(t - a) * (t - b)} \leq \sqrt{t * t}$ **using** *real-sqrt-le-mono* **by** *presburger*

thus *?thesis* **using** t **by** *auto*

qed

have m : $|t - (a + b)| / 2 = t - (a + b) / 2$ **using** $ab\ t$ **by** *force*

have $p8$: $0 \leq (t - a) * (t - b)$ **using** $ab\ t$

by *simp*

have pos : $0 < (|t - (a + b)| / 2) + \sqrt{(t - a) * (t - b)} / 2$

using $ab\ t\ m$ *real-sqrt-ge-zero* $[OF\ p8]$

by *argo*

have le : $(|t - (a + b)| / 2) + \sqrt{(t - a) * (t - b)} / 2 \leq t$ **using** $ab\ t\ m\ sq$

by *argo*

show *?thesis* **unfolding** U **using** $pos\ le\ t$ *ln-le-cancel-iff* **by** *simp*

qed

lemma *sum-cK*: $(\sum\ k < 16.\ cK\ k) = \text{lam-r}$

proof –

have l : $\text{length}\ table1 = 16$ **by** (*simp add: table1-def*)

have $(\sum\ e \leftarrow table1.\ tC\ e) = (\sum\ k < \text{length}\ table1.\ tC\ (table1\ !\ k))$

using *sum-list-sum-nth* $[of\ map\ tC\ table1]$ **by** (*simp add: atLeast0LessThan*)

thus *?thesis* **using** $table1\text{-}sum\text{-}tC\ l$ **unfolding** $cK\text{-}def$ **by** *simp*

qed

lemma *logPot-le-of-two-le*: $2 \leq t \implies \text{logPot}\ \rho\ t \leq \text{lam-r} * \ln\ t$

proof –

assume t : $2 \leq t$

have $\text{logPot}\ \rho\ t = (\sum\ k < 16.\ cK\ k * Uk\ k\ t)$ **by** (*rule logPot-rho*)

also have $\dots \leq (\sum\ k < 16.\ cK\ k * \ln\ t)$

proof (*rule sum-mono*)

fix k **assume** $(k::nat) \in \{..<16\}$

hence k : $k < 16$ **by** *blast*

have c : $0 \leq cK\ k$ **using** $table1\text{-}nth[OF\ k]$ **by** *satx*

show $cK\ k * Uk\ k\ t \leq cK\ k * \ln\ t$ **using** *mult-left-mono* $[OF\ Uk\text{-}le\text{-}log[OF\ k\ t]]$

c] **by** *satx*

qed

also have ... = lam-r * ln t using sum-cK sum-distrib-right[of cK {.. 16 } ln t]
 by presburger
 finally show ?thesis .
 qed

lemma Ic1-ge: $0 < t \implies \ln t \leq \int_{0..1} (\lambda(u::\text{real}). \ln (t + u^2))$
proof –
 assume t: $0 < t$
 have tq: $t + u^2 \neq 0$ for u using add-pos-nonneg[OF t, of u^2]
 by simp
 have cont: continuous-on $\{0..1\}$ $(\lambda u. \ln (t + u^2))$
 by (force simp: tq intro: continuous-intros)
 have i1: $(\lambda u. \ln (t + u^2))$ integrable-on $\{0..1\}$
 using integrable-continuous-interval cont by blast
 have i0: $(\lambda u. \ln t)$ integrable-on $\{0..(1::\text{real})\}$ by blast
 have pw: $\bigwedge u. u \in \{0..1\} \implies \ln t \leq \ln (t + u^2)$ using t
 using less-eq-real-def ln-mono by auto
 have integral $\{0..1\}$ $(\lambda(u::\text{real}). \ln t) \leq \int_{0..1} (\lambda u. \ln (t + u^2))$
 using integral-le[OF i0 i1 pw] by presburger
 thus ?thesis by fastforce
 qed

lemma Icalpha-le: $0 < (t::\text{real}) \implies \int_{0..3/40} (\lambda u. \ln (t + u^2)) \leq 3/40 * \ln (t + (3/40)^2)$
proof –
 assume t: $0 < t$
 have tq: $t + u^2 \neq 0$ for u using add-pos-nonneg[OF t, of u^2]
 by simp
 have cont: continuous-on $\{0..3/40\}$ $(\lambda u. \ln (t + u^2))$
 using tq by (fast intro: continuous-intros)
 have i1: $(\lambda u. \ln (t + u^2))$ integrable-on $\{0..3/40\}$
 using integrable-continuous-interval[OF cont]
 by fast
 have i0: $(\lambda u. \ln (t + (3/40)^2))$ integrable-on $\{0..(3/40::\text{real})\}$
 by blast
 have pw: $\bigwedge u. u \in \{0..3/40\} \implies \ln (t + u^2) \leq \ln (t + (3/40)^2)$
proof –
 fix u :: real assume u: $u \in \{0..3/40\}$
 have $u^2 \leq (3/40)^2$ using u power-mono[of u 3/40 2] by presburger
 moreover have $0 < t + u^2$ using t by (simp add: add-pos-nonneg)
 ultimately show $\ln (t + u^2) \leq \ln (t + (3/40)^2)$ using ln-le-cancel-iff
 by simp
 qed
 have integral $\{0..3/40\}$ $(\lambda u. \ln (t + u^2)) \leq \int_{0..3/40} (\lambda(u::\text{real}). \ln (t + (3/40)^2))$


```

    using integral-le[OF i1 i0 pw] by presburger
    thus ?thesis by auto
qed

(6.8) on  $[2, \infty)$ , by hand.

lemma tail-bound:  $2 \leq t \implies 2 * \logPot \rho t - \forall f t < - 6645002 / 10 ^ 6$ 
proof -
  assume t:  $2 \leq t$ 
  have t0:  $0 < t$  using t by simp
  have logPot rho t  $\leq \text{lam-r} * \ln t$ 
    by (rule logPot-le-of-two-le[OF t])
  moreover have  $2 * \pi * \sqrt{t} + \ln t - 6 * (3 / 40 * \ln (t + (3 / 40) ^ 2))$ 
 $\leq \forall f t$ 
    unfolding Vf-def using Ic1-ge[OF t0] Icalpha-le[OF t0] by linarith
  moreover have  $2 * \text{lam-r} * \ln t - (2 * \pi * \sqrt{t} + \ln t - 6 * (3 / 40 * \ln$ 
 $(t + (3 / 40) ^ 2))) < - 6645002 / 10 ^ 6$ 
  proof -
    define s where  $s = \sqrt{t}$ 
    have s0:  $0 < s$  using t0 unfolding s-def
      by simp
    have s2:  $\sqrt{2} \leq s$  unfolding s-def using t real-sqrt-le-mono
      by blast
    have r2:  $1.4 \leq \sqrt{2::\text{real}}$  using real-le-rsqrt[of 1.4 2]
      by (simp add: power2-eq-square)
    have lt:  $\ln t \leq 2 * (s - 1)$ 
  proof -
    have  $\ln t = 2 * \ln s$  unfolding s-def using ln-sqrt[of t] t0
      by linarith
    moreover have  $\ln s \leq s - 1$  using ln-le-minus-one[OF s0] .
    ultimately show ?thesis
      by simp
  qed
  have la:  $\ln (t + (3 / 40) ^ 2) \leq \ln t + (3 / 40) ^ 2 / 2$ 
  proof -
    have m6:  $t * (1 + (3 / 40) ^ 2 / t) = t + (3 / 40) ^ 2$  using t0
      by (simp add: distrib-left)
    have p6:  $1 + (3 / 40) ^ 2 / t \neq 0$  using add-pos-nonneg[of 1 (3 / 40) ^ 2 / t] t0
      by simp
    have e:  $\ln (t + (3 / 40) ^ 2) = \ln t + \ln (1 + (3 / 40) ^ 2 / t)$ 
      using t0 ln-mult[of t 1 + (3 / 40) ^ 2 / t] m6 p6
      by simp
    have ln (1 + (3 / 40) ^ 2 / t)  $\leq (3 / 40) ^ 2 / t$ 
      using ln-le-minus-one[of 1 + (3 / 40) ^ 2 / t] t0
      using add-pos-nonneg[of 1 (3 / 40) ^ 2 / t] t0
      by simp
    moreover have  $(3 / 40) ^ 2 / t \leq (3 / 40) ^ 2 / 2$  using t
      by (fastforce intro: divide-left-mono)

```

```

ultimately show ?thesis using e
  by linarith
qed
have pis:  $3 * s \leq \pi * s$  using pi-gt3 s0
  by simp
have c2:  $(3 / 40 :: \text{real})^2 = 9 / 1600$ 
  by (simp add: power2-eq-square)
have key:  $2 * (37 / 40) * \ln t - (2 * \pi * s + \ln t - 6 * (3 / 40 * \ln (t +$ 
 $(3 /$ 
 $40)^2))) \leq -\gamma$ 
  using lt la pis s2 r2 c2
  by argo
show ?thesis unfolding lam-r-def s-def[symmetric] using lt la pis key
  by simp
qed
ultimately show ?thesis by linarith
qed

```

28.4 Ingredients of (6.9) that live here

The circle average of $\log|z - a|$ (*circ-log*) is proved in the circle-average section above (a second copy of the statement here was a duplicate fact declaration).

The mass bound for ϱ : $\varrho([x - r, x + r]) \leq 40\sqrt{r}$ (from the density).

lemma *table1-gap*: $\forall e \in \text{set table1}. 1 / 225 < tB\ e - tA\ e$
 by (simp add: table1-def tA-def tB-def)

lemma *rho-Icc-aux*:

```

assumes  $\forall e \in \text{set } xs. 0 \leq tC\ e \wedge tA\ e < tB\ e \wedge$ 
 $1 / 225 < tB\ e - tA\ e$  and  $r: 0 \leq r$ 
shows  $(\int^+ t. \text{ennreal} (\sum e \leftarrow xs. tC\ e * \text{arcsine-dens} (tA\ e) (tB\ e)\ t) * \text{indicator} \{x - r..x + r\}\ t\ \partial \text{lborel})$ 
 $\leq \text{ennreal} ((\sum e \leftarrow xs. tC\ e) * (40 * \text{sqrt } r))$ 
using assms(1)
proof (induction xs)
  case Nil
  thus ?case by simp
next
  case (Cons e xs)
  have c:  $0 \leq tC\ e \wedge tA\ e < tB\ e \wedge 1 / 225 < tB\ e - tA\ e$  using Cons.prem by
  auto
  have r0:  $0 \leq (\sum e \leftarrow xs. tC\ e * \text{arcsine-dens} (tA\ e) (tB\ e)\ t)$  for t
  using Cons.prem by (intro sum-list-nonneg) (auto intro!: mult-nonneg-nonneg
    arcsine-dens-nonneg)
  have s0:  $0 \leq (\sum e \leftarrow xs. tC\ e)$  using Cons.prem
  by (intro sum-list-nonneg) auto
  have md:  $(\lambda t. \text{ennreal} (\text{arcsine-dens} (tA\ e) (tB\ e)\ t)) \in \text{borel-measurable lborel}$ 

```

```

unfolding arcsine-dens-def by measurable
have ( $\int^+ t. \text{ennreal} (\sum e' \leftarrow e \# xs. tC e' * \text{arcsine-dens} (tA e') (tB e') t) * \text{indicator} \{x - r..x + r\} t \partial \text{lborel}$ )
  = ( $\int^+ t. \text{ennreal} (tC e) * (\text{ennreal} (\text{arcsine-dens} (tA e) (tB e) t) * \text{indicator} \{x - r..x + r\} t)$ 
    +  $\text{ennreal} (\sum e \leftarrow xs. tC e * \text{arcsine-dens} (tA e) (tB e) t) * \text{indicator} \{x - r..x + r\} t \partial \text{lborel}$ )
using c r0
by (force simp: ennreal-plus[symmetric] ennreal-mult arcsine-dens-nonneg distrib-right
  mult.assoc intro: nn-integral-cong)
also have ... =  $\text{ennreal} (tC e) * (\int^+ t. \text{ennreal} (\text{arcsine-dens} (tA e) (tB e) t) * \text{indicator} \{x - r..x + r\} t \partial \text{lborel})$ 
  + ( $\int^+ t. \text{ennreal} (\sum e \leftarrow xs. tC e * \text{arcsine-dens} (tA e) (tB e) t) * \text{indicator} \{x - r..x + r\} t \partial \text{lborel}$ )
using md
by (force simp: nn-integral-add nn-integral-cmult)
also have ...  $\leq \text{ennreal} (tC e) * \text{ennreal} (40 * \text{sqrt } r) + \text{ennreal} ((\sum e \leftarrow xs. tC e) * (40 * \text{sqrt } r))$ 
proof -
  have ( $\int^+ t. \text{ennreal} (\text{arcsine-dens} (tA e) (tB e) t) * \text{indicator} \{x - r..x + r\} t \partial \text{lborel}$ ) =  $\text{emeasure} (\text{arcsine} (tA e) (tB e)) \{x - r..x + r\}$ 
  unfolding arcsine-def using md by (simp add: emeasure-density)
  also have ...  $\leq \text{ennreal} (40 * \text{sqrt } r)$  by (rule arcsine-Icc-le40[OF c(2) c(3) r])
finally show ?thesis using Cons.IH Cons.premis
  by (force intro: add-mono mult-left-mono)
qed
also have ... =  $\text{ennreal} ((\sum e' \leftarrow e \# xs. tC e') * (40 * \text{sqrt } r))$ 
using c s0 r
using ennreal-mult[of tC e 40 * sqrt r] ennreal-plus[of tC e * (40 * sqrt r) ( $\sum e \leftarrow xs. tC e$ ) * (40 * sqrt r)] real-sqrt-ge-zero[OF r]
by (auto simp: distrib-right)
finally show ?case .
qed

lemma rho-Icc-le:  $0 \leq r \implies \text{emeasure } \rho \{x - r..x + r\} \leq \text{ennreal} (40 * \text{sqrt } r)$ 
proof -
  assume r:  $0 \leq r$ 
  have pr:  $\forall e \in \text{set } \text{table1}. 0 \leq tC e \wedge tA e < tB e \wedge 1 / 225 < tB e - tA e$  using table1-props table1-gap by auto
  have e:  $\text{emeasure } \rho \{x - r..x + r\} = (\int^+ t. \text{ennreal} (\sum e \leftarrow \text{table1}. tC e * \text{arcsine-dens} (tA e) (tB e) t) * \text{indicator} \{x - r..x + r\} t \partial \text{lborel})$ 
  unfolding rho-def by (subst emeasure-density) auto
  have l:  $\text{lam-}r * (40 * \text{sqrt } r) \leq 40 * \text{sqrt } r$  using real-sqrt-ge-zero[OF r]
  unfolding lam-r-def by simp
  show ?thesis unfolding e using rho-Icc-aux[OF pr r, of x] table1-sum-tC l

```

by (force intro: order-trans ennreal-leI)
qed

(6.7): $2U^{\wedge}Q(t) - V(t) + \sqrt{t}/K \leq M-0 + \sqrt{2}/K$ for $t \geq 0$, $K \geq 40$.

lemma *V-lower*: $2 \leq t \implies$
 $2 * \pi * \text{sqrt } t + (1 - 6 * \text{alpha-r}) * \ln t - 2 * \text{alpha-r}^{\wedge} 3 / t \leq \forall f t$

proof –
 assume $t: 2 \leq t$
 have $t0: 0 < t$ using t
 by simp
 have $I1: \ln t \leq \text{integral } \{0..1\} (\lambda u. \ln (t + u^{\wedge} 2))$ by (rule *Ic1-ge*[*OF* $t0$])
 have $tp: 0 < t + u^{\wedge} 2$ for u using *add-pos-nonneg*[*OF* $t0$, *of* $u^{\wedge} 2$]
 by simp
 have $tq: t + u^{\wedge} 2 \neq 0$ for u using tp [*of* u]
 by simp
 have $cont: \text{continuous-on } \{0..3 / 40\} (\lambda u. \ln (t + u^{\wedge} 2))$
 using tq by (fastforce intro: *continuous-intros*)
 have $i1: (\lambda u. \ln (t + u^{\wedge} 2)) \text{ integrable-on } \{0..3 / 40\}$
 using *integrable-continuous-interval*[*OF* $cont$]
 by meson
 have $cont2: \text{continuous-on } \{0..3 / 40\} (\lambda u. \ln t + u^{\wedge} 2 / t)$
 using $t0$
 by (fastforce intro: *continuous-intros*)
 have $i2: (\lambda u. \ln t + u^{\wedge} 2 / t) \text{ integrable-on } \{0..3 / 40\}$
 using *integrable-continuous-interval*[*OF* $cont2$]
 by meson
 have $pw: \bigwedge u. u \in \{0..3 / 40\} \implies \ln (t + u^{\wedge} 2) \leq \ln t + u^{\wedge} 2 / t$
proof –
 fix $u :: \text{real}$
 have $m6: t * (1 + u^{\wedge} 2 / t) = t + u^{\wedge} 2$ using $t0$
 by (simp add: *field-simps*)
 have $p6: 1 + u^{\wedge} 2 / t \neq 0$ using *add-pos-nonneg*[*of* $1 + u^{\wedge} 2 / t$] $t0$
 by simp
 have $e: \ln (t + u^{\wedge} 2) = \ln t + \ln (1 + u^{\wedge} 2 / t)$
 using $t0$ *ln-mult*[*of* t $1 + u^{\wedge} 2 / t$] $m6$ $p6$
 by simp
 have $\ln (1 + u^{\wedge} 2 / t) \leq u^{\wedge} 2 / t$ using *ln-le-minus-one*[*of* $1 + u^{\wedge} 2 / t$]
 $t0$
 using *add-pos-nonneg*[*of* $1 + u^{\wedge} 2 / t$] $t0$
 by simp
 thus $\ln (t + u^{\wedge} 2) \leq \ln t + u^{\wedge} 2 / t$ using e
 by simp
 qed
 have $Ia: \text{integral } \{0..3 / 40\} (\lambda u. \ln (t + u^{\wedge} 2)) \leq$
 $\text{integral } \{0..3 / 40\} (\lambda u. \ln t + u^{\wedge} 2 / t)$ by (rule *integral-le*[*OF* $i1$ $i2$ pw])
 have $Ib: \text{integral } \{0..3 / 40\} (\lambda u. \ln t + u^{\wedge} 2 / t) =$
 $3 / 40 * \ln t + (3 / 40)^{\wedge} 3 / (3 * t)$
proof –
 define G where $G u = u * \ln t + u^{\wedge} 3 / (3 * t)$ for u

```

have Dv:  $\bigwedge u. u \in \{0..3 / 40\} \implies$ 
  ( $G$  has-vector-derivative ( $\ln t + u^2 / t$ )) (at  $u$  within  $\{0..3 / 40\}$ )
proof -
  fix u :: real
  have ( $G$  has-real-derivative ( $\ln t + u^2 / t$ )) (at  $u$ ) unfolding  $G$ -def[abs-def]
using t0
  by (intro derivative-eq-intros) auto
  thus ( $G$  has-vector-derivative ( $\ln t + u^2 / t$ )) (at  $u$  within  $\{0..3 / 40\}$ )
  using has-real-derivative-iff-has-vector-derivative has-vector-derivative-at-within
  by blast
qed
have (( $\lambda u. \ln t + u^2 / t$ ) has-integral ( $G (3 / 40) - G 0$ ))  $\{0..3 / 40\}$ 
  using fundamental-theorem-of-calculus[OF - Dv]
  by linarith
hence integral  $\{0..3 / 40\} (\lambda u. \ln t + u^2 / t) = G (3 / 40) - G 0$ 
  by (blast intro: integral-unique)
thus ?thesis unfolding  $G$ -def
  by simp
qed
have six:  $6 * ((3 / 40)^3 / (3 * t)) = 2 * (3 / 40)^3 / t$  using t0
  by (simp add: field-simps)
show ?thesis unfolding Vf-def alpha-r-def using I1 Ia Ib six
  by argo
qed

end

theory Zeta5-Numeric-Enclosures
  imports Zeta5-Equilibrium-Measure HOL-Decision-Procs.Approximation
begin

```

29 Numeric enclosures: (A.9), (A.10) and Lemma 6.1

The explicit numeric inequalities of Appendix A: the sixteen-term enclosures (A.10) of the energy of ϱ and the region check (A.9) of the potential inequality on $[0, 2]$, followed by the statements that depend on them (Lemma 6.1, (6.4), (6.7)). Every numeric fact is marked (** numeric **) and is proved by the *approximation* method of HOL-Decision_Procs, whose certified interval arithmetic runs through the *holds* oracle of code evaluation; these 72 facts are the only source of the oracle in the final theorem (*thm-oracles zeta5-irrational* lists only *holds*). The Lean formalisation performs the same checks by kernel computation on 1052 dyadic intervals (Lean theories *Lemma61Check*, *Lemma61Final*).

29.1 Elementary enclosures (A.10)

definition *eterm* :: *nat* $\Rightarrow$ *real* **where**

eterm *j* = (*Spart* (*j* + 1) $\wedge$ 2 – *Spart* *j* $\wedge$ 2) * *ln* ((*tB* (*table1* ! *j*) – *tA* (*table1* ! *j*)) / 4)

lemma *eterm-0*: – 737284788112975 / 1000000000000000000 < *eterm* 0 $\wedge$
eterm 0 < – 737284788112774 / 1000000000000000000

unfolding *eterm-def*

apply (*simp add*: *Spart-def table1-def tA-def tB-def tC-def*)

by (*approximation 80*)

lemma *eterm-1*: – 8523957090042450 / 1000000000000000000 < *eterm* 1 $\wedge$
eterm 1 < – 8523957090042249 / 1000000000000000000

unfolding *eterm-def*

apply (*simp add*: *Spart-def table1-def tA-def tB-def tC-def*)

by (*approximation 80*)

lemma *eterm-2*: – 26925610911599487 / 1000000000000000000 < *eterm* 2 $\wedge$
eterm 2 < – 26925610911599286 / 1000000000000000000

unfolding *eterm-def*

apply (*simp add*: *Spart-def table1-def tA-def tB-def tC-def*)

by (*approximation 80*)

lemma *eterm-3*: – 59722954555535072 / 1000000000000000000 < *eterm* 3 $\wedge$
eterm 3 < – 59722954555534871 / 1000000000000000000

unfolding *eterm-def*

apply (*simp add*: *Spart-def table1-def tA-def tB-def tC-def*)

by (*approximation 80*)

lemma *ln-split-4*: *ln* (65272891657 / 400000000000 :: *real*) =
ln (65272891657 / 62500000000) – 6 * *ln* 2

proof –

have *e*: (65272891657 / 400000000000 :: *real*) = (65272891657 / 62500000000)
/ 2 $\wedge$ 6 **by** *simp*

have *d*: *ln* ((65272891657 / 62500000000) / 2 $\wedge$ 6 :: *real*) =
ln (65272891657 / 62500000000) – *ln* (2 $\wedge$ 6)

using *ln-divide-pos*[*of* 65272891657 / 62500000000 2 $\wedge$ 6]

by *fastforce*

have *p*: *ln* (2 $\wedge$ 6 :: *real*) = 6 * *ln* 2 **using** *ln-realpow*[*of* 2 6]

by *force*

show ?thesis **unfolding** *e d p* ..

qed

lemma *eterm-4*: – 99809230236437933 / 1000000000000000000 < *eterm* 4 $\wedge$
eterm 4 < – 99809230236429932 / 1000000000000000000

unfolding *eterm-def*

apply (*simp add*: *Spart-def table1-def tA-def tB-def tC-def*)

unfolding *ln-split-4*

by (*approximation 60*)

```

lemma eterm-5:  $-145606013686162115 / 10000000000000000 < \text{eterm } 5 \wedge$ 
 $\text{eterm } 5 < -145606013686161914 / 10000000000000000$ 
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
by (approximation 80)

lemma ln-split-6:  $\ln (144041816267 / 400000000000 :: \text{real}) =$ 
 $\ln (144041816267 / 125000000000) - 5 * \ln 2$ 
proof –
have e:  $(144041816267 / 400000000000 :: \text{real}) = (144041816267 / 125000000000)$ 
 $/ 2^5$  by simp
have d:  $\ln ((144041816267 / 125000000000) / 2^5 :: \text{real}) =$ 
 $\ln (144041816267 / 125000000000) - \ln (2^5)$ 
using ln-divide-pos[of  $144041816267 / 125000000000$   $2^5$ ]
by fastforce
have p:  $\ln (2^5 :: \text{real}) = 5 * \ln 2$  using ln-realpow[of  $2$   $5$ ]
by auto
show ?thesis unfolding e d p ..
qed

lemma eterm-6:  $-186983992097036267 / 10000000000000000 < \text{eterm } 6 \wedge$ 
 $\text{eterm } 6 < -186983992097028266 / 10000000000000000$ 
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
unfolding ln-split-6
by (approximation 60)

lemma eterm-7:  $-221100588014681656 / 10000000000000000 < \text{eterm } 7 \wedge$ 
 $\text{eterm } 7 < -221100588014681455 / 10000000000000000$ 
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
by (approximation 80)

lemma ln-split-8:  $\ln (269180899217 / 400000000000 :: \text{real}) =$ 
 $\ln (269180899217 / 250000000000) - 4 * \ln 2$ 
proof –
have e:  $(269180899217 / 400000000000 :: \text{real}) = (269180899217 / 250000000000)$ 
 $/ 2^4$  by simp
have d:  $\ln ((269180899217 / 250000000000) / 2^4 :: \text{real}) =$ 
 $\ln (269180899217 / 250000000000) - \ln (2^4)$ 
using ln-divide-pos[of  $269180899217 / 250000000000$   $2^4$ ]
by fastforce
have p:  $\ln (2^4 :: \text{real}) = 4 * \ln 2$  using ln-realpow[of  $2$   $4$ ]
by force
show ?thesis unfolding e d p ..
qed

lemma eterm-8:  $-241370916063655491 / 10000000000000000 < \text{eterm } 8 \wedge$ 

```

```

eterm 8 < - 241370916063647490 / 1000000000000000000
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
unfolding ln-split-8
by (approximation 60)

lemma eterm-9: - 247972224603869700 / 1000000000000000000 < eterm 9 ∧
eterm 9 < - 247972224603869499 / 1000000000000000000
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
by (approximation 80)

lemma eterm-10: - 237562912254961608 / 1000000000000000000 < eterm 10 ∧
eterm 10 < - 237562912254961407 / 1000000000000000000
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
by (approximation 80)

lemma ln-split-11: ln (128866386789 / 1000000000000 :: real) =
ln (128866386789 / 125000000000) - 3 * ln 2
proof -
have e: (128866386789 / 1000000000000 :: real) = (128866386789 / 125000000000)
/ 2 ^ 3 by simp
have d: ln ((128866386789 / 125000000000) / 2 ^ 3 :: real) =
ln (128866386789 / 125000000000) - ln (2 ^ 3)
using ln-divide-pos[of 128866386789 / 125000000000 2 ^ 3]
by fastforce
have p: ln (2 ^ 3 :: real) = 3 * ln 2 using ln-realpow[of 2 3]
by fastforce
show ?thesis unfolding e d p ..
qed

lemma eterm-11: - 216269698612393279 / 1000000000000000000 < eterm 11 ∧
eterm 11 < - 216269698612385278 / 1000000000000000000
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
unfolding ln-split-11
by (approximation 60)

lemma eterm-12: - 182633164362425959 / 1000000000000000000 < eterm 12 ∧
eterm 12 < - 182633164362425758 / 1000000000000000000
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)
by (approximation 80)

lemma eterm-13: - 138201939148464454 / 1000000000000000000 < eterm 13 ∧
eterm 13 < - 138201939148464253 / 1000000000000000000
unfolding eterm-def
apply (simp add: Spart-def table1-def tA-def tB-def tC-def)

```


by (*approximation 80*)

lemma *eterm-14*: $- 94725877510899352 / 10000000000000000 < \text{eterm } 14 \wedge$
 $\text{eterm } 14 < - 94725877510899151 / 10000000000000000$
unfolding *eterm-def*
apply (*simp add: Spart-def table1-def tA-def tB-def tC-def*)
by (*approximation 80*)

lemma *eterm-15*: $- 18447081210789815 / 10000000000000000 < \text{eterm } 15 \wedge$
 $\text{eterm } 15 < - 18447081210789614 / 10000000000000000$
unfolding *eterm-def*
apply (*simp add: Spart-def table1-def tA-def tB-def tC-def*)
by (*approximation 80*)

lemma *logEnergy-eterm*: $\log\text{Energy } \rho = (\sum j < 16. \text{eterm } j)$
unfolding *logEnergy-explicit eterm-def ..*

theorem *enclosures-A10*:
 $- 2126593445148 / 10^{12} < \log\text{Energy } \rho \wedge \log\text{Energy } \rho < - 2126593445147 / 10^{12} \wedge$
 $2653035990340 / 10^{12} < C\text{star} \wedge C\text{star} < 2653035990341 / 10^{12}$
proof –
have *I*: $- 2126593445148 / 10^{12} < \log\text{Energy } \rho \wedge$
 $\log\text{Energy } \rho < - 2126593445147 / 10^{12}$
proof –
have *lo*: $- 2126593445148 / 1000000000000 < \log\text{Energy } \rho$ **unfolding** *logEnergy-eterm sum16*
using *eterm-0[THEN conjunct1] eterm-1[THEN conjunct1] eterm-2[THEN conjunct1] eterm-3[THEN conjunct1] eterm-4[THEN conjunct1] eterm-5[THEN conjunct1] eterm-6[THEN conjunct1] eterm-7[THEN conjunct1] eterm-8[THEN conjunct1] eterm-9[THEN conjunct1] eterm-10[THEN conjunct1] eterm-11[THEN conjunct1] eterm-12[THEN conjunct1] eterm-13[THEN conjunct1] eterm-14[THEN conjunct1] eterm-15[THEN conjunct1]*
by *linarith*
have *hi*: $\log\text{Energy } \rho < - 2126593445147 / 1000000000000$ **unfolding** *logEnergy-eterm sum16*
using *eterm-0[THEN conjunct2] eterm-1[THEN conjunct2] eterm-2[THEN conjunct2] eterm-3[THEN conjunct2] eterm-4[THEN conjunct2] eterm-5[THEN conjunct2] eterm-6[THEN conjunct2] eterm-7[THEN conjunct2] eterm-8[THEN conjunct2] eterm-9[THEN conjunct2] eterm-10[THEN conjunct2] eterm-11[THEN conjunct2] eterm-12[THEN conjunct2] eterm-13[THEN conjunct2] eterm-14[THEN conjunct2] eterm-15[THEN conjunct2]*
by *linarith*

```

show ?thesis using lo hi
by auto
qed — evaluate the sixteen summands symbolically: simp add: table1-def Spart-def
tA-def tB-def tC-def leaves an expression in ln of explicit rationals; then the numeric
fact by approximation 120 (* numeric *) - Lean proves the same enclosures from
22 terms of the atanh series (log-series-bounds) per logarithm; that route (bounds
by simp on rationals) is the kernel-only alternative if approximation is too slow:
19 logarithms, each via log-series-bounds [of z 22] with  $z = (y - 1)/(y + 1)$ ,  $y =$ 
 $(b-a)/4 \cdot 2^e \in (1, 2)$ , and  $\ln 2$  ( $z = 1/3$ )
have C:  $2653035990340 / 10^{12} < Cstar \wedge Cstar < 2653035990341 / 10^{12}$ 
unfolding Cstar-def lam-r-def alpha-r-def
by (approximation 60) — approximation 80 (* numeric *) on the explicit
expression in ln ( $3/40$ ) and ln ( $37/20$ ); or log-series-bounds with ln ( $3/40$ ) = ln
( $6/5$ ) -  $4 \ln 2$  ( $z = 1/11$ ) and ln ( $37/20$ ) ( $z = 17/57$ )
show ?thesis using I C by simp
qed

```

29.2 (A.9): the region check on $[0, 2]$

(A.9) on $[0, 2]$. Lean checks 1052 dyadic intervals with certified rational bounds for $\sqrt{\cdot}$, *log*, *arctan* (*decide + kernel*). Isabelle route: $2 U^{\hat{}}_{\varrho}(t) - V(t)$ is an explicit expression in *ln*, *sqrt*, *arctan* of *t* on each of the 33 regions cut out by the table endpoints $a-16 < \dots < a-1 < b-1 < \dots < b-16$ (*Uk-eq*, *Vf-closed*); on each region the inequality $\forall t \in [l, r]. \text{expr}(t) \leq Bd$ is proved by *approximation N* (*splitting: t = k*) (* numeric *).

lemma *Uk-max*:

```

assumes k:  $k < 16$ 
shows  $Uk\ k\ t = \ln (\max ((tB\ (table1\ !\ k) - tA\ (table1\ !\ k)) / 4)$ 
 $((|t - (tA\ (table1\ !\ k) + tB\ (table1\ !\ k)) / 2| + \sqrt{\max\ 0\ ((t -$ 
 $tA\ (table1\ !\ k)) * (t - tB\ (table1\ !\ k))}))) / 2))$ 
proof —
define a where  $a = tA\ (table1\ !\ k)$ 
define b where  $b = tB\ (table1\ !\ k)$ 
have ab:  $a < b$  using table1-nth[OF k] unfolding a-def b-def by auto
show ?thesis
proof (cases  $a \leq t \wedge t \leq b$ )
case True
have p:  $(t - a) * (t - b) \leq 0$  using True by (simp add: mult-nonneg-nonpos)
have m:  $|t - (a + b) / 2| / 2 \leq (b - a) / 4$  using True
using abs-le-iff by argo
have e1:  $\max\ 0\ ((t - a) * (t - b)) = 0$  using p by simp
have e2:  $\max ((b - a) / 4) (|t - (a + b) / 2| + \sqrt{0}) / 2 = (b - a) / 4$ 
using m
by simp
have  $Uk\ k\ t = Lk\ k$  using Uk-const[OF k] True unfolding a-def b-def by blast
thus ?thesis unfolding Lk-def a-def[symmetric] b-def[symmetric] e1 e2
by meson

```

```

next
  case False
  have p:  $0 \leq (t - a) * (t - b)$  using False ab
  by (fastforce intro: mult-nonneg-nonneg mult-nonpos-nonpos)
  have far:  $(b - a) / 2 \leq |t - (a + b) / 2|$  using False by argo
  have e1:  $\max 0 ((t - a) * (t - b)) = (t - a) * (t - b)$  using p by simp
  have e2:  $\max ((b - a) / 4) (|t - (a + b) / 2| + \sqrt{(t - a) * (t - b)}) / 2) =$ 
     $(|t - (a + b) / 2| + \sqrt{(t - a) * (t - b)}) / 2$ 
  using far real-sqrt-ge-zero[OF p]
  using max-def by argo
  have Uk k t =  $\ln (|t - (a + b) / 2| + \sqrt{(t - a) * (t - b)}) / 2$ 
  using Uk-eq[OF k, of t] False unfolding a-def b-def by presburger
  thus ?thesis unfolding a-def[symmetric] b-def[symmetric] e1 e2
  by meson
qed
qed

```

definition $Vc :: \text{real} \Rightarrow \text{real}$ **where**

$$Vc\ t = 2 * \pi * \sqrt{t} + (\ln(t + 1) - 2 + 2 * \sqrt{t} * (\pi / 2 - \arctan(\sqrt{t})))$$

$$- 6 * (3 / 40 * \ln(t + (3 / 40) ^ 2) - 2 * (3 / 40) + 2 * \sqrt{t} * (\pi / 2 - \arctan(\sqrt{t} / (3 / 40))))$$

lemma $Vf \vdash Vc$: **assumes** $0 \leq t$ **shows** $Vf\ t = Vc\ t$

proof (cases $t = 0$)

case True

have l: $\ln((3 / 40 :: \text{real}) ^ 2) = 2 * \ln(3 / 40)$ by (simp add: ln-realpow)

show ?thesis unfolding True Vf-zero Vc-def using l

by (simp add: algebra-simps)

next

case False

hence t0: $0 < t$ using assms by simp

have s0: $0 < \sqrt{t}$ using t0 by simp

have a1: $\arctan(1 / \sqrt{t}) = \pi / 2 - \arctan(\sqrt{t})$ using arctan-inverse[of $\sqrt{t}$] s0

by (simp add: inverse-eq-divide)

have pos: $0 < \sqrt{t} / (3 / 40)$ using s0 by simp

have inv: $\text{inverse}(\sqrt{t} / (3 / 40)) = 3 / 40 / \sqrt{t}$

by simp

have a2: $\arctan(3 / 40 / \sqrt{t}) = \pi / 2 - \arctan(\sqrt{t} / (3 / 40))$

using arctan-inverse[of $\sqrt{t} / (3 / 40)$] pos unfolding inv

by fast

show ?thesis unfolding Vf-closed[OF t0] Vc-def a1 a2 ..

qed

definition $Eb :: \text{real} \Rightarrow \text{real}$ **where**

$$Eb\ t = 2 * (10515596180 / 1000000000000 * \ln(\max(5085947445 / 4000000000000))$$

$$\begin{aligned}
& ((|t - 12899443617 / 200000000000| + \sqrt{\max(0, (t - 3906748086 / 100000000000))} \\
& * (t - 8992695531 / 100000000000))) / 2)) + \\
& 29471737793 / 100000000000 * \ln(\max(13028749591 / 400000000000)) \\
& ((|t - 17653246119 / 200000000000| + \sqrt{\max(0, (t - 2312248264 / 100000000000))} * (t - \\
& 15340997855 / 100000000000))) / 2)) + \\
& 42934365099 / 100000000000 * \ln(\max(24327894059 / 400000000000)) \\
& ((|t - 27132467389 / 200000000000| + \sqrt{\max(0, (t - 1402286665 / 100000000000))} * (t - \\
& 25730180724 / 100000000000))) / 2)) + \\
& 58204231966 / 100000000000 * \ln(\max(41027852890 / 400000000000)) \\
& ((|t - 42791303602 / 200000000000| + \sqrt{\max(0, (t - 881725356 / 100000000000))} * (t - \\
& 41909578246 / 100000000000))) / 2)) + \\
& 69037621310 / 100000000000 * \ln(\max(65272891657 / 400000000000)) \\
& ((|t - 66429287469 / 200000000000| + \sqrt{\max(0, (t - 578197906 / 100000000000))} * (t - \\
& 65851089563 / 100000000000))) / 2)) + \\
& 78873099189 / 100000000000 * \ln(\max(99084713271 / 400000000000)) \\
& ((|t - 99877362497 / 200000000000| + \sqrt{\max(0, (t - 396324613 / 100000000000))} * (t - \\
& 99481037884 / 100000000000))) / 2)) + \\
& 84856120711 / 100000000000 * \ln(\max(144041816267 / 400000000000)) \\
& ((|t - 144609638649 / 200000000000| + \sqrt{\max(0, (t - 283911191 / 100000000000))} * (t - \\
& 144325727458 / 100000000000))) / 2)) + \\
& 88396082127 / 100000000000 * \ln(\max(200893556541 / 400000000000)) \\
& ((|t - 201317968917 / 200000000000| + \sqrt{\max(0, (t - 212206188 / 100000000000))} * (t - \\
& 201105762729 / 100000000000))) / 2)) + \\
& 88303382125 / 100000000000 * \ln(\max(269180899217 / 400000000000)) \\
& ((|t - 269511094589 / 200000000000| + \sqrt{\max(0, (t - 165097686 / 100000000000))} * (t - \\
& 269345996903 / 100000000000))) / 2)) + \\
& 85472321255 / 100000000000 * \ln(\max(346956207024 / 400000000000)) \\
& ((|t - 347222901288 / 200000000000| + \sqrt{\max(0, (t - 133347132 / 100000000000))} * (t - \\
& 347089554156 / 100000000000))) / 2)) +
\end{aligned}$$

$$\begin{aligned}
& 78899184238 / 100000000000 * \ln (\max (430695182301 / 400000000000) \\
& ((|t - \\
& \quad 430918226529 / 200000000000| + \text{sqrt} (\max 0 ((t - 111522114 / \\
& 100000000000) * (t - \\
& \quad 430806704415 / 100000000000)))) / 2)) + \\
& 70353471918 / 100000000000 * \ln (\max (515465547156 / 400000000000) \\
& ((|t - \\
& \quad 515658245866 / 200000000000| + \text{sqrt} (\max 0 ((t - 96349355 / 100000000000) \\
& * (t - \\
& \quad 515561896511 / 100000000000)))) / 2)) + \\
& 58838976615 / 100000000000 * \ln (\max (595362962907 / 400000000000) \\
& ((|t - \\
& \quad 595534594185 / 200000000000| + \text{sqrt} (\max 0 ((t - 85815639 / 100000000000) \\
& * (t - \\
& \quad 595448778546 / 100000000000)))) / 2)) + \\
& 44421321106 / 100000000000 * \ln (\max (664162715772 / 400000000000) \\
& ((|t - \\
& \quad 664320051194 / 200000000000| + \text{sqrt} (\max 0 ((t - 78667711 / 100000000000) \\
& * (t - \\
& \quad 664241383483 / 100000000000)))) / 2)) + \\
& 30462865791 / 100000000000 * \ln (\max (716086447547 / 400000000000) \\
& ((|t - \\
& \quad 716234706677 / 200000000000| + \text{sqrt} (\max 0 ((t - 74129565 / 100000000000) \\
& * (t - \\
& \quad 716160577112 / 100000000000)))) / 2)) + \\
& 5959622577 / 100000000000 * \ln (\max (746565554359 / 400000000000) \\
& ((|t - 746709036979 / \\
& \quad 200000000000| + \text{sqrt} (\max 0 ((t - 71741310 / 100000000000) * (t - \\
& 746637295669 / \\
& \quad 100000000000)))) / 2))) - Vc t
\end{aligned}$$

lemma *Eb-eq*: **assumes** $0 \leq t$ **shows** $2 * \log \text{Pot} \text{ rho } t - V f t = E b t$
by (*simp add: logPot-rho Vf-Vc[OF assms] sum16 Uk-max cK-def table1-def*
tA-def tB-def tC-def
Eb-def)

lemma *Eb-R29*: $595448778546 / 100000000000 \leq t \wedge$
 $t \leq 664241383483 / 100000000000 \implies E b t \leq - 66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-R28*: $515561896511 / 100000000000 \leq t \wedge$
 $t \leq 595448778546 / 100000000000 \implies E b t \leq - 66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-R30*: $664241383483 / 100000000000 \leq t \wedge$
 $t \leq 716160577112 / 100000000000 \implies E b t \leq - 66451 / 10^4$
unfolding *Eb-def Vc-def*

by (approximation 30 splitting: $t = 9$)

lemma Eb-R23: $144325727458 / 1000000000000 \leq t \wedge$
 $t \leq 201105762729 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R24: $201105762729 / 1000000000000 \leq t \wedge$
 $t \leq 269345996903 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R25: $269345996903 / 1000000000000 \leq t \wedge$
 $t \leq 347089554156 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R26: $347089554156 / 1000000000000 \leq t \wedge$
 $t \leq 430806704415 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R27: $430806704415 / 1000000000000 \leq t \wedge$
 $t \leq 515561896511 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R31: $716160577112 / 1000000000000 \leq t \wedge$
 $t \leq 746637295669 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R16: $3906748086 / 1000000000000 \leq t \wedge$
 $t \leq 8992695531 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R18: $15340997855 / 1000000000000 \leq t \wedge$
 $t \leq 25730180724 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R19: $25730180724 / 1000000000000 \leq t \wedge$
 $t \leq 41909578246 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
 unfolding Eb-def Vc-def
 by (approximation 30 splitting: $t = 9$)

lemma Eb-R20: $41909578246 / 1000000000000 \leq t \wedge$
 $t \leq 65851089563 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$

unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-R21*: $65851089563 / 1000000000000 \leq t \wedge$
 $t \leq 99481037884 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-R22*: $99481037884 / 1000000000000 \leq t \wedge$
 $t \leq 144325727458 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-Q0*: $8992695531 / 1000000000000 \leq t \wedge$
 $t \leq 9786233321 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 7*)

lemma *Eb-Q2*: $10579771112 / 1000000000000 \leq t \wedge$
 $t \leq 11373308902 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 7*)

lemma *Eb-Q3*: $11373308902 / 1000000000000 \leq t \wedge$
 $t \leq 12166846693 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 7*)

lemma *Eb-P0*: $9786233321 / 1000000000000 \leq t \wedge$
 $t \leq 9984617768 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 6*)

lemma *Eb-P2*: $10183002216 / 1000000000000 \leq t \wedge$
 $t \leq 10381386664 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 6*)

lemma *Eb-P3*: $10381386664 / 1000000000000 \leq t \wedge$
 $t \leq 10579771112 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 6*)

lemma *ln-shift64*: **assumes** $0 \leq t$ **shows** $\ln(t + (3 / 40)^2) =$
 $\ln(64 * (t + (3 / 40)^2)) - 6 * \ln(2 :: real)$
proof –
have $p: 0 < t + (3 / 40 :: real)^2$ **using** *assms* **by** (*simp add: add-nonneg-pos*)
have $\ln(64 * (t + (3 / 40)^2)) = \ln 64 + \ln(t + (3 / 40 :: real)^2)$
using *ln-mult[of 64 t + (3 / 40)^2] p*

```

    by force
    moreover have  $\ln(64 :: \text{real}) = 6 * \ln 2$  using ln-realpow[of 2 6]
    by auto
    ultimately show ?thesis by simp
qed

lemma Eb-P1:  $9984617768 / 1000000000000 \leq t \wedge$ 
 $t \leq 10183002216 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$ 
proof -
  assume h:  $9984617768 / 1000000000000 \leq t \wedge t \leq 10183002216 / 1000000000000$ 
  have e:  $\ln(t + (3 / 40)^2) = \ln(64 * (t + (3 / 40)^2)) - 6 * \ln(2 :: \text{real})$  using h
  by (intro ln-shift64) simp
  show ?thesis unfolding Eb-def Vc-def e using h
  by (approximation 30 splitting:  $t = 6$ )
qed

lemma Eb-T17b:  $12166846693 / 1000000000000 \leq t \wedge$ 
 $t \leq 15340997855 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$ 
unfolding Eb-def Vc-def
by (approximation 30 splitting:  $t = 9$ )

lemma Eb-R11:  $396324613 / 1000000000000 \leq t \wedge$ 
 $t \leq 578197906 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$ 
unfolding Eb-def Vc-def
by (approximation 30 splitting:  $t = 5$ )

lemma Eb-R12:  $578197906 / 1000000000000 \leq t \wedge$ 
 $t \leq 881725356 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$ 
unfolding Eb-def Vc-def
by (approximation 30 splitting:  $t = 5$ )

lemma Eb-R13:  $881725356 / 1000000000000 \leq t \wedge$ 
 $t \leq 1402286665 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$ 
unfolding Eb-def Vc-def
by (approximation 30 splitting:  $t = 5$ )

lemma Eb-R14:  $1402286665 / 1000000000000 \leq t \wedge$ 
 $t \leq 2312248264 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$ 
unfolding Eb-def Vc-def
by (approximation 30 splitting:  $t = 7$ )

lemma Eb-R15:  $2312248264 / 1000000000000 \leq t \wedge$ 
 $t \leq 3906748086 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$ 
unfolding Eb-def Vc-def
by (approximation 30 splitting:  $t = 6$ )

lemma Eb-R0:  $0 \leq t \wedge t \leq 71741310 / 1000000000000 \implies$ 
 $Eb\ t \leq -66451 / 10^4$ 

```


unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R1*: $71741310 / 100000000000 \leq t \wedge$
 $t \leq 74129565 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R2*: $74129565 / 100000000000 \leq t \wedge$
 $t \leq 78667711 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R3*: $78667711 / 100000000000 \leq t \wedge$
 $t \leq 85815639 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R4*: $85815639 / 100000000000 \leq t \wedge$
 $t \leq 96349355 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R5*: $96349355 / 100000000000 \leq t \wedge$
 $t \leq 111522114 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R6*: $111522114 / 100000000000 \leq t \wedge$
 $t \leq 133347132 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R7*: $133347132 / 100000000000 \leq t \wedge$
 $t \leq 165097686 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R8*: $165097686 / 100000000000 \leq t \wedge$
 $t \leq 212206188 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R9*: $212206188 / 100000000000 \leq t \wedge$
 $t \leq 283911191 / 100000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-R10*: $283911191 / 100000000000 \leq t \wedge$

$t \leq 396324613 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 5*)

lemma *Eb-S0*: $746637295669 / 1000000000000 \leq t \wedge$
 $t \leq 800000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S1*: $800000000000 / 1000000000000 \leq t \wedge$
 $t \leq 900000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S2*: $900000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1000000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S3*: $1000000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1100000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S4*: $1100000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1200000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S5*: $1200000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1300000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S6*: $1300000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1400000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S7*: $1400000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1500000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S8*: $1500000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1600000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting: t = 9*)

lemma *Eb-S9*: $1600000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1700000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting*: $t = 9$)

lemma *Eb-S10*: $1700000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1800000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting*: $t = 9$)

lemma *Eb-S11*: $1800000000000 / 1000000000000 \leq t \wedge$
 $t \leq 1900000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting*: $t = 9$)

lemma *Eb-S12*: $1900000000000 / 1000000000000 \leq t \wedge$
 $t \leq 2000000000000 / 1000000000000 \implies Eb\ t \leq -66451 / 10^4$
unfolding *Eb-def Vc-def*
by (*approximation 30 splitting*: $t = 9$)

lemma *Eb-from-51*: $1900000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S12[of t]* **by** *linarith*

lemma *Eb-from-50*: $1800000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S11[of t]* *Eb-from-51[of t]* **by** *linarith*

lemma *Eb-from-49*: $1700000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S10[of t]* *Eb-from-50[of t]* **by** *linarith*

lemma *Eb-from-48*: $1600000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S9[of t]* *Eb-from-49[of t]* **by** *linarith*

lemma *Eb-from-47*: $1500000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S8[of t]* *Eb-from-48[of t]* **by** *linarith*

lemma *Eb-from-46*: $1400000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S7[of t]* *Eb-from-47[of t]* **by** *linarith*

lemma *Eb-from-45*: $1300000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S6[of t]* *Eb-from-46[of t]* **by** *linarith*

lemma *Eb-from-44*: $1200000000000 / 1000000000000 \leq t \implies$

$t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S5*[of *t*] *Eb-from-45*[of *t*] **by** *linarith*

lemma *Eb-from-43*: $1100000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S4*[of *t*] *Eb-from-44*[of *t*] **by** *linarith*

lemma *Eb-from-42*: $1000000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S3*[of *t*] *Eb-from-43*[of *t*] **by** *linarith*

lemma *Eb-from-41*: $900000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S2*[of *t*] *Eb-from-42*[of *t*] **by** *linarith*

lemma *Eb-from-40*: $800000000000 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S1*[of *t*] *Eb-from-41*[of *t*] **by** *linarith*

lemma *Eb-from-39*: $746637295669 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-S0*[of *t*] *Eb-from-40*[of *t*] **by** *linarith*

lemma *Eb-from-38*: $716160577112 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R31*[of *t*] *Eb-from-39*[of *t*] **by** *linarith*

lemma *Eb-from-37*: $664241383483 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R30*[of *t*] *Eb-from-38*[of *t*] **by** *linarith*

lemma *Eb-from-36*: $595448778546 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R29*[of *t*] *Eb-from-37*[of *t*] **by** *linarith*

lemma *Eb-from-35*: $515561896511 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R28*[of *t*] *Eb-from-36*[of *t*] **by** *linarith*

lemma *Eb-from-34*: $430806704415 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R27*[of *t*] *Eb-from-35*[of *t*] **by** *linarith*

lemma *Eb-from-33*: $347089554156 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R26*[of *t*] *Eb-from-34*[of *t*] **by** *linarith*

lemma *Eb-from-32*: $269345996903 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$

using *Eb-R25*[of *t*] *Eb-from-33*[of *t*] **by** *linarith*

lemma *Eb-from-31*: $201105762729 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R24*[of *t*] *Eb-from-32*[of *t*] **by** *linarith*

lemma *Eb-from-30*: $144325727458 / 1000000000000 \leq t \implies$
 $t \leq 2 \implies Eb\ t \leq -66451 / 10^4$
using *Eb-R23*[of *t*] *Eb-from-31*[of *t*] **by** *linarith*

lemma *Eb-from-29*: $99481037884 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R22*[of *t*] *Eb-from-30*[of *t*] **by** *linarith*

lemma *Eb-from-28*: $65851089563 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R21*[of *t*] *Eb-from-29*[of *t*] **by** *linarith*

lemma *Eb-from-27*: $41909578246 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R20*[of *t*] *Eb-from-28*[of *t*] **by** *linarith*

lemma *Eb-from-26*: $25730180724 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R19*[of *t*] *Eb-from-27*[of *t*] **by** *linarith*

lemma *Eb-from-25*: $15340997855 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R18*[of *t*] *Eb-from-26*[of *t*] **by** *linarith*

lemma *Eb-from-24*: $12166846693 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-T17b*[of *t*] *Eb-from-25*[of *t*] **by** *linarith*

lemma *Eb-from-23*: $11373308902 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-Q3*[of *t*] *Eb-from-24*[of *t*] **by** *linarith*

lemma *Eb-from-22*: $10579771112 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-Q2*[of *t*] *Eb-from-23*[of *t*] **by** *linarith*

lemma *Eb-from-21*: $10381386664 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-P3*[of *t*] *Eb-from-22*[of *t*] **by** *linarith*

lemma *Eb-from-20*: $10183002216 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-P2*[of *t*] *Eb-from-21*[of *t*] **by** *linarith*

lemma *Eb-from-19*: $9984617768 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-P1*[of *t*] *Eb-from-20*[of *t*] **by** *linarith*

lemma *Eb-from-18*: $9786233321 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-P0*[of *t*] *Eb-from-19*[of *t*] **by** *linarith*

lemma *Eb-from-17*: $8992695531 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-Q0*[of *t*] *Eb-from-18*[of *t*] **by** *linarith*

lemma *Eb-from-16*: $3906748086 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R16*[of *t*] *Eb-from-17*[of *t*] **by** *linarith*

lemma *Eb-from-15*: $2312248264 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R15*[of *t*] *Eb-from-16*[of *t*] **by** *linarith*

lemma *Eb-from-14*: $1402286665 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R14*[of *t*] *Eb-from-15*[of *t*] **by** *linarith*

lemma *Eb-from-13*: $881725356 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R13*[of *t*] *Eb-from-14*[of *t*] **by** *linarith*

lemma *Eb-from-12*: $578197906 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R12*[of *t*] *Eb-from-13*[of *t*] **by** *linarith*

lemma *Eb-from-11*: $396324613 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R11*[of *t*] *Eb-from-12*[of *t*] **by** *linarith*

lemma *Eb-from-10*: $283911191 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R10*[of *t*] *Eb-from-11*[of *t*] **by** *linarith*

lemma *Eb-from-9*: $212206188 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R9*[of *t*] *Eb-from-10*[of *t*] **by** *linarith*

lemma *Eb-from-8*: $165097686 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$
using *Eb-R8*[of *t*] *Eb-from-9*[of *t*] **by** *linarith*

lemma *Eb-from-7*: $133347132 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$

using *Eb-R7*[of *t*] *Eb-from-8*[of *t*] **by** *linarith*

lemma *Eb-from-6*: $111522114 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$

using *Eb-R6*[of *t*] *Eb-from-7*[of *t*] **by** *linarith*

lemma *Eb-from-5*: $96349355 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$

using *Eb-R5*[of *t*] *Eb-from-6*[of *t*] **by** *linarith*

lemma *Eb-from-4*: $85815639 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$

using *Eb-R4*[of *t*] *Eb-from-5*[of *t*] **by** *linarith*

lemma *Eb-from-3*: $78667711 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$

using *Eb-R3*[of *t*] *Eb-from-4*[of *t*] **by** *linarith*

lemma *Eb-from-2*: $74129565 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$

using *Eb-R2*[of *t*] *Eb-from-3*[of *t*] **by** *linarith*

lemma *Eb-from-1*: $71741310 / 1000000000000 \leq t \implies t \leq 2 \implies$
 $Eb\ t \leq -66451 / 10^4$

using *Eb-R1*[of *t*] *Eb-from-2*[of *t*] **by** *linarith*

lemma *Eb-from-0*: $0 \leq t \implies t \leq 2 \implies$

$Eb\ t \leq -66451 / 10^4$

using *Eb-R0*[of *t*] *Eb-from-1*[of *t*] **by** *linarith*

lemma *potential-bound-0-2*: $0 \leq t \implies t \leq 2 \implies$

$2 * \logPot\ rho\ t - \forall f\ t \leq -66451 / 10^4$

using *Eb-eq*[of *t*] *Eb-from-0*[of *t*]

by *linarith*

(A.9) with (6.8): $2U^{\varrho}(t) - V(t) < -6645002/10^6 < M-0$ for $t \geq 0$.

theorem *potential-A9*: $0 \leq t \implies 2 * \logPot\ rho\ t - \forall f\ t < -6645002 / 10^6$

proof –

assume *t*: $0 \leq t$

show $2 * \logPot\ rho\ t - \forall f\ t < -6645002 / 10^6$

proof (cases $t \leq 2$)

case *True*

thus ?thesis **using** *potential-bound-0-2*[OF *t True*] **by** *simp*

next

case *False*

thus ?thesis **by** (intro tail-bound) *simp*

qed

qed

(6.4).

theorem *rho-energy-bound*: $\text{lam-r} * \text{of-rat } M0c - \text{logEnergy } \rho + Cstar \leq \text{of-rat } Ubar$

using *enclosures-A10* **by** (*simp add: lam-r-def M0c-def Ubar-def of-rat-divide of-rat-minus*)

Lemma 6.1: ϱ has mass λ , is supported in $(0, 2)$, satisfies (6.2) and (6.4).

theorem *rho-properties*:

$\text{emeasure } \rho \text{ UNIV} = \text{ennreal } \text{lam-r} \wedge \text{emeasure } \rho (-\{0 < \cdot < 2\}) = 0 \wedge$
 $(\forall t. 0 \leq t \longrightarrow 2 * \text{logPot } \rho t - \text{Vf } t \leq \text{of-rat } M0c) \wedge$
 $\text{lam-r} * \text{of-rat } M0c - \text{logEnergy } \rho + Cstar \leq \text{of-rat } Ubar$

proof (*intro conjI allI impI*)

fix $t :: \text{real}$ **assume** $0 \leq t$

hence $2 * \text{logPot } \rho t - \text{Vf } t < -6645002 / 10^6$ **by** (*rule potential-A9*)

thus $2 * \text{logPot } \rho t - \text{Vf } t \leq \text{of-rat } M0c$ **by** (*simp add: M0c-def of-rat-divide of-rat-minus*)

qed (*simp-all add: rho-univ rho-compl rho-energy-bound*)

29.3 (6.9) ingredient

(6.7): $2U^\wedge_\varrho(t) - V(t) + \sqrt{t}/K \leq M-0 + \sqrt{2}/K$ for $t \geq 0, K \geq 40$.

lemma *field-bound*: $40 \leq K \implies 0 \leq t \implies$

$2 * \text{logPot } \rho t - \text{Vf } t + \text{sqrt } t / K \leq \text{of-rat } M0c + \text{sqrt } 2 / K$

proof –

assume $K: 40 \leq K$ **and** $t: 0 \leq t$

show *?thesis*

proof (*cases t ≤ 2*)

case *True*

have $2 * \text{logPot } \rho t - \text{Vf } t \leq \text{of-rat } M0c$ **using** *potential-A9[OF t]*

by (*simp add: M0c-def of-rat-divide of-rat-minus*)

moreover have $\text{sqrt } t / K \leq \text{sqrt } 2 / K$ **using** *True K*

by (*intro divide-right-mono real-sqrt-le-mono*) *auto*

ultimately show *?thesis* **by** *linarith*

next

case *False*

hence $t2: 2 \leq t$ **by** *simp*

have $U: \text{logPot } \rho t \leq \text{lam-r} * \ln t$ **by** (*rule logPot-le-of-two-le[OF t2]*)

have $V: 2 * \pi * \text{sqrt } t + (1 - 6 * \alpha-r) * \ln t - 2 * \alpha-r^3 / t \leq$

$\text{Vf } t$

by (*rule V-lower[OF t2]*)

define s **where** $s = \text{sqrt } t$

have $t0: 0 < t$ **using** $t2$ **by** *simp*

have $s0: 0 < s$ **using** $t0$ **unfolding** $s\text{-def}$ **by** *simp*

have $s2: \text{sqrt } 2 \leq s$ **unfolding** $s\text{-def}$ **using** $t2$ *real-sqrt-le-mono* **by** *blast*

have $r2: 1.4 \leq \text{sqrt } (2::\text{real})$ **using** *real-le-rsqrt[of 1.4 2]*

by (*simp add: power2-eq-square*)


```

have lt:  $\ln t \leq 2 * (s - 1)$ 
proof -
  have  $\ln t = 2 * \ln s$  unfolding s-def using ln-sqrt[of t] t0 by linarith
  moreover have  $\ln s \leq s - 1$  using ln-le-minus-one[OF s0] .
  ultimately show ?thesis by simp
qed
have  $a3: 2 * \alpha\text{-}r^3 / t \leq \alpha\text{-}r^3$  using t2 unfolding alpha-r-def
  by (simp add: field-simps)
have  $sK: s / K \leq s / 40$  using K s0
  by (fastforce intro: divide-left-mono)
have  $K0: 0 \leq \sqrt{2} / K$  using K
  by simp
have  $M: \text{of-rat } M0c = (-1329 / 200 :: \text{real})$  unfolding M0c-def
  by (simp add: of-rat-divide of-rat-minus)
have  $\text{pis}: 3 * s \leq \pi * s$  using pi-gt3 s0
  by simp
have  $c3: (3 / 40 :: \text{real})^3 = 27 / 64000$  by (simp add: power3-eq-cube)
show ?thesis unfolding M s-def[symmetric] using U V lt a3 sK K0 pis s2 r2
  unfolding lam-r-def alpha-r-def s-def[symmetric]
  using c3
  by arg0
qed
qed

```

end

theory *Zeta5-Archimedean-Bound*

imports *Zeta5-Numeric-Enclosures Andreief-Identity*

begin

30 The archimedean bound: Proposition 6.3

Section 6 of the paper: the bound (6.11) on the weight, (6.10) by Andreief's identity, the one-coordinate bound (6.12), the bound (6.9) for every configuration through the logarithmic energy of ϱ and the zero-mass energy inequality, and the estimates (6.14), (6.15) leading to Proposition 6.3 (*log-FK-zeta5-bound*): $\ln F_K(\zeta(5)) \leq \text{\textit{Ubar}} \cdot K^2 + O(K \ln K)$.
 Lean theories *Section6*, *Energy6*, *Section6b*, *Section6c*, *Config69*.

30.1 (6.11) and Lemma 6.2

(6.11): $w(y) \leq 8192(1+y)^5 e^{-2\pi y}$.

lemma *binomial-term-le*:

fixes $t :: \text{real}$
assumes $0 \leq t$
shows $\text{real } (n \text{ choose } k) * t^k \leq (1 + t)^n$
proof (*cases* $k \leq n$)
 case *True*
 have $(t + 1)^n = (\sum i \leq n. \text{real } (n \text{ choose } i) * t^i * 1^{(n-i)})$
 by (*rule binomial-ring*)
 also have $\dots \geq \text{real } (n \text{ choose } k) * t^k * 1^{(n-k)}$
 by (*rule member-le-sum*) (*use assms True in auto*)
 finally show ?thesis **by** (*simp add: add.commute*)
 next
 case *False*
 hence $n < k$ **by** *simp*
 thus ?thesis **using** *assms* **by** (*simp add: binomial-eq-0*)
qed

lemma *pow4-exp-bound*:

fixes $x :: \text{real}$
assumes $0 \leq x$
shows $x^4 * \exp(-x) \leq 42$
proof –
 have $\exp(x / 12) \geq 1 + x / 12$ **by** (*rule exp-ge-add-one-self*)
 hence $(1 + x / 12)^{12} \leq \exp(x / 12)^{12}$ **using** *assms* **by** (*intro power-mono*)
auto
 also have $\exp(x / 12)^{12} = \exp x$ **by** (*simp flip: exp-of-nat-mult*)
 finally have $e: (1 + x / 12)^{12} \leq \exp x$.
 have $\text{real } (12 \text{ choose } 4) * (x / 12)^4 \leq (1 + x / 12)^{12}$
 by (*rule binomial-term-le*) (*use assms in simp*)
 moreover have $(12 \text{ choose } 4) = 495 :: \text{nat}$
 using *binomial-fact-lemma*[*of 4 12*] **by** (*simp add: fact-numeral*)
 ultimately have $495 * (x / 12)^4 \leq \exp x$ **using** e **by** *simp*
 hence $x^4 \leq 20736 / 495 * \exp x$ **by** (*simp add: power-divide field-simps*)
 also have $\dots \leq 42 * \exp x$ **by** *simp*
 finally show ?thesis **by** (*simp add: exp-minus field-simps*)
qed

theorem *weight-le*: $0 < y \implies \text{weight } y \leq 8192 * (1 + y)^5 * \exp(-2 * \pi * y)$

proof –

assume $y: 0 < y$
define Y **where** $Y = \min y 1$
define s **where** $s = \pi * Y$
have $Y0: 0 < Y$ $Y \leq y$ $Y \leq 1$ **using** y **by** (*auto simp: Y-def*)
have $s0: 0 < s$ **and** $spi: s \leq \pi$ **using** $Y0$ **by** (*auto simp: s-def*)
define a **where** $a \ l = \text{real } (l + 1)^4 * \exp(-2 * \pi * \text{real } (l + 1) * y)$ **for** l

```

define b where b l = real (l + 1) ^ 4 * exp (- 2 * pi * real l * y) for l
define E where E = exp (- 2 * pi * y)
have ab: a l = E * b l for l
  by (simp add: a-def b-def E-def algebra-simps flip: exp-add)
have sa: summable a unfolding a-def by (rule weight-summable[OF y])
have ba: b l = inverse E * a l for l using ab[of l] by (simp add: E-def)
have sb: summable b unfolding ba by (rule summable-mult[OF sa])
define q where q = exp (- s)
have q0: 0 < q < 1 using s0 by (auto simp: q-def)
define g where g l = 42 * exp s / s ^ 4 * q ^ l for l
have bg: b l ≤ g l for l
proof -
  define x where x = s * real (l + 1)
  have x0: 0 ≤ x using s0 by (simp add: x-def)
  have Y * real l ≤ y * real l by (rule mult-right-mono) (use Y0 in auto)
  hence exp (- 2 * pi * real l * y) ≤ exp (- 2 * s * real l)
    by (simp add: s-def algebra-simps)
  hence b l ≤ real (l + 1) ^ 4 * exp (- 2 * s * real l) by (simp add: b-def
mult-left-mono)
  also have real (l + 1) ^ 4 * exp (- 2 * s * real l) =
    (x ^ 4 * exp (- x)) * exp s * q ^ l / s ^ 4
  proof -
    have ql: q ^ l = exp (real l * (- s)) unfolding q-def by (rule exp-of-nat-mult[symmetric])
    have exp (- x) * exp s * q ^ l = exp (- x + s + real l * (- s)) unfolding ql
      by (simp add: mult-exp-exp algebra-simps)
    also have - x + s + real l * (- s) = - 2 * s * real l by (simp add: x-def
algebra-simps)
    finally have e: exp (- x) * exp s * q ^ l = exp (- 2 * s * real l) .
    have x4: x ^ 4 = s ^ 4 * real (l + 1) ^ 4 by (simp add: x-def power-mult-distrib)
    have real (l + 1) ^ 4 * exp (- 2 * s * real l) =
      real (l + 1) ^ 4 * (exp (- x) * exp s * q ^ l)
    by (simp only: e)
    also have ... = (x ^ 4 * exp (- x)) * exp s * q ^ l / s ^ 4 unfolding x4
using s0
  by simp
  finally show ?thesis .
qed
also have ... ≤ 42 * exp s * q ^ l / s ^ 4
using pow4-exp-bound[OF x0] s0 q0 by (intro divide-right-mono mult-right-mono)
auto
  finally show ?thesis by (simp add: g-def)
qed
have G: g sums (42 * exp s / s ^ 4 * (1 / (1 - q)))
  unfolding g-def by (rule sums-mult) (rule geometric-sums, use q0 in simp)
have suminf b ≤ suminf g by (rule suminf-le[OF bg sb sums-summable[OF G]])
also have suminf g = 42 * exp s / s ^ 4 * (1 / (1 - q)) using G by (simp
add: sums-iff)
  finally have Sg: suminf b ≤ 42 * exp s / s ^ 4 * (1 / (1 - q)) .
  have inv: 1 / (1 - q) ≤ (1 + s) / s

```

proof –
 have $1 + s \leq \exp s$ **by** (rule exp-ge-add-one-self)
 hence $q \leq 1 / (1 + s)$ **using** $s0$ **by** (simp add: q-def exp-minus field-simps)
 hence $le: s / (1 + s) \leq 1 - q$ **using** $s0$ **by** (simp add: field-simps)
 have $1 / (1 - q) \leq 1 / (s / (1 + s))$
 by (rule divide-left-mono[OF le]) (use $q0\ s0$ in ⟨simp-all add:
 zero-less-mult-iff⟩)
 thus ?thesis **by** simp
qed
 have $42 * \exp s / s^4 * (1 / (1 - q)) \leq 42 * \exp s / s^4 * ((1 + s) / s)$
 using inv $s0$ **by** (intro mult-left-mono) auto
 with Sg have Sb' : $\text{suminf } b \leq 42 * \exp s / s^4 * ((1 + s) / s)$ **by** linarith
 have $s5: s^5 = s * s^4$ **by** (simp add: eval-nat-numeral)
 have $Sb: \text{suminf } b \leq 42 * \exp s * (1 + s) / s^5$ **using** Sb' **unfolding** $s5$
by (simp add: field-simps)
 have $W: \text{weight } y = (2 * \pi)^4 * y^5 / 12 * E * \text{suminf } b$
proof –
 have $\text{suminf } a = E * \text{suminf } b$ **unfolding** ab **by** (rule suminf-mult[OF sb])
 thus ?thesis **unfolding** weight-def a-def[symmetric] **by** simp
qed
 have $C0: 0 \leq (2 * \pi)^4 * y^5 / 12 * E$ **using** y **by** (simp add: E-def)
 have $\text{weight } y \leq (2 * \pi)^4 * y^5 / 12 * E * (42 * \exp s * (1 + s) / s^5)$
unfolding W **by** (rule mult-left-mono[OF Sb C0])
 also have $\dots = 56 * \pi^4 * (\exp s * (1 + s)) * (y / s)^5 * E$
using $s0$ **by** (simp add: power-divide power-mult-distrib field-simps)
 also have $\dots \leq 56 * \pi^4 * (\exp s * (1 + s)) * ((1 + y) / \pi)^5 * E$
proof –
 have $ys: y / s \leq (1 + y) / \pi$
proof (cases $y \leq 1$)
 case True
 hence $s = \pi * y$ **by** (simp add: s-def Y-def)
 thus ?thesis **using** y **by** (simp add: field-simps)
 next
 case False
 hence $s = \pi$ **by** (simp add: s-def Y-def)
 thus ?thesis **by** (simp add: divide-right-mono)
qed
 have $(y / s)^5 \leq ((1 + y) / \pi)^5$ **using** $ys\ y\ s0$ **by** (intro power-mono)
 auto
 thus ?thesis **using** $s0$ **by** (intro mult-right-mono mult-left-mono) (auto simp:
 E-def)
qed
 also have $\dots = (56 * (\exp s * (1 + s)) / \pi) * (1 + y)^5 * E$
proof –
 have $p5: \pi^5 = \pi * \pi^4$ **by** (simp add: eval-nat-numeral)
 have $h: \pi^4 * ((1 + y) / \pi)^5 = (1 + y)^5 / \pi$ **unfolding** power-divide
 $p5$ **by** simp
 have $56 * \pi^4 * (\exp s * (1 + s)) * ((1 + y) / \pi)^5 * E$
 = $56 * (\exp s * (1 + s)) * (\pi^4 * ((1 + y) / \pi)^5) * E$ **by** (simp only:

```

mult-ac)
  also have ... = 56 * (exp s * (1 + s)) * ((1 + y) ^ 5 / pi) * E by (simp
only: h)
  also have ... = (56 * (exp s * (1 + s)) / pi) * (1 + y) ^ 5 * E by simp
  finally show ?thesis .
qed
also have ... ≤ 8192 * (1 + y) ^ 5 * E
proof -
  have exp s ≤ exp 4 using spi pi-less-4 by simp
  also have exp (4 :: real) = exp 1 ^ 4 using exp-of-nat-mult[of 4 1 :: real] by
simp
  also have ... ≤ 3 ^ 4 using exp-le by (intro power-mono) auto
  finally have es: exp s ≤ 81 by simp
  have s5: 1 + s ≤ 5 using spi pi-less-4 by simp
  have exp s * (1 + s) ≤ 81 * 5 using es s5 s0 by (intro mult-mono) auto
  hence 56 * (exp s * (1 + s)) / pi ≤ 56 * (81 * 5) / 3
    using pi-gt3 s0 by (intro frac-le) auto
  hence 56 * (exp s * (1 + s)) / pi ≤ 8192 by simp
  thus ?thesis using y by (intro mult-right-mono) (auto simp: E-def)
qed
finally show weight y ≤ 8192 * (1 + y) ^ 5 * exp (- 2 * pi * y) by (simp
add: E-def)
qed

```

theorem *DeltaK-zeta5-integral*:

```

assumes 40 dvd K
shows real-eval (DeltaK K) zeta5 = 1 / fact (hof K) *
  set-lebesgue-integral (PiM {..

```

proof –

Instead of the restricted measure on $(0, \infty)$ (whose product would need PiM of restricted spaces, absent from the library), apply Andreief on $lborel$ with the indicator of $(0, \infty)$ put into the first family.

```

define h where h = hof K
define g :: real ⇒ real where g =
  (λy. real-eval (Dm (Nof K)) (y ^ 2) ^ 6 / real-eval (Dm K) (y ^ 2) * weight
y)
define φ :: nat ⇒ real ⇒ real where φ =
  (λi y. y ^ (2 * i) * indicator {0<..} y)
define ψ :: nat ⇒ real ⇒ real where ψ =
  (λj y. y ^ (2 * j) * g y)
have gy: ∧t. real-eval (Dm (Nof K)) (t ^ 2) ^ 6 / real-eval (Dm K) (t ^ 2) *
weight t = g t
  by (simp add: g-def)

```

```

have S0: (0::nat) ∉ {1..K}
by simp
have DK: sqPoleProd {1..K} = Dm K
by (simp add: Dm-def sqPoleProd-def)
have pt:  $\bigwedge m y. \text{real-eval } (Dm (Nof K) \wedge^6 * pX \wedge^m) (y \wedge^2) / \text{real-eval } (Dm$ 
 $K) (y \wedge^2) *$ 
weight  $y = (y \wedge^2) \wedge^m * g y$ 
unfolding g-def real-eval-mult real-eval-power real-eval-pX
by (simp add: mult-ac)
have php:  $\bigwedge i j y. \varphi i y * \psi j y =$ 
indicator {0<..}  $y *_R ((y \wedge^2) \wedge^{(i+j)} * g y)$ 
unfolding  $\varphi$ -def  $\psi$ -def power-add power-mult
by (simp add: mult-ac)
have entry:  $\bigwedge i j. i < h \implies j < h \implies$ 
real-eval (GK K $$ (i, j)) zeta5 = integralL lborel ( $\lambda y. \varphi i y *$ 
 $\psi j y$ )
proof -
fix i j assume ij:  $i < h \wedge j < h$ 
have e1: GK K $$ (i, j) = muX {1..K} (Dm (Nof K)  $\wedge^6 * pX \wedge^{(i+j)}$ )
using ij by (simp add: GK-def h-def)
show real-eval (GK K $$ (i, j)) zeta5 =
integralL lborel ( $\lambda y. \varphi i y * \psi j y$ )
unfolding e1 muX-zeta5-integral[OF finite-atLeastAtMost S0] DK pt php
set-lebesgue-integral-def
by simp
qed
have detG: real-eval (DeltaK K) zeta5 =
Determinant.det (Matrix.mat h h ( $\lambda(i,$ 
 $j). \text{integral}^L \text{lborel } (\lambda y. \varphi i y * \psi j y)$ ))
proof -
have d1: real-eval (DeltaK K) zeta5 =
Determinant.det (map-mat ( $\lambda P. \text{real-eval } P \text{ zeta5}$ ) (GK K))
unfolding DeltaK-def by (rule comm-ring-hom.hom-det[OF real-eval-hom,
symmetric])
have d2: map-mat ( $\lambda P. \text{real-eval } P \text{ zeta5}$ ) (GK K) =
Matrix.mat h h ( $\lambda(i, j). \text{integral}^L \text{lborel } (\lambda y. \varphi i y *$ 
 $\psi j y)$ )
proof (rule eq-matI)
fix i j assume i:  $i < \text{dim-row } (Matrix.mat h h (\lambda(i,$ 
 $j). \text{integral}^L \text{lborel } (\lambda y. \varphi i y * \psi j y)))$ 
and j:  $j < \text{dim-col } (Matrix.mat h h (\lambda(i,$ 
 $j). \text{integral}^L \text{lborel } (\lambda y. \varphi i y * \psi j y)))$ 
show map-mat ( $\lambda P. \text{real-eval } P \text{ zeta5}$ ) (GK K) $$ (i, j) =
Matrix.mat h h ( $\lambda(i, j). \text{integral}^L \text{lborel } (\lambda y. \varphi i y *$ 
 $\psi j y)$ ) $$ (i, j)
using i j entry by (simp add: GK-def h-def)
next
show dim-row (map-mat ( $\lambda P. \text{real-eval } P \text{ zeta5}$ ) (GK K)) =
dim-row (Matrix.mat h h ( $\lambda(i,$ 

```

```

      j). integralL lborel (λy. φ i y * ψ j y)))
    by (simp add: GK-def h-def)
  next
    show dim-col (map-mat (λP. real-eval P zeta5) (GK K)) =
      dim-col (Matrix.mat h h (λ(i,
        j). integralL lborel (λy. φ i y * ψ j y)))
        by (simp add: GK-def h-def)
    qed
  show ?thesis
    unfolding d1 d2
    by simp
  qed
  have integ: ∧i j. i < h ⇒ j < h ⇒
    integrable lborel (λy. φ i y * ψ j y)
  proof -
    fix i j assume ij: i < h j < h
    have r: set-integrable lborel {0<..} (λy. real-eval (Dm (Nof K) ^ 6 * pX ^ (i
+
      j)) (y ^ 2) / real-eval (sqPoleProd {1..K}) (y ^ 2) * weight y)
      by (rule rat-integrable[OF finite-atLeastAtMost S0])
    show integrable lborel (λy. φ i y * ψ j y)
      using r[unfolded set-integrable-def DK pt]
      by (simp add: php)
    qed
  have A: Determinant.det (Matrix.mat h h (λ(i,
    j). integralL lborel (λy. φ i y * ψ j y)))
    = 1 / fact h * integralL (PiM {..<h} (λ-. lborel))
      (λx. Determinant.det (Matrix.mat h h (λ(a,
        b). φ a (x b))) * Determinant.det (Matrix.mat h h (λ(a,
        b). ψ a (x b))))
    by (rule andreief[OF sigma-finite-lborel integ])
  have V: ∧y :: nat ⇒ real. Determinant.det (Matrix.mat h h (λ(a,
    b). y b ^ (2 * a))) = (∏ i < h. ∏ j ∈ {j. i < j ∧
    j < h}. (y j ^ 2 - y i ^ 2))
  proof -
    fix y :: nat ⇒ real
    have Determinant.det (Matrix.mat h h (λ(a, b). y b ^ (2 * a))) =
      Determinant.det (Matrix.mat h h (λ(a, b). (y a ^ 2) ^ b))
      using andreief-det-swap[of h λa b. (y a ^ 2) ^ b] by (simp add: power-mult)
    also have ... = (∏ i < h. ∏ j ∈ {j. i < j ∧
      j < h}. (y j ^ 2 - y i ^ 2))
      using det-vandermonde-jnf[of h λa. y a ^ 2] by simp
    finally show Determinant.det (Matrix.mat h h (λ(a, b). y b ^ (2 * a))) = (∏
i <
      h. ∏ j ∈ {j. i < j ∧ j < h}. (y j ^ 2 - y i ^ 2)) .
    qed
  have Wphi: ∧y. Determinant.det (Matrix.mat h h (λ(a, b). φ a (y b))) =
    (∏ b < h. indicator {0<..} (y b)) * Determinant.det (Matrix.mat h h (λ(a,
    b). y b ^ (2 * a)))

```

```

    unfolding  $\varphi$ -def
  by (simp add: andreief-det-scale-cols)
have Wpsi:  $\bigwedge y. \text{Determinant.det } (\text{Matrix.mat } h \ h \ (\lambda(a, b). \psi \ a \ (y \ b))) =$ 
  ( $\prod b < h. g \ (y \ b)) * \text{Determinant.det } (\text{Matrix.mat } h \ h \ (\lambda(a, b). y \ b \ ^{(2 * a)}))$ 
  unfolding  $\psi$ -def
  by (simp add: andreief-det-scale-cols)
have ind:  $\bigwedge y :: \text{nat} \Rightarrow \text{real}. (\prod b < h. \text{indicator } \{0 <..\} \ (y \ b) :: \text{real}) =$ 
   $\text{indicator } \{y. \forall i < h. 0 < y \ i\} \ y$ 
proof -
  fix  $y :: \text{nat} \Rightarrow \text{real}$ 
  show  $(\prod b < h. \text{indicator } \{0 <..\} \ (y \ b) :: \text{real}) = \text{indicator } \{y. \forall i < h. 0 < y \ i\} \ y$ 
i} y
  proof (cases  $\forall i < h. 0 < y \ i$ )
    case True
    then show ?thesis
      by (simp add: indicator-def)
  next
    case False
    then show ?thesis
      by (auto simp: indicator-def prod-zero-iff)
  qed
qed
have sq:  $\bigwedge a \ b :: \text{real}. (b - a) * (b - a) = (a - b) \ ^2$ 
  by (simp add: power2-eq-square algebra-simps)
have Vsq:  $\bigwedge y :: \text{nat} \Rightarrow \text{real}. (\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \ j \ ^2 - y \ i \ ^2)) * (\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \ j \ ^2 - y \ i \ ^2))$ 
  =  $(\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \ i \ ^2 - y \ j \ ^2) \ ^2)$ 
  unfolding prod.distrib[symmetric] sq
  by simp
have pw:  $\bigwedge y. \text{Determinant.det } (\text{Matrix.mat } h \ h \ (\lambda(a, b). \varphi \ a \ (y \ b))) * \text{Determinant.det } (\text{Matrix.mat } h \ h \ (\lambda(a, b). \psi \ a \ (y \ b)))$ 
  =  $\text{indicator } \{y. \forall i < h. 0 < y \ i\} \ y *_R ((\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \ i \ ^2 - y \ j \ ^2) \ ^2) * (\prod i < h. g \ (y \ i)))$ 
  unfolding Wphi Wpsi ind V Vsq[symmetric]
  by (simp add: mult-ac)
have fin:  $\text{integral}^L \ (\text{PiM } \{..<h\} \ (\lambda-. \text{lborel})) \ (\lambda x. \text{Determinant.det } (\text{Matrix.mat } h \ h \ (\lambda(a, b). \varphi \ a \ (x \ b))) * \text{Determinant.det } (\text{Matrix.mat } h \ h \ (\lambda(a, b). \psi \ a \ (x \ b))))$ 
  =  $\text{set-lebesgue-integral } (\text{PiM } \{..<h\} \ (\lambda-. \text{lborel})) \ \{y. \forall i < h. 0 < y \ i\} \ (\lambda y. (\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \ i \ ^2 - y \ j \ ^2) \ ^2) * (\prod i < h. g \ (y \ i)))$ 
  unfolding set-lebesgue-integral-def pw
  by simp
show ?thesis
  unfolding h-def[symmetric] gy detG A fin
  by simp
qed

```


30.2 (6.12) and the one-coordinate bound

(6.12): $0 \leq \sum_{j \leq m} \log(t + (j/K)^2) - K \int_0^{m/K} \log(t + u^2) du \leq 2 \log m + 2.$

lemma *sum-ge-int*:

$0 < t \implies 0 < \kappa \implies$

$\kappa * \text{integral } \{0.. \text{real } m / \kappa\} (\lambda u. \ln (t + u^2)) \leq (\sum j = 1..m. \ln (t + (\text{real } j / \kappa)^2))$

proof (*induction m*)

case 0

thus ?case **by** *simp*

next

case (*Suc m*)

define *f* **where** $f u = \ln (t + u^2)$ **for** $u :: \text{real}$

have *fe*: $(\lambda u. \ln (t + u^2)) = f$ **by** (*simp add: f-def fun-eq-iff*)

have *t*: $0 < t$ **and** *k*: $0 < \kappa$ **using** *Suc.prem*s **by** *auto*

have *tp*: $0 < t + u^2$ **for** $u :: \text{real}$ **using** *t* **by** (*simp add: add-pos-nonneg*)

have *cont*: *continuous-on* {*a..b*} *f* **for** *a b*

unfolding *f-def* **by** (*intro continuous-intros*) (*use tp in <auto simp: less-le>*)

have *intg*: *f* *integrable-on* {*a..b*} **for** *a b* **by** (*rule integrable-continuous-interval* [*OF cont*])

define *a* **where** $a = \text{real } m / \kappa$

define *b* **where** $b = \text{real } (\text{Suc } m) / \kappa$

have *ab*: $0 \leq a \leq b$ **using** *k* **by** (*auto simp: a-def b-def divide-right-mono*)

have *comb*: $\text{integral } \{0..a\} f + \text{integral } \{a..b\} f = \text{integral } \{0..b\} f$

by (*rule Henstock-Kurzweil-Integration.integral-combine* [*OF ab intg*])

have $\text{integral } \{a..b\} f \leq \text{integral } \{a..b\} (\lambda-. f b)$

proof (*rule integral-le* [*OF intg*])

show $(\lambda-. f b)$ *integrable-on* {*a..b*}

by (*rule integrable-continuous-interval*) (*rule continuous-on-const*)

fix *u* **assume** $u \in \{a..b\}$

hence $0 \leq u \leq b$ **using** *ab* **by** *auto*

hence $u^2 \leq b^2$ **by** (*intro power-mono*) *auto*

thus $f u \leq f b$ **unfolding** *f-def* **using** *tp* **by** (*subst ln-le-cancel-iff*) *auto*

qed

also have $\dots = (b - a) * f b$ **using** *ab* **by** *simp*

also have $b - a = 1 / \kappa$ **using** *k* **by** (*simp add: a-def b-def field-simps*)

finally have *le'*: $\kappa * \text{integral } \{a..b\} f \leq f b$ **using** *k* **by** (*simp add: field-simps*)

have *IH*: $\kappa * \text{integral } \{0..a\} f \leq (\sum j = 1..m. \ln (t + (\text{real } j / \kappa)^2))$

using *Suc.IH* [*OF t k*] **unfolding** *fe a-def* .

have $\kappa * \text{integral } \{0..b\} f = \kappa * \text{integral } \{0..a\} f + \kappa * \text{integral } \{a..b\} f$

by (*simp add: comb[symmetric] distrib-left*)

also have $\dots \leq (\sum j = 1..m. \ln (t + (\text{real } j / \kappa)^2)) + f b$

using *IH le'* **by** *linarith*

also have $\dots = (\sum j = 1..\text{Suc } m. \ln (t + (\text{real } j / \kappa)^2))$

by (*simp add: b-def f-def*)

finally show ?case **unfolding** *fe b-def* .

qed

```

lemma sum-log-le:  $1 \leq m \implies$ 
   $(\sum k = 1..m. \ln (\text{real } k)) \leq \text{real } m * \ln (\text{real } m) - \text{real } m + \ln (\text{real } m) + 1$ 
proof -
  assume  $1 \leq m$ 
  thus ?thesis
  proof (induction m rule: nat-induct-at-least)
    case base
    show ?case
    by simp
  next
    case (Suc m)
    define A :: real where  $A = \ln (\text{real } m)$ 
    define B :: real where  $B = \ln (\text{real } (\text{Suc } m))$ 
    have m1:  $1 \leq \text{real } m$  using Suc.hyps
    by simp
    have IH:  $(\sum k = 1..m. \ln (\text{real } k)) \leq \text{real } m * A - \text{real } m + A + 1$  using
Suc.IH
    unfolding A-def .
    have  $\ln (\text{real } m / \text{real } (\text{Suc } m)) \leq \text{real } m / \text{real } (\text{Suc } m) - 1$ 
    using m1 ln-le-minus-one[of real m / real (Suc m)]
    by simp
    moreover have  $\ln (\text{real } m / \text{real } (\text{Suc } m)) = A - B$  using m1 unfolding
A-def B-def
    by (simp add: ln-div)
    moreover have  $\text{real } m / \text{real } (\text{Suc } m) - 1 = - 1 / \text{real } (\text{Suc } m)$  using m1
    by (simp add: field-simps)
    ultimately have d4a:  $A - B \leq - 1 / \text{real } (\text{Suc } m)$ 
    by simp
    have d4:  $\text{real } (\text{Suc } m) * (A - B) \leq - 1$  using mult-left-mono[OF d4a, of real
(Suc m)] m1
    by simp
    have f:  $(\sum k = 1..\text{Suc } m. \ln (\text{real } k)) = B + (\sum k = 1..m. \ln (\text{real } k))$ 
    unfolding B-def
    by simp
    have e:  $\text{real } (\text{Suc } m) * (A - B) = \text{real } m * A + A - \text{real } (\text{Suc } m) * B$ 
    by (simp add: algebra-simps)
    have s:  $\text{real } (\text{Suc } m) = \text{real } m + 1$ 
    by simp
    have  $(\sum k = 1..\text{Suc } m. \ln (\text{real } k)) \leq \text{real } (\text{Suc } m) * B - \text{real } (\text{Suc } m) + B$ 
    using e f d4 IH s
    by simp
    thus ?case unfolding B-def .
  qed
qed

lemma ln-int0:  $0 < (c :: \text{real}) \implies \ln \text{ integrable-on } \{0..c\}$ 
using set-borel-integral-eq-integral(1)[OF ln-integrable-near-zero] integrable-on-Icc-iff-Ioo
by auto

```

lemma *integral-ln-0*: $0 < (b :: \text{real}) \implies$
 $\text{integral } \{0..b\} (\lambda u. \ln u) = b * \ln b - b$
proof –
 assume $b: 0 < b$
 have $\text{integral } \{0..b\} (\lambda u. \ln u) =$
 $\text{integral } \{0..b\} (\lambda u. (1 / 2) *_{\mathbb{R}} \ln (0 + u^2))$
 by (*simp add: ln-realpow*)
 also have $\dots = (1 / 2) * (2 * (b * \ln b - b))$ **using** *Ic-zero[OF b]*
 by *simp*
 also have $\dots = b * \ln b - b$
 by *simp*
 finally show *?thesis* .
qed

Telescoping a sum of integrals over the cells $[(j-1)/\kappa, j/\kappa]$.

lemma *tele-gen*:
fixes $g :: \text{real} \Rightarrow \text{real}$ **and** $\kappa :: \text{real}$
assumes $k: 0 < \kappa$ **and** $gi: \bigwedge c. 0 < c \implies g \text{ integrable-on } \{0..c\}$
shows $(\sum j = 1..n. \text{integral } \{\text{real } (j-1) / \kappa.. \text{real } j / \kappa\} g) =$
 $\text{integral } \{0.. \text{real } n / \kappa\} g$
proof (*induction n*)
 case 0
 show *?case*
 by *simp*
next
 case (*Suc n*)
 have $a: 0 \leq \text{real } n / \kappa$ **using** k
 by *simp*
 have $b: \text{real } n / \kappa \leq \text{real } (\text{Suc } n) / \kappa$ **using** k
 by (*simp add: divide-right-mono*)
 have $i: g \text{ integrable-on } \{0.. \text{real } (\text{Suc } n) / \kappa\}$ **using** $gi \ k$
 by *simp*
 have $c: \text{integral } \{0.. \text{real } n / \kappa\} g + \text{integral } \{\text{real } n / \kappa.. \text{real } (\text{Suc } n) / \kappa\} g = \text{integral } \{0.. \text{real } (\text{Suc } n) / \kappa\} g$
using *Henstock-Kurzweil-Integration.integral-combine[OF a b i]* .
 show *?case* **using** *Suc.IH c*
 by *simp*
qed

(6.12): $\sum_{j \leq m} \log(t + (j/\kappa)^2) - \kappa \int_{0^{m/\kappa}} \log(t + u^2) du \leq 2 \log m + 2$: on each cell $[(j-1)/\kappa, j/\kappa]$, $\log(t + b^2) - \log(t + u^2) \leq 2 \log b - 2 \log u$.

lemma *sum-le-int*:
 $0 < t \implies 0 < \kappa \implies 1 \leq m \implies$
 $(\sum j = 1..m. \ln(t + (\text{real } j / \kappa)^2)) \leq$
 $\kappa * \text{integral } \{0.. \text{real } m / \kappa\} (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } m) +$
 2
proof –

```

assume  $t: 0 < t$  and  $k: 0 < \kappa$  and  $m: 1 \leq m$ 
have  $cell: \bigwedge j. 1 \leq j \implies$ 
   $\ln (t + (\text{real } j / \kappa) ^ 2) / \kappa - \text{integral } \{ \text{real } (j - 1) / \kappa .. \text{real } j /$ 
   $\kappa \} (\lambda u. \ln (t + u ^ 2))$ 
   $\leq \text{integral } \{ \text{real } (j - 1) / \kappa .. \text{real } j / \kappa \} (\lambda u. 2 * \ln (\text{real } j / \kappa) - 2 * \ln u)$ 
proof -
  fix  $j :: \text{nat}$  assume  $j: 1 \leq j$ 
  define  $a$  where  $a = \text{real } (j - 1) / \kappa$ 
  define  $b$  where  $b = \text{real } j / \kappa$ 
  have  $a0: 0 \leq a$  unfolding  $a\text{-def}$  using  $k$ 
    by  $\text{simp}$ 
  have  $b0: 0 < b$  unfolding  $b\text{-def}$  using  $k\ j$ 
    by  $\text{simp}$ 
  have  $ab: a \leq b$  unfolding  $a\text{-def}$   $b\text{-def}$  using  $k\ j$ 
    by  $(\text{simp add: divide-right-mono})$ 
  have  $ba: b - a = 1 / \kappa$  unfolding  $a\text{-def}$   $b\text{-def}$  using  $j$ 
    by  $(\text{simp add: of-nat-diff diff-divide-distrib})$ 
  have  $tp: 0 < t + u ^ 2$  for  $u :: \text{real}$  using  $t$ 
    by  $(\text{simp add: add-pos-nonneg})$ 
  have  $t + u ^ 2 \neq 0$  for  $u :: \text{real}$  using  $tp[\text{of } u]$ 
    by  $\text{simp}$ 
  hence  $c1: \text{continuous-on } \{a..b\} (\lambda u. \ln (t + u ^ 2))$ 
    by  $(\text{fast intro: continuous-intros})$ 
  have  $i1: (\lambda u. \ln (t + u ^ 2)) \text{integrable-on } \{a..b\}$ 
    using  $\text{integrable-continuous-interval}[OF\ c1]$ 
    by  $\text{simp}$ 
  have  $i2: (\lambda u. 2 * \ln b - 2 * \ln u) \text{integrable-on } \{a..b\}$ 
proof -
  have  $l: \ln \text{integrable-on } \{a..b\}$ 
    using  $\text{integrable-on-subinterval}[OF\ \ln\text{-int0}[OF\ b0], \text{ of } a\ b]\ a0$ 
    by  $\text{simp}$ 
  show  $?thesis$  using  $l$ 
  by  $(\text{auto intro: Henstock-Kurzweil-Integration.integrable-diff integrable-const-ivl integrable-on-cmult-left})$ 
qed
  have  $i3: (\lambda u. \ln (t + b ^ 2) - \ln (t + u ^ 2)) \text{integrable-on } \{a..b\}$  using  $i1$ 
  by  $(\text{auto intro: Henstock-Kurzweil-Integration.integrable-diff integrable-const-ivl})$ 
  have  $pw: \ln (t + b ^ 2) - \ln (t + u ^ 2) \leq 2 * \ln b - 2 * \ln u$  if  $u: 0 < u$ 
   $\leq b$ 
    for  $u$ 
proof -
  have  $m: (t + b ^ 2) * u ^ 2 \leq b ^ 2 * (t + u ^ 2)$ 
proof -
  have  $u ^ 2 \leq b ^ 2$  using  $u$ 
    by  $(\text{simp add: power-mono})$ 
  hence  $t * u ^ 2 \leq t * b ^ 2$  using  $t$ 
    by  $(\text{simp add: mult-left-mono})$ 
  thus  $?thesis$ 

```

```

    by (simp add: algebra-simps)
  qed
  have l:  $\ln ((t + b^2) * u^2) \leq \ln (b^2 * (t + u^2))$ 
    using m u tp[of u] tp[of b] b0
    by simp
  have l1:  $\ln ((t + b^2) * u^2) = \ln (t + b^2) + 2 * \ln u$  using u tp[of
b]
    by (simp add: ln-mult ln-realpow)
  have l2:  $\ln (b^2 * (t + u^2)) = 2 * \ln b + \ln (t + u^2)$  using b0 tp[of
u]
    by (simp add: ln-mult ln-realpow)
  show ?thesis using l l1 l2
    by simp
  qed
  have le:  $\text{integral } \{a <..< b\} (\lambda u. \ln (t + b^2) - \ln (t + u^2)) \leq$ 
 $\text{integral } \{a <..< b\} (\lambda u. 2 * \ln b - 2 * \ln u)$ 
  proof (rule integral-le)
    show  $(\lambda u. \ln (t + b^2) - \ln (t + u^2))$  integrable-on  $\{a <..< b\}$  using i3
      by (simp add: integrable-on-Icc-iff-Ioo[symmetric])
    show  $(\lambda u. 2 * \ln b - 2 * \ln u)$  integrable-on  $\{a <..< b\}$  using i2
      by (simp add: integrable-on-Icc-iff-Ioo[symmetric])
    fix u assume u  $\in \{a <..< b\}$ 
    thus  $\ln (t + b^2) - \ln (t + u^2) \leq 2 * \ln b - 2 * \ln u$  using pw a0
      by simp
  qed
  have eqL:  $\ln (t + b^2) / \kappa - \text{integral } \{a..b\} (\lambda u. \ln (t + u^2)) =$ 
 $\text{integral } \{a <..< b\} (\lambda u. \ln (t + b^2) - \ln (t + u^2))$ 
  proof -
    have  $\text{integral } \{a..b\} (\lambda u. \ln (t + b^2) - \ln (t + u^2)) =$ 
 $\text{integral } \{a..b\} (\lambda u. \ln (t + b^2)) - \text{integral } \{a..b\} (\lambda u. \ln (t + u^2))$ 
      using i1
      using Henstock-Kurzweil-Integration.integral-diff by blast
    moreover have  $\text{integral } \{a..b\} (\lambda u. \ln (t + b^2)) = \ln (t + b^2) / \kappa$ 
      using ab ba
      by (simp add: content-real)
    moreover have  $\text{integral } \{a..b\} (\lambda u. \ln (t + b^2) - \ln (t + u^2)) =$ 
 $\text{integral } \{a <..< b\} (\lambda u. \ln (t + b^2) - \ln (t + u^2))$ 
      by (simp add: integral-open-interval-real)
    ultimately show ?thesis
      by simp
  qed
  have eqR:  $\text{integral } \{a..b\} (\lambda u. 2 * \ln b - 2 * \ln u) =$ 
 $\text{integral } \{a <..< b\} (\lambda u. 2 * \ln b - 2 * \ln u)$ 
    by (simp add: integral-open-interval-real)
  show  $\ln (t + (\text{real } j / \kappa)^2) / \kappa - \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j /$ 
 $\kappa\} (\lambda u. \ln (t + u^2))$ 
 $\leq \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 *$ 
 $\ln (\text{real } j / \kappa) - 2 * \ln u)$ 
    using le eqL eqR unfolding a-def b-def

```

```

    by simp
  qed
  have fi:  $(\lambda u. \ln (t + u^2))$  integrable-on  $\{0..c\}$  for  $c$ 
  proof -
    have  $t + u^2 \neq 0$  for  $u :: \text{real}$  using add-pos-nonneg[OF t zero-le-power2[of
u]]
    by simp
    hence continuous-on  $\{0..c\}$   $(\lambda u. \ln (t + u^2))$ 
    by (fast intro: continuous-intros)
    thus ?thesis
    by (auto intro: integrable-continuous-interval)
  qed
  have tele:  $(\sum j = 1..m. \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. \ln (t + u^2))) =$ 
 $\text{integral } \{0.. \text{real } m / \kappa\} (\lambda u. \ln (t + u^2))$ 
    using tele-gen[OF k fi]
    by simp
  have rhs:  $(\sum j = 1..m. \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 * \ln (\text{real } j / \kappa) - 2 * \ln u))$ 
 $= (2 / \kappa) * (\sum j =$ 
 $1..m. \ln (\text{real } j)) - 2 * (\text{real } m / \kappa) * \ln (\text{real } m) + 2 * \text{real } m / \kappa$ 
  proof -
    have li:  $\ln$  integrable-on  $\{0..c\}$  if  $0 < c$  for  $c :: \text{real}$  using ln-int0[OF that] .
    have cellR:  $\text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 * \ln (\text{real } j / \kappa) - 2 * \ln u) = 2 * \ln (\text{real } j / \kappa) / \kappa - 2 * \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} \ln$  if  $j: 1 \leq j$  for  $j$ 
    proof -
      have jb:  $0 < \text{real } j / \kappa$  using j k
      by simp
      have lc:  $\ln$  integrable-on  $\{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\}$ 
      using integrable-on-subinterval[OF li[OF jb],
of  $\text{real } (j - 1) / \kappa$   $\text{real } j / \kappa$ ] k
      by (simp add: divide-right-mono)
      have l2:  $(\lambda u. 2 * \ln u)$  integrable-on  $\{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\}$ 
      using lc
      by (auto intro: integrable-on-cmult-left)
      have d:  $\text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 * \ln (\text{real } j / \kappa) - 2 * \ln u) = \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 * \ln (\text{real } j / \kappa)) - \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 * \ln u)$ 
      using l2 Henstock-Kurzweil-Integration.integral-diff[OF integrable-const-ivl
l2]
      by simp
      have c:  $\text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 * \ln (\text{real } j / \kappa)) = 2 * \ln (\text{real } j / \kappa) / \kappa$  using j k
      by (simp add: content-real of-nat-diff diff-divide-distrib)
      have e:  $\text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} (\lambda u. 2 * \ln u) = 2 * \text{integral } \{\text{real } (j - 1) / \kappa.. \text{real } j / \kappa\} \ln$ 
      using integral-mult[OF lc, of 2]

```

```

    by simp
  show ?thesis unfolding d c e ..
qed
have s1: (∑ j = 1..m. integral {real (j - 1) / κ..real j /
  κ} (λu. 2 * ln (real j / κ) - 2 * ln u)) = (∑ j =
  1..m. 2 * ln (real j / κ) / κ) - 2 * (∑ j =
  1..m. integral {real (j - 1) / κ..real j / κ} ln)
  using cellR
  by (simp add: sum-subtractf sum-distrib-left)
have lnd: ln (real j / κ) = ln (real j) - ln κ if 1 ≤ j for j using that k
  by (simp add: ln-div)
have s2: (∑ j = 1..m. 2 * ln (real j / κ) / κ) =
  (2 / κ) * (∑ j = 1..m. ln (real j)) - 2 * real m / κ * ln κ
proof -
  have (∑ j = 1..m. 2 * ln (real j / κ) / κ) = (∑ j =
    1..m. (2 / κ) * ln (real j) - (2 / κ) * ln κ)
  proof (rule sum.cong)
    fix j assume j ∈ {1..m}
    hence j1: 1 ≤ j
    by simp
    show 2 * ln (real j / κ) / κ =
      (2 / κ) * ln (real j) - (2 / κ) * ln κ using lnd[OF j1]
    by (simp add: diff-divide-distrib right-diff-distrib)
  next
    show {1..m} = {1..m}
    by simp
  qed
  also have ... = (2 / κ) * (∑ j =
    1..m. ln (real j)) - 2 * real m / κ * ln κ
    by (simp add: sum-subtractf sum-distrib-left)
  finally show ?thesis .
qed
have sumL: (∑ j = 1..m. integral {real (j - 1) / κ..real j / κ} ln) =
  integral {0..real m / κ} ln using tele-gen[OF k li] .
have mk: 0 < real m / κ using m k
  by simp
have lnI: integral {0..real m / κ} ln =
  real m / κ * ln (real m / κ) - real m / κ using integral-ln-0[OF mk] .
have lnm: ln (real m / κ) = ln (real m) - ln κ using m k
  by (simp add: ln-div)
show ?thesis unfolding s1 s2 sumL lnI lnm
  by (simp add: algebra-simps)
qed
have (∑ j = 1..m. ln (t + (real j / κ) ^ 2)) / κ - integral {0..real m /
  κ} (λu. ln (t + u ^ 2))
  ≤ (2 / κ) * (∑ j =
    1..m. ln (real j)) - 2 * (real m / κ) * ln (real m) + 2 * real m / κ
  unfolding tele[symmetric] rhs[symmetric] sum-divide-distrib sum-subtractf[symmetric]
  by (rule sum-mono) (use cell in simp)

```

also have $\dots \leq (2 / \kappa) * (\ln (\text{real } m) + 1)$
using *sum-log-le*[*OF m*] *k* **by** (*simp add: field-simps*)
finally have $F: (\sum j = 1..m. \ln (t + (\text{real } j / \kappa) ^ 2)) / \kappa -$
 $\text{integral } \{0..\text{real } m / \kappa\} (\lambda u. \ln (t + u ^ 2)) \leq$
 $(2 / \kappa) * (\ln (\text{real } m) + 1) .$
have $\kappa * ((\sum j = 1..m. \ln (t + (\text{real } j / \kappa) ^ 2)) / \kappa -$
 $\text{integral } \{0..\text{real } m / \kappa\} (\lambda u. \ln (t + u ^ 2))) \leq$
 $\kappa * ((2 / \kappa) * (\ln (\text{real } m) + 1))$
using *mult-left-mono*[*OF F less-imp-le*[*OF k*]] **by** *simp*
moreover have $\kappa * ((\sum j =$
 $1..m. \ln (t + (\text{real } j / \kappa) ^ 2)) / \kappa - \text{integral } \{0..\text{real } m /$
 $\kappa\} (\lambda u. \ln (t + u ^ 2))) = (\sum j =$
 $1..m. \ln (t + (\text{real } j / \kappa) ^ 2)) - \kappa * \text{integral } \{0..\text{real } m /$
 $\kappa\} (\lambda u. \ln (t + u ^ 2))$
using *k* **by** (*simp add: algebra-simps*)
moreover have $\kappa * ((2 / \kappa) * (\ln (\text{real } m) + 1)) = 2 * \ln (\text{real } m) + 2$
using *k* **by** (*simp add: algebra-simps*)
ultimately show *?thesis* **by** *simp*
qed

The one-coordinate bound: (6.11) and (6.12) give $D \cdot N(y^2)^6 / D \cdot K(y^2)$
 $w(y) \leq 8192 e^{1/2} K^{12N+12} / K^{2K} (1+y)^5 e^{-K V((y/K)^2)}$.

lemma *g-le*:

assumes $1 \leq N \leq K$ $\text{real } N / \text{real } K = 3 / 4$ $0 < y$
shows $\text{real-eval } (Dm \ N) (y ^ 2) ^ 6 / \text{real-eval } (Dm \ K) (y ^ 2) * \text{weight } y \leq$
 $8192 * \exp 12 * \text{real } K ^ (12 * N + 12) / \text{real } K ^ (2 * K) *$
 $((1 + y) ^ 5 * \exp (- \text{real } K * \text{Vf } ((y / \text{real } K) ^ 2)))$
proof –
define *t* **where** $t = (y / \text{real } K) ^ 2$
define κ **where** $\kappa = \text{real } K$
have *K0*: $0 < \kappa$ **using** *assms* **unfolding** κ -*def* **by** *simp*
have *t0*: $0 < t$ **using** *assms* **unfolding** *t*-*def* **by** *simp*
have *Dm-exp*: $\bigwedge m. \text{real-eval } (Dm \ m) (y ^ 2) = \kappa ^ (2 * m) * \exp (\sum j =$
 $1..m. \ln (t + (\text{real } j / \kappa) ^ 2))$
proof –
fix *m*
have *f*: $\text{real-eval } (Dm \ m) (y ^ 2) = (\prod j = 1..m. y ^ 2 + \text{real } j ^ 2)$
unfolding *Dm-def* *real-eval-prod*
by (*simp add: real-eval-simps of-rat-power*)
have *each*: $y ^ 2 + \text{real } j ^ 2 = \kappa ^ 2 * \exp (\ln (t + (\text{real } j / \kappa) ^ 2))$ **for** *j*
proof –
have *p*: $0 < t + (\text{real } j / \kappa) ^ 2$ **using** *t0*
by (*simp add: add-pos-nonneg*)
have $y ^ 2 + \text{real } j ^ 2 = \kappa ^ 2 * (t + (\text{real } j / \kappa) ^ 2)$
unfolding *t-def* κ -*def* **using** *K0* **unfolding** κ -*def*
by (*simp add: power-divide field-simps*)
thus *?thesis* **using** *p*
by *simp*
qed

have $(\prod j = 1..m. y^2 + \text{real } j^2) = (\prod j = 1..m. \kappa^2 * \exp(\ln(t + (\text{real } j / \kappa)^2)))$ **unfolding each ..**
also have $\dots = (\kappa^2)^m * \exp(\sum j = 1..m. \ln(t + (\text{real } j / \kappa)^2))$
by (*simp add: prod.distrib exp-sum*)
also have $\dots = \kappa^{(2 * m)} * \exp(\sum j = 1..m. \ln(t + (\text{real } j / \kappa)^2))$
by (*simp add: power-mult*)
finally show $\text{real-eval } (Dm \ m) \ (y^2) = \kappa^{(2 * m)} * \exp(\sum j = 1..m. \ln(t + (\text{real } j / \kappa)^2))$ **unfolding f .**
qed
have *lower*: $\kappa * \text{integral } \{0.. \text{real } K / \kappa\} \ (\lambda u. \ln(t + u^2)) \leq$
 $(\sum j = 1..K. \ln(t + (\text{real } j / \kappa)^2))$
by (*rule sum-ge-int[OF t0 K0]*)
have *upper*: $(\sum j = 1..N. \ln(t + (\text{real } j / \kappa)^2)) \leq$
 $\kappa * \text{integral } \{0.. \text{real } N / \kappa\} \ (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N) + 2$
by (*rule sum-le-int[OF t0 K0 assms(1)]*)
have $V: \kappa * \forall f \ t = 2 * \pi * y + \kappa * \text{integral } \{0..1\} \ (\lambda u. \ln(t + u^2)) -$
 $6 * \kappa * \text{integral } \{0..3 / 40\} \ (\lambda u. \ln(t + u^2))$
unfolding *Vf-def t-def κ-def*
using *real-sqrt-abs[of y / real K] assms(4) K0[unfolded κ-def]*
by (*simp add: algebra-simps*)
have *ends*: $\text{real } K / \kappa = 1 \ \text{real } N / \kappa = 3 / 40$ **using** *assms(3) K0*
unfolding *κ-def* **by** *simp-all*
have *w*: $\text{weight } y \leq 8192 * (1 + y)^5 * \exp(-2 * \pi * y)$ **by** (*rule weight-le[OF assms(4)]*)
have $\text{real-eval } (Dm \ N) \ (y^2)^6 / \text{real-eval } (Dm \ K) \ (y^2) * \text{weight } y$
 $\leq \kappa^{(12 * N)} / \kappa^{(2 * K)} * \exp(6 * (\kappa * \text{integral } \{0..3 / 40\} \ (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N) + 2) - \kappa * \text{integral } \{0..1\} \ (\lambda u. \ln(t + u^2))) * (8192 * (1 + y)^5 * \exp(-2$
 $* \pi * y))$
proof –
have *q*: $\text{real-eval } (Dm \ N) \ (y^2)^6 / \text{real-eval } (Dm \ K) \ (y^2) =$
 $\kappa^{(12 * N)} / \kappa^{(2 * K)} * \exp(6 * (\sum j = 1..N. \ln(t + (\text{real } j / \kappa)^2)) - (\sum j = 1..K. \ln(t + (\text{real } j / \kappa)^2)))$
unfolding *Dm-exp* **using** *exp-of-nat-mult[of 6 ∑ j = 1..N. ln(t + (real j / κ)²)]*
by (*simp add: power-mult-distrib exp-diff power-mult[symmetric]*)
have *ex*: $\exp(6 * (\sum j = 1..N. \ln(t + (\text{real } j / \kappa)^2)) - (\sum j = 1..K. \ln(t + (\text{real } j / \kappa)^2))) \leq$
 $\exp(6 * (\kappa * \text{integral } \{0..3 / 40\} \ (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N) + 2) - \kappa * \text{integral } \{0..1\} \ (\lambda u. \ln(t + u^2)))$
using *upper lower* **unfolding** *ends*
by *simp*
have *w0*: $0 < \text{weight } y$ **by** (*rule weight-pos[OF assms(4)]*)
have *c0*: $0 \leq \kappa^{(12 * N)} / \kappa^{(2 * K)}$ **using** *K0*
by *simp*

```

have real-eval (Dm N) (y ^ 2) ^ 6 / real-eval (Dm K) (y ^ 2) * weight y =
   $\kappa^{(12 * N) / \kappa^{(2 * K)} * \exp(6 * (\sum j =$ 
   $1..N. \ln(t + (\text{real } j / \kappa)^2)) - (\sum j =$ 
   $1..K. \ln(t + (\text{real } j / \kappa)^2))) * \text{weight } y$ 
unfolding q ..
also have ... ≤  $\kappa^{(12 * N) / \kappa^{(2 * K)} * \exp(6 * (\kappa * \text{integral } \{0..3 / 40\} (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N) + 2) - \kappa * \text{integral } \{0..1\} (\lambda u. \ln(t + u^2))) * (8192 * (1 + y)^5 * \exp(-2 * \pi$ 
* y))
using mult-mono[OF mult-left-mono[OF ex c0] w] less-imp-le[OF w0] mult-nonneg-nonneg[OF
c0
exp-ge-zero]
by simp
finally show ?thesis .
qed
also have ... ≤  $8192 * \exp 12 * \text{real } K^{(12 * N + 12) / \text{real } K^{(2 * K)} * ((1 + y)^5 * \exp(-\text{real } K * \text{Vf } ((y / \text{real } K)^2)))$ 
proof -
have V':  $6 * (\kappa * \text{integral } \{0..3 / 40\} (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N)$ 
+
 $2) - \kappa * \text{integral } \{0..1\} (\lambda u. \ln(t + u^2)) + (-2 * \pi * y) =$ 
 $12 * \ln(\text{real } N) + 12 + (-\kappa * \text{Vf } t)$ 
using V
by simp
have N0:  $0 < \text{real } N$  using assms(1)
by simp
have e1:  $\exp(6 * (\kappa * \text{integral } \{0..3 / 40\} (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N) + 2) - \kappa * \text{integral } \{0..1\} (\lambda u. \ln(t + u^2))) * \exp(-2 * \pi$ 
*
 $y) = \text{real } N^{12} * \exp 12 * \exp(-\kappa * \text{Vf } t)$ 
proof -
have  $\exp(6 * (\kappa * \text{integral } \{0..3 / 40\} (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N) + 2) - \kappa * \text{integral } \{0..1\} (\lambda u. \ln(t + u^2))) * \exp(-2 * \pi * y) = \exp(12 * \ln(\text{real } N)) * \exp 12 * \exp(-\kappa * \text{Vf } t)$ 
unfolding mult-exp-exp V'
by (simp add: exp-add)
also have  $\exp(12 * \ln(\text{real } N)) = \text{real } N^{12}$ 
using N0 exp-ln[of real N ^ 12] ln-realpow[of real N 12]
by simp
finally show ?thesis .
qed
have N12:  $\text{real } N^{12} \leq \text{real } K^{12}$  using assms(2)
by (simp add: power-mono)
have  $\kappa^{(12 * N) / \kappa^{(2 * K)} * \exp(6 * (\kappa * \text{integral } \{0..3 / 40\} (\lambda u. \ln(t + u^2)) + 2 * \ln(\text{real } N) + 2) - \kappa * \text{integral } \{0..1\} (\lambda u. \ln(t + u^2))) * (8192 * (1 + y)^5 * \exp(-2 * \pi$ 
* y))
=  $8192 * \exp 12 * \kappa^{(12 * N) * \text{real } N^{12} / \kappa^{(2 * K)} * ((1 + y)^5$ 

```

```

5 *
  exp (- κ * Vf t))
  using e1
  by (simp add: algebra-simps)
  also have ... ≤ 8192 * exp 12 * κ ^ (12 * N) * real K ^ 12 / κ ^ (2 *
    K) * ((1 + y) ^ 5 * exp (- κ * Vf t))
  using N12 K0 asms(4)
  by (simp add: mult-left-mono mult-right-mono divide-right-mono)
  also have ... = 8192 * exp 12 * real K ^ (12 * N + 12) / real K ^ (2 * K)
  * ((1 + y) ^ 5 *
    exp (- real K * Vf ((y / real K) ^ 2)))
  unfolding κ-def t-def
  by (simp add: power-add)
  finally show ?thesis .
qed
  finally show ?thesis .
qed

```

30.3 (6.9) The bound for every configuration

(6.9): for every configuration $0 \leq t-1, \dots, t-h$ of distinct points, $2\sum -\{i < j\} \log|t-i - t-j| - K\sum V(t-i) + \sum \sqrt{t-i} \leq (\lambda M - 0 - I(\varrho))K^2 + (120 + \sqrt{2})h + 2h \log K$. Proof: regularise by circle measures of radius $\varepsilon = 1/(4K^2)$, apply Lemma 6.2 to $\sigma = K^{-1}\sum \omega_i$ and ϱ , the circle average of $\log|z-a|$, the arcsine mass bound $\varrho([x-r, x+r]) \leq 40\sqrt{r}$, and (6.7) (Lean: *Config69*).

lemma *rho-singleton*: $\text{emeasure } \rho \{a\} = 0$
 using *rho-Icc-le*[of 0 a] by *simp*

The cross bound: $\int \ln^+(\varepsilon/|x - u|) d\rho(u) \leq 80 \sqrt{\varepsilon}$, by the layer cake over *rho-Icc-le*.

lemma *cross-bound*:

```

fixes x e :: real
assumes e: 0 < e
shows (∫+ u. ennreal (max 0 (ln (e / |x - u|))) ∂rho) ≤
  ennreal (80 * sqrt e)

```

proof –

```

have cake: ⋀ a :: real. 0 ≤ a ⇒
  ennreal a = (∫+ s. indicator {0..<a} s ∂lborel)
  by (simp add: emeasure-lborel-Ico)
have level: ⋀ s u. 0 ≤ s ⇒ u ≠ x ⇒
  (s < ln (e / |x - u|)) ⟷ |x - u| < e * exp (- s)

```

proof –

```

fix s u :: real
assume s0: 0 ≤ s and ux: u ≠ x
have d0: 0 < |x - u| using ux by simp
have l1: (s < ln (e / |x - u|)) ⟷ exp s < e / |x - u|
  using d0 e ln-less-cancel-iff[of exp s e / |x - u|] by simp
have l2a: e * exp (- s) = e / exp s by (simp add: exp-minus divide-inverse)

```

```

have l2:  $\exp s < e / |x - u| \iff$ 
 $|x - u| < e * \exp (- s)$  unfolding l2a
using d0 pos-less-divide-eq[of  $|x -$ 
 $u| \exp s e$ ] pos-less-divide-eq[of  $\exp s |x - u| e$ ]
by (simp add: mult.commute)
show ( $s < \ln (e / |x - u|)$ )  $\iff |x - u| < e * \exp (- s)$ 
using l1 l2 by (rule trans)
qed
have swap:  $(\int^+ u. \text{ennreal } (\max 0 (\ln (e / |x - u|))) \partial \rho) =$ 
 $(\int^+ s. \text{emeasure } \rho \{u. |x - u| <$ 
 $e * \exp (- s)\} * \text{indicator } \{0.. \} s \partial \text{lborel})$ 
proof -
define f where  $f u = \max 0 (\ln (e / |x - u|))$  for  $u :: \text{real}$ 
have fnn:  $\bigwedge u. 0 \leq f u$  unfolding f-def by simp
have A:  $(\int^+ u. \text{ennreal } (f u) \partial \rho) =$ 
 $(\int^+ u. (\int^+ s. \text{indicator } \{0.. < f u\} s \partial \text{lborel})$ 
 $\partial \rho)$ 
using cake[OF fnn] by simp
have finite-measure rho using rho-univ by (auto simp: rho-def intro: fi-
nite-measureI)
hence ps: pair-sigma-finite rho lborel using sigma-finite-lborel
unfolding pair-sigma-finite-def finite-measure-def by simp
have sr:  $\text{sets } (\rho \otimes_M \text{lborel}) = \text{sets } (\text{borel} \otimes_M \text{borel})$ 
unfolding rho-def using sets-pair-measure-cong by simp
have mf0a:  $(\lambda p :: \text{real} \times \text{real}. \text{indicator } \{p :: \text{real} \times \text{real}. 0 \leq$ 
 $\text{snd } p \wedge \text{snd } p < \max 0 (\ln (e / |x - \text{fst } p|))\} p :: \text{ennreal}) \in$ 
 $\text{borel-measurable } (\text{borel} \otimes_M \text{borel})$ 
by measurable
have mf0e:  $(\lambda p :: \text{real} \times \text{real}. \text{indicator } \{0.. < f (\text{fst } p)\} (\text{snd } p) :: \text{ennreal}) =$ 
 $(\lambda p :: \text{real} \times \text{real}. \text{indicator } \{p :: \text{real} \times \text{real}. 0 \leq \text{snd } p \wedge$ 
 $\text{snd } p < \max 0 (\ln (e / |x - \text{fst } p|))\} p :: \text{ennreal})$ 
unfolding f-def by (simp add: indicator-def)
have mf0:  $(\lambda p :: \text{real} \times$ 
 $\text{real}. \text{indicator } \{0.. < f (\text{fst } p)\} (\text{snd } p) :: \text{ennreal}) \in$ 
 $\text{borel-measurable } (\text{borel} \otimes_M \text{borel})$  unfolding mf0e by (rule mf0a)
have mf:  $(\lambda (u, s). \text{indicator } \{0.. < f u\} s :: \text{ennreal}) \in$ 
 $\text{borel-measurable } (\rho \otimes_M \text{lborel})$  unfolding split-beta'
using mf0 measurable-cong-sets[OF sr refl] by auto
have B:  $(\int^+ u. (\int^+ s. \text{indicator } \{0.. < f u\} s \partial \text{lborel})$ 
 $\partial \rho) = (\int^+ s. (\int^+ u. \text{indicator } \{0.. < f u\} s$ 
 $\partial \rho) \partial \text{lborel})$ 
using pair-sigma-finite.Fubini'[OF ps mf] by simp
have C:  $\bigwedge s. (\int^+ u. \text{indicator } \{0.. < f u\} s \partial \rho) =$ 
 $\text{emeasure } \rho \{u. |x - u| < e * \exp (- s)\} * \text{indicator } \{0.. \} s$ 
proof -
fix s :: real
show  $(\int^+ u. \text{indicator } \{0.. < f u\} s \partial \rho) =$ 
 $\text{emeasure } \rho \{u. |x - u| < e * \exp (- s)\} * \text{indicator } \{0.. \} s$ 
proof (cases  $0 \leq s$ )

```

```

    case False
    have z:  $\bigwedge u. (\text{indicator } \{0..<f\ u\} \ s :: \text{ennreal}) = 0$  using False by simp
    show ?thesis using False unfolding z by simp
next
case True
have ind:  $\bigwedge u. (\text{indicator } \{0..<f\ u\} \ s :: \text{ennreal}) = \text{indicator } (\{u. |x - u| < e * \exp(-s)\} - \{x\}) \ u$ 
proof -
  fix u :: real
  show  $(\text{indicator } \{0..<f\ u\} \ s :: \text{ennreal}) = \text{indicator } (\{u. |x - u| < e * \exp(-s)\} - \{x\}) \ u$ 
  proof (cases u = x)
    case True
    have fx:  $f\ u = 0$  unfolding f-def True by simp
    show ?thesis unfolding fx using True by simp
  next
  case False
  have  $(s < f\ u) \longleftrightarrow |x - u| < e * \exp(-s)$ 
    unfolding f-def using level[OF  $\langle 0 \leq s \rangle$  False]  $\langle 0 \leq s \rangle$  by linarith
  thus ?thesis using False  $\langle 0 \leq s \rangle$  unfolding indicator-def by simp
qed
qed
have meas2:  $\{u. |x - u| < e * \exp(-s)\} \in \text{sets } \rho$  unfolding rho-def by simp
have null:  $\{x\} \in \text{null-sets } \rho$  using rho-singleton[of x] unfolding rho-def by auto
have meas3:  $\{u. |x - u| < e * \exp(-s)\} - \{x\} \in \text{sets } \rho$  using meas2 null by auto
show ?thesis unfolding ind using True meas3 emeasure-Diff-null-set[OF null meas2] by simp
qed
qed
have  $(\int^+ u. \text{ennreal } (f\ u) \ \partial \rho) = (\int^+ s. \text{emeasure } \rho \ (\{u. |x - u| < e * \exp(-s)\} * \text{indicator } \{0..\} \ s \ \partial \text{lborel}))$ 
  unfolding A B C ..
  thus ?thesis unfolding f-def .
qed
have ball:  $\bigwedge s. \text{emeasure } \rho \ (\{u. |x - u| < e * \exp(-s)\}) \leq \text{ennreal } (40 * \sqrt{e * \exp(-s)})$ 
proof -
  fix s :: real
  have r0:  $0 \leq e * \exp(-s)$  using e by simp
  have sub:  $\{u. |x - u| < e * \exp(-s)\} \subseteq \{x - e * \exp(-s)..x + e * \exp(-s)\}$  by auto
  have meas:  $\{x - e * \exp(-s)..x + e * \exp(-s)\} \in \text{sets } \rho$  unfolding rho-def by simp

```

```

show emeasure rho {u. |x - u| < e * exp (- s)} ≤
  ennreal (40 * sqrt (e * exp (- s)))
  using emeasure-mono[OF sub meas] rho-Icc-le[OF r0, of x] by (rule or-
der-trans)
qed
have tail: (∫+ s. ennreal (40 * sqrt (e * exp (- s))) *
  indicator {0..} s ∂lborel) = ennreal (80 * sqrt e)
proof -
  have sq: ∧s. sqrt (e * exp (- s)) = sqrt e * exp (- s / 2)
  proof -
    fix s :: real
    have ee: exp (- s) = exp (- s / 2) * exp (- s / 2) by (simp add:
exp-add[symmetric])
    show sqrt (e * exp (- s)) = sqrt e * exp (- s / 2) unfolding ee
    using real-sqrt-mult[of e exp (- s / 2) * exp (- s / 2)] by simp
  qed
have m: (λs. 40 * (sqrt e * exp (- s / 2))) ∈ borel-measurable borel by simp
have d: ∧s. 0 ≤ s ⇒
  ((λs. - 80 * sqrt e * exp (- s / 2)) has-real-derivative 40 * (sqrt e * exp (-
s /
2))) (at s) by (force intro: derivative-eq-intros)
have n: ∧s. 0 ≤ s ⇒ 0 ≤ 40 * (sqrt e * exp (- s / 2)) using e
by simp
have h1: filterlim (λs::real. 1 / 2 * s) at-top at-top
  using filterlim-tendsto-pos-mult-at-top[OF tendsto-const - filterlim-ident,
of 1 / 2 :: real] by simp
have h2: ((λy::real. exp (- y)) → 0) at-top
  by (intro filterlim-compose[OF exp-at-bot] filterlim-uminus-at-bot-at-top)
have h3: ((λs::real. exp (- (1 / 2 * s))) → 0) at-top
  using filterlim-compose[OF h2 h1] by (simp add: o-def)
have l: ((λs. - 80 * sqrt e * exp (- s / 2)) → 0) at-top
  using tendsto-mult-right-zero[OF h3, of - 80 * sqrt e] by simp
have ftc: (∫+ s. ennreal (40 * (sqrt e * exp (- s / 2))) *
  indicator {0..} s ∂lborel) = ennreal (80 * sqrt e)
  using nn-integral-FTC-atLeast[where f = λs. 40 * (sqrt e * exp (- s / 2))]
and F =
  λs. - 80 * sqrt e * exp (- s / 2) and a = 0 and T = 0, OF m d n l] by
simp
show ?thesis unfolding sq using ftc .
qed
have mono: (∫+ s. emeasure rho {u. |x - u| <
  e * exp (- s)} * indicator {0..} s ∂lborel) ≤
  (∫+ s. ennreal (40 * sqrt (e * exp (- s))) *
  indicator {0..} s ∂lborel)
  using ball by (force intro: nn-integral-mono mult-right-mono)
show ?thesis
  unfolding swap using mono tail by simp
qed

```

Helpers for (6.9): the normalised arclength density on $[0, 2\pi]$ and the circle

measures.

abbreviation *cden* :: *real measure* **where**

cden $\equiv$ *density lborel* ($\lambda \vartheta. \text{ennreal} (\text{indicator } \{0..2 * \pi\} \vartheta / (2 * \pi))$)

lemma *cden-univ*: *emeasure cden UNIV* = 1

proof –

have *emeasure cden UNIV* = $(\int^+ \vartheta. \text{ennreal} (\text{indicator } \{0..2 * \pi\} \vartheta / (2 * \pi)) \partial \text{lborel})$

by (*simp add: emeasure-density*)

also have $\dots = (\int^+ \vartheta. \text{ennreal} (1 / (2 * \pi)) * \text{indicator } \{0..2 * \pi\} \vartheta \partial \text{lborel})$

by (*auto simp: indicator-def intro: nn-integral-cong*)

also have $\dots = \text{ennreal} (1 / (2 * \pi)) * \text{ennreal} (2 * \pi)$

by (*simp add: nn-integral-cmult-indicator*)

also have $\dots = 1$

by (*simp add: ennreal-mult[symmetric]*)

finally show *?thesis* .

qed

lemma *cden-finite*: *finite-measure cden*

using *cden-univ* **by** (*auto intro: finite-measureI*)

lemma *circ-meas*: $(\lambda \vartheta :: \text{real}. c + \text{of-real } \varepsilon * \text{cis } \vartheta) \in$

borel-measurable borel

by (*fast intro: borel-measurable-continuous-onI continuous-intros*)

lemma *circ-meas'*: $(\lambda \vartheta :: \text{real}. c + \text{of-real } \varepsilon * \text{cis } \vartheta) \in$

measurable cden borel

using *circ-meas[of c ε]* **by** *simp*

lemma *sets-circ*: *sets (circ c ε)* = *sets borel*

unfolding *circ-def* **by** *simp*

lemma *space-circ*: *space (circ c ε)* = *UNIV*

unfolding *circ-def* **by** *simp*

lemma *circ-univ*: *emeasure (circ c ε) UNIV* = 1

unfolding *circ-def* **using** *emeasure-distr[OF circ-meas'[of c ε], of UNIV]* *cden-univ* **by** *simp*

lemma *circ-finite*: *finite-measure (circ c ε)*

using *circ-univ[of c ε]* *space-circ[of c ε]* **by** (*auto intro: finite-measureI*)

lemma *circ-AE*: *AE z in circ c ε . norm (z - c) = $|\varepsilon|$*

proof –

have *P*: $\{z \in \text{space borel}. \text{norm } (z - c) = |\varepsilon|\} \in$

sets (borel :: complex measure)

by *measurable*

have $pt: AE \ \vartheta \text{ in cden. } norm \ (c + of\text{-real } \varepsilon * cis \ \vartheta - c) =$
 $|\varepsilon|$
by (*simp add: norm-mult*)
show ?thesis **unfolding** *circ-def* **using** *AE-distr-iff*[*OF circ-meas'*[*of c ε*] *P*] *pt*
by *simp*
qed

lemma *circ-integrable-bound*:
fixes $f :: complex \Rightarrow real$ **and** $B :: real$
assumes $m: f \in borel\text{-measurable } borel$ **and** $b: \bigwedge z. norm \ (z - c) =$
 $|\varepsilon| \Rightarrow |f \ z| \leq B$
shows *integrable* (*circ c ε*) *f*
proof –
have $m': f \in borel\text{-measurable } (circ \ c \ \varepsilon)$
using *m measurable-cong-sets*[*OF sets-circ*[*of c ε*] *refl*]
by *auto*
have $ae: AE \ z \text{ in } circ \ c \ \varepsilon. norm \ (f \ z) \leq B$ **using** *circ-AE*[*of c ε*] *b*
by (*auto elim: eventually-mono*)
show ?thesis **using** *finite-measure.integrable-const-bound*[*OF circ-finite ae m'*].
qed

lemma *cis-eq-null*: $\{\vartheta :: real. cis \ \vartheta = w\} \in null\text{-sets } lborel$
proof –
have $bad: \{\vartheta. cis \ \vartheta = w\} \subseteq$
 $range \ (\lambda n::int. Arg \ w + 2 * pi * of\text{-int } n)$
proof
fix ϑ **assume** $th: \vartheta \in \{\vartheta. cis \ \vartheta = w\}$
have $w0: w \neq 0$ **using** *th*
by *auto*
have $cw: cis \ (Arg \ w) = w$ **using** *cis-Arg*[*OF w0*] *th*
by (*force simp: sgn-div-norm*)
have $\sin \ \vartheta = \sin \ (Arg \ w) \wedge \cos \ \vartheta = \cos \ (Arg \ w)$ **using** *th cw*
by (*simp add: complex-eq-iff*)
then obtain $n :: int$ **where** $\vartheta = Arg \ w + 2 * pi * n$ **using** *sin-cos-eq-iff* **by**
blast
thus $\vartheta \in range \ (\lambda n::int. Arg \ w + 2 * pi * of\text{-int } n)$ **by** *auto*
qed
show ?thesis **using** *countable-imp-null-set-lborel countable-subset*[*OF bad*]
by *fast*
qed

lemma *circ-log-integrable*:
assumes $e: 0 < \varepsilon$
shows *integrable* (*circ c ε*) $(\lambda z. \ln \ (norm \ (z - a)))$
proof –
define w **where** $w = (a - c) / of\text{-real } \varepsilon$
have $mF: (\lambda z :: complex. \ln \ (norm \ (z - a))) \in borel\text{-measurable } borel$
by *simp*
have $mh: (\lambda \vartheta. \ln \ (norm \ (c + of\text{-real } \varepsilon * cis \ \vartheta - a))) \in$


```

    borel-measurable borel
  using measurable-compose[OF circ-meas[of c ε] mF] by (simp add: o-def)
  have n2: (λϑ. norm (cis ϑ - w)) ∈ borel-measurable borel
  by (fast intro: borel-measurable-continuous-onI continuous-intros)
  have mk: (λϑ. ln ε + ln (norm (cis ϑ - w))) ∈
    borel-measurable borel using borel-measurable-ln[OF n2]
  by (auto intro: borel-measurable-add)
  have pw: ln (norm (c + of-real ε * cis ϑ - a)) =
    ln ε + ln (norm (cis ϑ - w)) if nw: cis ϑ ≠ w for ϑ
  proof -
    have eq: c + of-real ε * cis ϑ - a = of-real ε * (cis ϑ - w)
    using e unfolding w-def
    by (simp add: field-simps)
    show ?thesis unfolding eq norm-mult using e nw ln-mult[of ε norm (cis ϑ -
w)]
    by simp
  qed
  have ae: AE ϑ ∈ einterval (min (ereal 0) (ereal (2 * pi))) (max (ereal 0) (ereal
(2 *
pi))) in lborel. ln (norm (c + of-real ε * cis ϑ - a)) =
    ln ε + ln (norm (cis ϑ - w))
  using AE-I'[OF cis-eq-null[of w], of λϑ. ϑ ∈
    einterval (min (ereal 0) (ereal (2 * pi))) (max (ereal 0) (ereal (2 * pi))) →
    ln (norm (c + of-real ε * cis ϑ - a)) =
    ln ε + ln (norm (cis ϑ - w))] pw
  by blast
  have ii0: interval-lebesgue-integrable lborel (ereal 0) (ereal (2 *
pi)) (λϑ. ln ε + ln (norm (cis ϑ - w)))
  using interval-lebesgue-integral-add(1)[OF interval-integral-const(1) circle-log-integrable[of
w]] .
  have interval-lebesgue-integrable lborel (ereal 0) (ereal (2 *
pi)) (λϑ. ln (norm (c + of-real ε * cis ϑ - a)))
  using interval-lebesgue-integrable-cong-AE[of λϑ. ln (norm (c +
of-real ε * cis ϑ - a)) λϑ. ln ε +
    ln (norm (cis ϑ - w)) ereal 0 ereal (2 * pi)] mh mk ae ii0 by simp
  hence set-integrable lborel {0 <..< 2 * pi} (λϑ. ln (norm (c + of-real ε *
cis ϑ - a)))
  unfolding interval-lebesgue-integrable-def by simp
  hence set-integrable lborel {0..2 * pi} (λϑ. ln (norm (c + of-real ε *
cis ϑ - a)))
  using set-integrable-discrete-difference[where X = {0, 2 * pi} and M = lborel
and A =
    {0..2 * pi} and B = {0 <..< 2 * pi} and f =
    λϑ. ln (norm (c + of-real ε * cis ϑ - a))]
  by force
  hence i1a: integrable lborel (λϑ. (1 / (2 * pi)) * (indicator {0..2 *
pi} ϑ *R ln (norm (c + of-real ε * cis ϑ - a))))
  unfolding set-integrable-def by simp
  have eqf: (λϑ. (indicator {0..2 * pi} ϑ / (2 * pi)) *R ln (norm (c +

```

```

    of-real  $\varepsilon * \text{cis } \vartheta - a$ ))) =
    ( $\lambda \vartheta. (1 / (2 * \text{pi})) * (\text{indicator } \{0..2 * \text{pi}\} \vartheta *_R \ln (\text{norm } (c +$ 
    of-real  $\varepsilon * \text{cis } \vartheta - a))$ )))
  by (simp add: fun-eq-iff)
have i1: integrable lborel ( $\lambda \vartheta. (\text{indicator } \{0..2 * \text{pi}\} \vartheta / (2 * \text{pi})) *_R \ln (\text{norm } (c + \text{of-real } \varepsilon * \text{cis } \vartheta - a))$ ))
  unfolding eqf by (rule i1a)
have mg: ( $\lambda \vartheta :: \text{real}. \text{indicator } \{0..2 * \text{pi}\} \vartheta / (2 * \text{pi})$ )  $\in$ 
  borel-measurable lborel
  by simp
have i2: integrable cden ( $\lambda \vartheta. \ln (\text{norm } (c + \text{of-real } \varepsilon * \text{cis } \vartheta - a))$ ))
  using integrable-density[of  $\lambda \vartheta. \ln (\text{norm } (c + \text{of-real } \varepsilon * \text{cis } \vartheta - a))$  lborel  $\lambda \vartheta. \text{indicator } \{0..2 * \text{pi}\} \vartheta / (2 * \text{pi})$ ] mh mg i1
  by simp
show ?thesis unfolding circ-def using integrable-distr-eq[OF circ-meas'[of  $c \ \varepsilon$ ] mF] i2
  by simp
qed

```

lemma *circ-singleton*:

```

  assumes  $e: 0 < \varepsilon$ 
  shows  $\text{emeasure } (\text{circ } c \ \varepsilon) \{a\} = 0$ 
proof -
  define  $w$  where  $w = (a - c) / \text{of-real } \varepsilon$ 
  have pre: ( $\lambda \vartheta. c + \text{of-real } \varepsilon * \text{cis } \vartheta$ ) - '  $\{a\} \cap \text{space cden} = \{\vartheta. \text{cis } \vartheta = w\}$ 
    unfolding w-def using e by (auto simp: field-simps)
  have  $\{\vartheta :: \text{real}. \text{cis } \vartheta = w\} \in \text{sets lborel}$  using cis-eq-null[of  $w$ ]
    by (simp add: null-sets-def)
  hence d1:  $\text{emeasure cden } \{\vartheta. \text{cis } \vartheta = w\} =$ 
    ( $\int^+ \vartheta. \text{ennreal } (\text{indicator } \{0..2 * \text{pi}\} \vartheta / (2 * \text{pi})) * \text{indicator } \{\vartheta. \text{cis } \vartheta = w\} \vartheta \ \partial \text{lborel}$ )
    by (simp add: emeasure-density)
  have d2: ( $\int^+ \vartheta. \text{ennreal } (\text{indicator } \{0..2 * \text{pi}\} \vartheta / (2 * \text{pi})) * \text{indicator } \{\vartheta. \text{cis } \vartheta = w\} \vartheta \ \partial \text{lborel}$ ) = 0
    by (rule nn-integral-null-set[OF cis-eq-null[of  $w$ ]])
  have d:  $\text{emeasure cden } \{\vartheta. \text{cis } \vartheta = w\} = 0$ 
    using d1 d2 by simp
  show ?thesis unfolding circ-def using emeasure-distr[OF circ-meas'[of  $c \ \varepsilon$ ], of  $\{a\}$ ] pre d
    by simp
qed

```

lemma *circ-neg-log*:

```

  fixes  $c \ z :: \text{complex}$  and  $\varepsilon \ R :: \text{real}$ 
  assumes  $e: 0 < \varepsilon$  and  $hc: \text{norm } c + \varepsilon \leq R$  and  $hz: \text{norm } z \leq R$ 
  shows ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \ \partial \text{circ } c \ \varepsilon$ )  $\leq$ 

```

```

    ennreal (2 * R - ln ε)
  proof -
    define g where g w = ln (norm (w - z)) for w :: complex
    have ig: integrable (circ c ε) g unfolding g-def
      using circ-log-integrable[OF e, of c z] .
    have ipos: integrable (circ c ε) (λw. max 0 (g w)) using ig
      using integrable-max by simp
    have ineg: integrable (circ c ε) (λw. max 0 (- g w)) using ig
      using integrable-max by simp
    have e1: (∫+ w. ennreal (- ln (norm (z - w))) ∂circ c ε) =
      (∫+ w. ennreal (max 0 (- g w)) ∂circ c ε)
      unfolding g-def by (simp add: norm-minus-commute ennreal-max-0)
    have e2: (∫+ w. ennreal (max 0 (- g w)) ∂circ c ε) =
      ennreal (integralL (circ c ε) (λw. max 0 (- g w)))
      using nn-integral-eq-integral[OF ineg] by simp
    have e3: integralL (circ c ε) (λw. max 0 (- g w)) =
      integralL (circ c ε) (λw. max 0 (g w)) -
      integralL (circ c ε) g
  proof -
    have pt: (λw. max 0 (- g w)) = (λw. max 0 (g w) - g w)
      by (simp add: fun-eq-iff max-def)
    show ?thesis unfolding pt using Bochner-Integration.integral-diff[OF ipos ig]
  .

qed
have h1: ln ε ≤ integralL (circ c ε) g unfolding g-def
  using circ-log[OF e, of c z] e
  by simp
have R0: 0 ≤ R using hc e norm-ge-zero[of c]
  by linarith
have hb: AE w in circ c ε. max 0 (g w) ≤ 2 * R
proof -
  have pt: max 0 (g w) ≤ 2 * R if hw: norm (w - c) = |ε| for w
  proof -
    have norm w ≤ R using hw hc e norm-triangle-ineq[of w - c c]
      by simp
    hence w2: norm (w - z) ≤ 2 * R using hz norm-triangle-ineq4[of w z]
      by simp
    have w3: g w ≤ norm (w - z) unfolding g-def
      by (metis less-eq-real-def ln-0 ln-less-self norm-ge-zero)
    show ?thesis using w2 w3 R0
      by simp
  proof -
    qed
    show ?thesis using circ-AE[of c ε] pt
      by (auto elim: eventually-mono)
  qed
qed
have h2: integralL (circ c ε) (λw. max 0 (g w)) ≤ 2 * R
proof -
  have integralL (circ c ε) (λw. max 0 (g w)) ≤
    integralL (circ c ε) (λw. 2 * R)

```

```

    using integral-mono-AE[OF ipos - hb] finite-measure.integrable-const[OF
circ-finite]
    by auto
    also have ... = 2 * R using circ-univ[of c ε] space-circ[of c ε]
    by (simp add: measure-def)
    finally show ?thesis .
qed
show ?thesis unfolding e1 e2 e3 using h1 h2
using ennreal-leI by simp
qed

```

Finite sums of circle measures, realised as a push-forward of *count-space* $I \otimes_M cden$.

lemma *csum-meas*:

```

fixes c :: nat ⇒ complex and I :: nat set
assumes I: countable I
shows (λ(i, ϑ). c i + of-real ε * cis ϑ) ∈
  measurable (count-space I ⊗M cden) borel
using measurable-pair-measure-countable1[OF I, of λ(i,
  ϑ). c i + of-real ε * cis ϑ cden borel] circ-meas'
by simp

```

lemma *csum-nn*:

```

fixes c :: nat ⇒ complex and I :: nat set and f :: complex ⇒ ennreal
assumes I: finite I and f: f ∈ borel-measurable borel
shows (∫+ z. f z ∂distr (count-space I ⊗M cden) borel
  (λ(i, ϑ). c i + of-real ε * cis ϑ))
  = (∑ i ∈ I. ∫+ z. f z ∂circ (c i) ε)

```

proof –

```

have m: (λ(i, ϑ). c i + of-real ε * cis ϑ) ∈
  measurable (count-space I ⊗M cden) borel
  using csum-meas[OF countable-finite[OF I]] .
have sf: sigma-finite-measure cden using cden-finite
  by (simp add: finite-measure-def)
have mfp: (λp. f ((λ(i,
  ϑ). c i + of-real ε * cis ϑ) p)) ∈
  borel-measurable (count-space I ⊗M cden)
  using measurable-compose[OF m f] by (simp add: o-def)
have e1: (∫+ z. f z ∂distr (count-space I ⊗M cden) borel
  (λ(i, ϑ). c i + of-real ε * cis ϑ))
  = (∫+ p. f ((λ(i,
  ϑ). c i + of-real ε *
  cis ϑ) p) ∂(count-space I ⊗M cden))
  using nn-integral-distr[OF m] f by simp
have e2: (∫+ p. f ((λ(i,
  ϑ). c i + of-real ε *
  cis ϑ) p) ∂(count-space I ⊗M cden))
  = (∫+ i. (∫+ ϑ. f (c i + of-real ε *
  cis ϑ) ∂cdens) ∂count-space I)

```

using *sigma-finite-measure.nn-integral-fst*[*OF sf mfp*] **by** *simp*
have *e4*: $\bigwedge i. (\int^+ \vartheta. f (c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta) \partial \text{cden}) = (\int^+ z. f\ z\ \partial \text{circ } (c\ i) \varepsilon)$
unfolding *circ-def* **using** *nn-integral-distr*[*OF circ-meas*] *f* **by** *simp*
have *e3*: $(\int^+ i. (\int^+ \vartheta. f (c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta) \partial \text{cden}) \partial \text{count-space } I)$
 $= (\sum i \in I. \int^+ z. f\ z\ \partial \text{circ } (c\ i) \varepsilon)$
unfolding *e4* **by** (*rule nn-integral-count-space-finite*[*OF I*])
show *?thesis* **unfolding** *e1 e2 e3 ..*
qed

lemma *csum-emeasure*:

fixes *c* :: *nat* $\Rightarrow$ *complex* **and** *I* :: *nat set*
assumes *I*: *finite I* **and** *A*: *A* $\in$ *sets borel*
shows *emeasure* (*distr* (*count-space I* $\otimes_M$ *cden*) *borel* ($\lambda(i, \vartheta). c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta$)) *A* = $(\sum i \in I. \text{emeasure } (\text{circ } (c\ i) \varepsilon) A)$
using *csum-nn*[*OF I*, **where** *f* = *indicator A* **and** *c* = *c* **and** $\varepsilon = \varepsilon$] *A sets-circ*
by *simp*

lemma *csum-integral*:

fixes *c* :: *nat* $\Rightarrow$ *complex* **and** *I* :: *nat set* **and** *f* :: *complex* $\Rightarrow$ *real*
assumes *I*: *finite I* **and** *f*: *f* $\in$ *borel-measurable borel* **and** *fi*: $\bigwedge i. i \in I \Rightarrow \text{integrable } (\text{circ } (c\ i) \varepsilon) f$
shows *integrable* (*distr* (*count-space I* $\otimes_M$ *cden*) *borel* ($\lambda(i, \vartheta). c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta$)) *f*
and *integral^L* (*distr* (*count-space I* $\otimes_M$ *cden*) *borel* ($\lambda(i, \vartheta). c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta$)) *f* = $(\sum i \in I. \text{integral}^L (\text{circ } (c\ i) \varepsilon) f)$

proof –

have *m*: $(\lambda(i, \vartheta). c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta) \in \text{measurable } (\text{count-space } I \otimes_M \text{cden}) \text{ borel}$
using *csum-meas*[*OF countable-finite*[*OF I*]] .
have *mn*: $(\lambda z. \text{ennreal } (\text{norm } (f\ z))) \in \text{borel-measurable borel}$ **using** *f*
by *simp*
have *fin*: $\bigwedge i. i \in I \Rightarrow (\int^+ z. \text{ennreal } (\text{norm } (f\ z)) \partial \text{circ } (c\ i) \varepsilon) < \infty$
using *fi*
by (*simp add: integrable-iff-bounded*)
have *nnD*: $(\int^+ z. \text{ennreal } (\text{norm } (f\ z)) \partial \text{distr } (\text{count-space } I \otimes_M \text{cden}) \text{ borel } (\lambda(i, \vartheta). c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta)) < \infty$
unfolding *csum-nn*[*OF I mn*] **using** *fin I ennreal-sum-less-top*[*OF I*] **by** *simp*
have *f* $\in$ *borel-measurable* (*distr* (*count-space I* $\otimes_M$ *cden*) *borel* ($\lambda(i, \vartheta). c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta$))
using *f* **by** *simp*
thus *integrable* (*distr* (*count-space I* $\otimes_M$ *cden*) *borel* ($\lambda(i, \vartheta). c\ i + \text{of-real } \varepsilon * \text{cis } \vartheta$)) *f*
using *nnD* **by** (*simp add: integrable-iff-bounded*)

```

hence  $iP$ : integrable (count-space  $I \otimes_M cden$ ) ( $\lambda p. f ((\lambda(i, \vartheta). c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta)\ p))$ 
using integrable-distr-eq[OF m f] by simp
have  $ps$ : pair-sigma-finite (count-space  $I$ ) cden
using sigma-finite-measure-count-space-finite[OF I] cden-finite
unfolding pair-sigma-finite-def by (simp add: finite-measure-def)
have  $e1$ :  $integral^L (distr (count\text{-}space\ I \otimes_M cden)\ borel\ (\lambda(i, \vartheta). c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta))\ f$ 
 $= integral^L (count\text{-}space\ I \otimes_M cden)\ (\lambda p. f ((\lambda(i, \vartheta). c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta)\ p))$ 
by (rule integral-distr[OF m f])
have  $e2$ :  $integral^L (count\text{-}space\ I \otimes_M cden)\ (\lambda p. f ((\lambda(i, \vartheta). c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta)\ p))$ 
 $= (\int i. (\int \vartheta. f (c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta)\ \partial cden)\ \partial count\text{-}space\ I)$ 
using pair-sigma-finite.integral-fst'[OF ps iP] by simp
have  $e4$ :  $\bigwedge i. (\int \vartheta. f (c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta)\ \partial cden) = integral^L (circ\ (c\ i)\ \varepsilon)\ f$ 
unfolding circ-def using integral-distr[OF circ-meas' f] by simp
have  $e3$ :  $(\int i. (\int \vartheta. f (c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta)\ \partial cden)\ \partial count\text{-}space\ I)$ 
 $= (\sum i \in I. integral^L (circ\ (c\ i)\ \varepsilon)\ f)$ 
unfolding  $e4$  by (rule lebesgue-integral-count-space-finite[OF I])
show  $integral^L (distr (count\text{-}space\ I \otimes_M cden)\ borel\ (\lambda(i, \vartheta). c\ i + of\text{-}real\ \varepsilon * cis\ \vartheta))\ f = (\sum i \in I. integral^L (circ\ (c\ i)\ \varepsilon)\ f)$ 
unfolding  $e1\ e2\ e3\ ..$ 
qed

```

A scaled measure is a density with a constant; the Bochner integral scales.

lemma *scale-density*:

```

fixes  $M :: 'a\ measure$  and  $r :: real$ 
shows scale-measure (ennreal  $r$ )  $M = density\ M\ (\lambda\cdot. ennreal\ r)$ 
proof (rule measure-eqI)
show sets (scale-measure (ennreal  $r$ )  $M$ ) = sets (density  $M\ (\lambda\cdot. ennreal\ r)$ )
by simp
fix  $A$  assume  $A: A \in sets\ (scale\text{-}measure\ (ennreal\ r)\ M)$ 
show emeasure (scale-measure (ennreal  $r$ )  $M$ )  $A = emeasure\ (density\ M\ (\lambda\cdot. ennreal\ r))\ A$ 
using  $A$  by (simp add: emeasure-density nn-integral-cmult-indicator)
qed

```

lemma *scale-integral*:

```

fixes  $M :: 'a\ measure$  and  $f :: 'a \Rightarrow real$  and  $r :: real$ 
assumes  $r: 0 \leq r$  and  $f: f \in borel\text{-}measurable\ M$ 
shows  $integral^L (scale\text{-}measure\ (ennreal\ r)\ M)\ f = r * integral^L\ M\ f$ 
and integrable  $M\ f \implies integrable\ (scale\text{-}measure\ (ennreal\ r)\ M)\ f$ 
proof –
have  $g: (\lambda\cdot. r) \in borel\text{-}measurable\ M$ 

```

```

    by simp
  have nn:  $\text{AE } x \text{ in } M. 0 \leq r$  using r
    by simp
  show  $\text{integral}^L (\text{scale-measure } (\text{ennreal } r) M) f = r * \text{integral}^L M f$ 
    unfolding scale-density
    using integral-density[OF f g nn] by simp
  show  $\text{integrable } M f \implies \text{integrable } (\text{scale-measure } (\text{ennreal } r) M) f$ 
    unfolding scale-density
    using integrable-density[OF f g nn] by simp
qed

```

Facts about ρ and its image on $\mathbb{C}$.

```

lemma rho-finite: finite-measure rho
  using rho-univ by (auto simp: rho-def intro: finite-measureI)

```

```

lemma rhoC-meas:  $(\text{of-real} :: \text{real} \Rightarrow \text{complex}) \in \text{measurable } \rho \text{ borel}$ 
  unfolding rho-def by simp

```

```

lemma sets-rhoC:  $\text{sets } (\text{distr } \rho \text{ borel } (\text{of-real} :: \text{real} \Rightarrow \text{complex})) = \text{sets borel}$ 
  by simp

```

```

lemma rhoC-finite: finite-measure (distr rho borel (of-real :: real  $\Rightarrow$  complex))
  using finite-measure.finite-measure-distr[OF rho-finite rhoC-meas] .

```

```

lemma rhoC-singleton:  $\text{emeasure } (\text{distr } \rho \text{ borel } (\text{of-real} :: \text{real} \Rightarrow \text{complex})) \{a\} = 0$ 

```

proof –

```

  have sub:  $(\text{of-real} :: \text{real} \Rightarrow \text{complex}) - ' \{a\} \cap \text{space } \rho \subseteq \{\text{Re } a\}$ 
    by auto
  have sm:  $\{\text{Re } a\} \in \text{sets } \rho$  unfolding rho-def
    by simp
  have le:  $\text{emeasure } \rho ((\text{of-real} :: \text{real} \Rightarrow \text{complex}) - ' \{a\} \cap \text{space } \rho) \leq \text{emeasure } \rho \{\text{Re } a\}$ 
    by (rule emeasure-mono[OF sub sm])
  show ?thesis using emeasure-distr[OF rhoC-meas, of {a}] le rho-singleton[of Re a]
    by simp
qed

```

```

lemma rhoC-AE:  $\text{AE } w \text{ in } \text{distr } \rho \text{ borel } (\text{of-real} :: \text{real} \Rightarrow \text{complex}). \text{norm } w \leq 2$ 

```

proof –

```

  have P:  $\{w \in \text{space borel}. \text{norm } w \leq (2 :: \text{real})\} \in \text{sets } (\text{borel} :: \text{complex measure})$ 
    by measurable
  have nul:  $-\{0 <..<2 :: \text{real}\} \in \text{null-sets } \rho$  using rho-compl unfolding rho-def
    by auto
  have ae:  $\text{AE } u \text{ in } \rho. \text{norm } (\text{of-real } u :: \text{complex}) \leq 2$ 
    using AE-I'[OF nul, of  $\lambda u. \text{norm } (\text{of-real } u :: \text{complex}) \leq 2]$ 
    by force

```

```

  show ?thesis using AE-distr-iff[OF rhoC-meas P] ae
    by simp
qed

lemma rhoC-neg-log:  $(\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \text{distr } \rho \text{ borel } (\text{of-real} :: \text{real} \Rightarrow \text{complex})) \leq \text{ennreal } 80$ 
proof -
  have mf:  $(\lambda w :: \text{complex}. \text{ennreal } (- \ln (\text{norm } (z - w)))) \in \text{borel-measurable borel}$ 
    by simp
  have e1:  $(\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \text{distr } \rho \text{ borel } (\text{of-real} :: \text{real} \Rightarrow \text{complex})) = (\int^+ u. \text{ennreal } (- \ln (\text{norm } (z - \text{of-real } u))) \partial \rho)$ 
    using nn-integral-distr[OF rhoC-meas] mf by simp
  have nul:  $\{\text{Re } z\} \in \text{null-sets } \rho$  using rho-singleton[of Re z] unfolding rho-def
    by auto
  have ae:  $AE \ u \text{ in } \rho. u \neq \text{Re } z$ 
    using AE-I'[OF nul, of  $\lambda u. u \neq \text{Re } z$ ] by auto
  have pt:  $\text{ennreal } (- \ln (\text{norm } (z - \text{of-real } u))) \leq \text{ennreal } (\max 0 (\ln (1 / |\text{Re } z - u|)))$  if  $u: u \neq \text{Re } z$  for  $u$ 
  proof -
    have d0:  $0 < |\text{Re } z - u|$  using u
      by simp
    have dle:  $|\text{Re } z - u| \leq \text{norm } (z - \text{of-real } u)$ 
      using abs-Re-le-cmod[of z - of-real u]
      by simp
    have l1:  $- \ln (\text{norm } (z - \text{of-real } u)) \leq - \ln |\text{Re } z - u|$ 
      using d0 dle ln-le-cancel-iff[of  $|\text{Re } z - u|$  norm (z - of-real u)]
      by linarith
    have l2:  $- \ln |\text{Re } z - u| = \ln (1 / |\text{Re } z - u|)$  using d0
      by (simp add: ln-div)
    show ?thesis using l1 l2
      using ennreal-leI by simp
  qed
  have e2:  $(\int^+ u. \text{ennreal } (- \ln (\text{norm } (z - \text{of-real } u))) \partial \rho) \leq (\int^+ u. \text{ennreal } (\max 0 (\ln (1 / |\text{Re } z - u|))) \partial \rho)$ 
    using ae pt by (auto intro: nn-integral-mono-AE elim: eventually-mono)
  have e3:  $(\int^+ u. \text{ennreal } (\max 0 (\ln (1 / |\text{Re } z - u|))) \partial \rho) \leq \text{ennreal } 80$ 
    using cross-bound[of 1 Re z] by simp
  show ?thesis unfolding e1 using e2 e3 by (rule order-trans)
qed

```

Integrability of $\log|z - w|$ against a product of finite measures carried by a ball.

lemma *ennreal-abs-ln-le:*

fixes $y :: \text{real}$

assumes $y: 0 \leq y$

shows $\text{ennreal } |\ln y| \leq \text{ennreal } y + \text{ennreal } (- \ln y)$

proof –
 have $\ln y \leq y$ **using** y
 by (*metis* *assms* *ln-0* *ln-bound* *less-eq-real-def*)
 hence $|\ln y| \leq y + \max 0 (-\ln y)$ **using** y
 by *simp*
 hence $\text{ennreal } |\ln y| \leq \text{ennreal } (y + \max 0 (-\ln y))$ **by** (*rule* *ennreal-leI*)
 also have $\dots = \text{ennreal } y + \text{ennreal } (-\ln y)$ **using** y
 by (*simp* *add: ennreal-plus* *ennreal-max-0*)
 finally **show** *?thesis* .
qed

lemma *integrable-log-prod*:

fixes $\mu \nu :: \text{complex measure}$ **and** $R C :: \text{real}$
assumes *fm*: *finite-measure* μ **and** *fn*: *finite-measure* ν **and** *sm*: *sets* $\mu =$
sets borel **and** *sn*: *sets* $\nu = \text{sets borel}$
and *aem*: $\text{AE } z \text{ in } \mu. \text{norm } z \leq R$ **and** *aen*: $\text{AE } w \text{ in } \nu. \text{norm } w \leq R$
and *lg*: $\bigwedge z. \text{norm } z \leq R \implies$
 $(\int^+ w. \text{ennreal } (-\ln (\text{norm } (z - w))) \partial \nu) \leq \text{ennreal } C$
shows *integrable* $(\mu \otimes_M \nu) (\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p)))$

proof –
 have *sp*: *sets* $(\mu \otimes_M \nu) = \text{sets } (\text{borel } \otimes_M \text{borel})$ **using** *sm sn*
using *sets-pair-measure-cong* **by** *simp*
 have $(\lambda p :: \text{complex} \times \text{complex}. \ln (\text{norm } (\text{fst } p - \text{snd } p))) \in$
borel-measurable $(\text{borel } \otimes_M \text{borel})$
by *simp*
 hence *m*: $(\lambda p. \ln (\text{norm } (\text{fst } p - \text{snd } p))) \in$
borel-measurable $(\mu \otimes_M \nu)$ **using** *measurable-cong-sets[OF sp refl]*
by *blast*
 have *sfn*: *sigma-finite-measure* ν **using** *fn*
by (*simp* *add: finite-measure-def*)
 have *ma*: $(\lambda p. \text{ennreal } (\text{norm } (\ln (\text{norm } (\text{fst } p - \text{snd } p))))) \in$
borel-measurable $(\mu \otimes_M \nu)$ **using** *m*
by *simp*
 have *fub*: $(\int^+ p. \text{ennreal } (\text{norm } (\ln (\text{norm } (\text{fst } p - \text{snd } p))))) \partial (\mu \otimes_M \nu) =$
 $(\int^+ z. (\int^+ w. \text{ennreal } (\text{norm } (\ln (\text{norm } (z - w))))) \partial \nu) \partial \mu$
using *sigma-finite-measure.nn-integral-fst[OF sfn ma]* **by** *simp*
define *B* **where** $B = \text{ennreal } (2 * R) * \text{emeasure } \nu (\text{space } \nu) + \text{ennreal } C$
 have *inner*: $\bigwedge z. \text{norm } z \leq R \implies$
 $(\int^+ w. \text{ennreal } (\text{norm } (\ln (\text{norm } (z - w))))) \partial \nu \leq B$
proof –
fix $z :: \text{complex}$ **assume** *hz*: $\text{norm } z \leq R$
 have *mw1*: $(\lambda w. \text{ennreal } (\text{norm } (z - w))) \in \text{borel-measurable } \nu$
unfolding *measurable-cong-sets[OF sn refl]*
by *simp*
 have *mw2*: $(\lambda w. \text{ennreal } (-\ln (\text{norm } (z - w)))) \in \text{borel-measurable } \nu$
unfolding *measurable-cong-sets[OF sn refl]*
by *simp*

```

have s1: (∫+ w. ennreal (norm (ln (norm (z - w)))) ∂ν) ≤
  (∫+ w. ennreal (norm (z - w)) + ennreal (- ln (norm (z -
  w)))) ∂ν)
using ennreal-abs-ln-le by (auto intro: nn-integral-mono)
have s2: (∫+ w. ennreal (norm (z - w)) + ennreal (- ln (norm (z -
  w)))) ∂ν = (∫+ w. ennreal (norm (z - w)) ∂ν) +
  (∫+ w. ennreal (- ln (norm (z - w))) ∂ν)
by (rule nn-integral-add[OF mw1 mw2])
have a: AE w in ν. ennreal (norm (z - w)) ≤ ennreal (2 * R)
proof -
  have pt: ennreal (norm (z - w)) ≤ ennreal (2 * R) if w: norm w ≤ R for w
    using w hz norm-triangle-ineq4[of z w] using ennreal-leI by simp
  show ?thesis using aen pt
    by (auto elim: eventually-mono)
qed
have s3: (∫+ w. ennreal (norm (z - w)) ∂ν) ≤
  ennreal (2 * R) * emeasure ν (space ν)
  using nn-integral-mono-AE[OF a] by simp
show (∫+ w. ennreal (norm (ln (norm (z - w)))) ∂ν) ≤ B
  unfolding B-def using s1 s2 s3 lg[OF hz]
  by (auto intro: add-mono order-trans)
qed
have outer: (∫+ z. (∫+ w. ennreal (norm (ln (norm (z -
  w)))) ∂ν) ∂μ) ≤ (∫+ z. B ∂μ)
  using aem inner by (fast intro: nn-integral-mono-AE elim: eventually-mono)
have B < ∞ unfolding B-def
  using finite-measure.emeasure-finite[OF fn, of space ν]
  by (simp add: ennreal-mult-less-top less-top)
hence fin: (∫+ z. B ∂μ) < ∞
  using finite-measure.emeasure-finite[OF fm, of space μ]
  by (simp add: ennreal-mult-less-top less-top)
show ?thesis unfolding integrable-iff-bounded using m fub outer fin
  by simp
qed

```

The sum of two measures with the Borel sets, as *measure-of*.

lemma *summ-sets*:

```

fixes μ ν :: 'a::topological-space measure
shows sets (measure-of UNIV (sets borel) (λA. emeasure μ A + emeasure ν A))
=
  sets (borel :: 'a measure)
using sets.sigma-sets-eq[of borel] sets-measure-of[of sets borel UNIV] by simp

```

lemma *summ-emeasure*:

```

fixes μ ν :: 'a::topological-space measure
assumes s1: sets μ = sets borel and s2: sets ν = sets borel and A: A ∈
  sets borel
shows emeasure (measure-of UNIV (sets borel) (λA. emeasure μ A +
  emeasure ν A)) A = emeasure μ A + emeasure ν A

```

```

proof (rule emeasure-measure-of-sigma)
  show sigma-algebra UNIV (sets borel) using sets.sigma-algebra-axioms[of borel]
    by simp
  show positive (sets borel) ( $\lambda A. \text{emeasure } \mu A + \text{emeasure } \nu A$ )
    unfolding positive-def
    by simp
  show countably-additive (sets borel) ( $\lambda A. \text{emeasure } \mu A + \text{emeasure } \nu A$ )
    unfolding countably-additive-def
  proof (intro allI impI)
    fix F :: nat  $\Rightarrow$  'a set
    assume F1: range F  $\subseteq$  sets borel and F2: disjoint-family F and F3:  $\bigcup (\text{range } F) \in \text{sets borel}$ 
    have a1: ( $\sum i. \text{emeasure } \mu (F i) = \text{emeasure } \mu (\bigcup (\text{range } F))$ )
      using suminf-emeasure[of F  $\mu$ ] F1 F2 s1
      by simp
    have a2: ( $\sum i. \text{emeasure } \nu (F i) = \text{emeasure } \nu (\bigcup (\text{range } F))$ )
      using suminf-emeasure[of F  $\nu$ ] F1 F2 s2
      by simp
    show ( $\sum i. \text{emeasure } \mu (F i) + \text{emeasure } \nu (F i) =$ 
       $\text{emeasure } \mu (\bigcup (\text{range } F)) + \text{emeasure } \nu (\bigcup (\text{range } F))$ )
      unfolding a1[symmetric] a2[symmetric]
      using suminf-add[of  $\lambda i. \text{emeasure } \mu (F i)$   $\lambda i. \text{emeasure } \nu (F i)$ ]
      using summableI by simp
    qed
  show A  $\in$  sets borel by (rule A)
qed

lemma summ-nn:
  fixes  $\mu \nu :: 'a::\text{topological-space measure}$  and  $f :: 'a \Rightarrow \text{ennreal}$ 
  assumes s1: sets  $\mu = \text{sets borel}$  and s2: sets  $\nu = \text{sets borel}$  and  $f: f \in \text{borel-measurable borel}$ 
  shows ( $\int^+ z. f z \partial \text{measure-of UNIV (sets borel)} (\lambda A. \text{emeasure } \mu A + \text{emeasure } \nu A) =$ 
    ( $\int^+ z. f z \partial \mu$ ) +
    ( $\int^+ z. f z \partial \nu$ )
  proof -
    define  $\eta$  where  $\eta = \text{measure-of UNIV (sets borel)} (\lambda A. \text{emeasure } \mu A + \text{emeasure } \nu A)$ 
    have s $\eta$ : sets  $\eta = \text{sets borel}$  unfolding  $\eta$ -def by (rule summ-sets)
    have mc:  $\bigwedge g :: 'a \Rightarrow \text{ennreal}. g \in \text{borel-measurable borel} \implies$ 
       $g \in \text{borel-measurable } \eta \wedge g \in \text{borel-measurable } \mu \wedge$ 
       $g \in \text{borel-measurable } \nu$ 
      using s $\eta$  s1 s2 measurable-cong-sets by simp
    have ( $\int^+ z. f z \partial \eta =$ 
      ( $\int^+ z. f z \partial \mu$ ) + ( $\int^+ z. f z \partial \nu$ )
      using f
    proof induction
      case (cong f g)
      show ?case using cong
        by (simp add: fun-eq-iff)

```

```

next
  case (set A)
  show ?case using set summ-emeasure[OF s1 s2 set] s $\eta$  s1 s2 unfolding  $\eta$ -def
    by simp
next
  case (mult u c)
  show ?case using mult mc[OF mult(2)]
    by (simp add: nn-integral-cmult distrib-left)
next
  case (add u v)
  show ?case using add mc
    by (simp add: nn-integral-add)
next
  case (seq U)
  have m:  $\bigwedge i. U\ i \in \text{borel-measurable } \eta \wedge$ 
     $U\ i \in \text{borel-measurable } \mu \wedge U\ i \in \text{borel-measurable } \nu$  using seq mc
    by simp
  have m $\eta$ :  $\bigwedge i. U\ i \in \text{borel-measurable } \eta$  using m by blast
  have m $\mu$ :  $\bigwedge i. U\ i \in \text{borel-measurable } \mu$  using m by blast
  have m $\nu$ :  $\bigwedge i. U\ i \in \text{borel-measurable } \nu$  using m by blast
  have c1:  $(\int^+ z. (\text{SUP } i. U\ i) z\ \partial\eta) =$ 
     $(\text{SUP } i. \int^+ z. U\ i\ z\ \partial\eta)$  unfolding SUP-apply
    by (rule nn-integral-monotone-convergence-SUP[OF  $\langle \text{incseq } U \rangle$  m $\eta$ ])
  have c2:  $(\int^+ z. (\text{SUP } i. U\ i) z\ \partial\mu) =$ 
     $(\text{SUP } i. \int^+ z. U\ i\ z\ \partial\mu)$  unfolding SUP-apply
    by (rule nn-integral-monotone-convergence-SUP[OF  $\langle \text{incseq } U \rangle$  m $\mu$ ])
  have c3:  $(\int^+ z. (\text{SUP } i. U\ i) z\ \partial\nu) =$ 
     $(\text{SUP } i. \int^+ z. U\ i\ z\ \partial\nu)$  unfolding SUP-apply
    by (rule nn-integral-monotone-convergence-SUP[OF  $\langle \text{incseq } U \rangle$  m $\nu$ ])
  have i2:  $\text{incseq } (\lambda i. \int^+ z. U\ i\ z\ \partial\mu)$ 
    using  $\langle \text{incseq } U \rangle$  unfolding incseq-def le-fun-def
    by (auto intro: nn-integral-mono)
  have i3:  $\text{incseq } (\lambda i. \int^+ z. U\ i\ z\ \partial\nu)$ 
    using  $\langle \text{incseq } U \rangle$  unfolding incseq-def le-fun-def
    by (auto intro: nn-integral-mono)
  show ?case unfolding c1 c2 c3 using seq ennreal-SUP-add[OF i2 i3]
    by simp
qed
thus ?thesis unfolding  $\eta$ -def .
qed

```

Logarithmic integrals against ρ .

```

lemma abs-ln-max-le:
  fixes e x :: real
  assumes e:  $0 < e$  and x:  $0 \leq x$ 
  shows  $|\ln (\max e x)| \leq |\ln e| + x$ 
proof (cases  $x \leq e$ )
  case True
  show ?thesis using True x

```

```

    by (simp add: max-def)
next
case False
have  $\ln x \leq x$  using x
  by (metis ln-0 ln-bound less-eq-real-def)
moreover have  $\ln e \leq \ln x$  using e False
  by simp
ultimately show ?thesis using False
  using e by linarith
qed

lemma rho-AE-Ioo: AE u in rho.  $0 < u \wedge u < 2$ 
proof -
  have nul:  $-\{0 < .. < 2 :: \text{real}\} \in \text{null-sets rho}$  using rho-compl unfolding rho-def
    by auto
  show ?thesis using AE-I'[OF nul, of  $\lambda u. 0 < u \wedge u < 2$ ]
    by force
qed

lemma rho-neg-log-le:  $(\int^+ u. \text{ennreal } (-\ln |x - u|) \partial \text{rho}) \leq$ 
   $\text{ennreal } 80$ 
proof -
  have pt:  $\bigwedge u. \text{ennreal } (-\ln |x - u|) =$ 
     $\text{ennreal } (\max 0 (\ln (1 / |x - u|)))$ 
  by (simp add: ln-div ennreal-max-0)
  show ?thesis unfolding pt using cross-bound[of 1 x]
    by simp
qed

lemma rho-log-integrable: integrable rho  $(\lambda u. \ln |x - u|)$ 
proof -
  have m:  $(\lambda u. \ln |x - u|) \in \text{borel-measurable rho}$  unfolding rho-def
    by simp
  have m1:  $(\lambda u. \text{ennreal } |x - u|) \in \text{borel-measurable rho}$  unfolding rho-def
    by simp
  have m2:  $(\lambda u. \text{ennreal } (-\ln |x - u|)) \in \text{borel-measurable rho}$ 
    unfolding rho-def
    by simp
  have s1:  $(\int^+ u. \text{ennreal } (\text{norm } (\ln |x - u|)) \partial \text{rho}) \leq$ 
     $(\int^+ u. \text{ennreal } |x - u| + \text{ennreal } (-\ln |x -$ 
     $u|) \partial \text{rho})$ 
    using ennreal-abs-ln-le by (auto intro: nn-integral-mono)
  have s2:  $(\int^+ u. \text{ennreal } |x - u| + \text{ennreal } (-\ln |x -$ 
     $u|) \partial \text{rho}) = (\int^+ u. \text{ennreal } |x - u| \partial \text{rho}) +$ 
     $(\int^+ u. \text{ennreal } (-\ln |x - u|) \partial \text{rho})$ 
    by (rule nn-integral-add[OF m1 m2])
  have a: AE u in rho.  $\text{ennreal } |x - u| \leq \text{ennreal } (|x| + 2)$ 
    using rho-AE-Ioo
  by (fastforce intro: ennreal-leI elim: eventually-mono)

```

```

have s3: ( $\int^+ u. \text{ennreal } |x - u| \partial \text{rho}$ )  $\leq$ 
   $\text{ennreal } (|x| + 2) * \text{emeasure } \text{rho } (\text{space } \text{rho})$ 
using nn-integral-mono-AE[OF a] by simp
have f1:  $\text{emeasure } \text{rho } (\text{space } \text{rho}) < \infty$ 
using finite-measure.emeasure-finite[OF rho-finite, of space rho]
by (simp add: less-top)
have b1: ( $\int^+ u. \text{ennreal } (\text{norm } (\ln |x - u|)) \partial \text{rho}$ )  $\leq$ 
   $\text{ennreal } (|x| + 2) * \text{emeasure } \text{rho } (\text{space } \text{rho}) + \text{ennreal } 80$ 
using s1 s2 s3 rho-neg-log-le[of x] by (auto intro: add-mono order-trans)
have b2:  $\text{ennreal } (|x| + 2) * \text{emeasure } \text{rho } (\text{space } \text{rho}) + \text{ennreal } 80 < \infty$ 
using f1
by (simp add: ennreal-mult-less-top less-top)
have fin: ( $\int^+ u. \text{ennreal } (\text{norm } (\ln |x - u|)) \partial \text{rho}$ )  $<$ 
   $\infty$  by (rule le-less-trans[OF b1 b2])
show ?thesis using m fin unfolding integrable-iff-bounded
by simp
qed

lemma rho-logmax-integrable:
  assumes e:  $0 < \varepsilon$ 
  shows integrable rho ( $\lambda u. \ln (\max \varepsilon |x - u|)$ )
proof -
  have m: ( $\lambda u. \ln (\max \varepsilon |x - u|)$ )  $\in \text{borel-measurable } \text{rho}$ 
    unfolding rho-def
    by simp
  have pt:  $\text{norm } (\ln (\max \varepsilon |x - u|)) \leq$ 
     $|\ln \varepsilon| + (|x| + 2)$  if u:  $0 < u \wedge u < 2$  for u
  proof -
    have  $|\ln (\max \varepsilon |x - u|)| \leq$ 
       $|\ln \varepsilon| + |x - u|$  by (rule abs-ln-max-le[OF e abs-ge-zero])
    thus ?thesis using u
    by (smt (verit, del-insts) real-norm-def)
  qed
  have ae: AE u in rho.  $\text{norm } (\ln (\max \varepsilon |x - u|)) \leq$ 
     $|\ln \varepsilon| + (|x| + 2)$  using rho-AE-Ioo pt
    by (auto elim: eventually-mono)
  show ?thesis using finite-measure.integrable-const-bound[OF rho-finite ae m] .
qed

lemma rho-logmax-le:
  assumes e:  $0 < \varepsilon$ 
  shows  $\text{integral}^L \text{rho } (\lambda u. \ln (\max \varepsilon |x - u|)) \leq$ 
     $\log \text{Pot } \text{rho } x + 80 * \text{sqrt } \varepsilon$ 
proof -
  define g where  $g u = \ln (\max \varepsilon |x - u|) - \ln |x - u|$  for u
  have i1: integrable rho ( $\lambda u. \ln (\max \varepsilon |x - u|)$ )
    by (rule rho-logmax-integrable[OF e])
  have i2: integrable rho ( $\lambda u. \ln |x - u|$ ) by (rule rho-log-integrable)
  have ig: integrable rho g unfolding g-def using i1 i2

```

```

  by simp
have dif: integralL rho (λu. ln (max ε |x - u|)) =
  logPot rho x + integralL rho g
  unfolding g-def logPot-def using Bochner-Integration.integral-diff[OF i1 i2]
  by simp
have nul: {x} ∈ null-sets rho using rho-singleton[of x] unfolding rho-def
  by auto
have ane: AE u in rho. u ≠ x using AE-I'[OF nul, of λu. u ≠ x]
  by auto
have gpt: g u = max 0 (ln (ε / |x - u|)) if u: u ≠ x for u
proof -
  have d0: 0 < |x - u| using u
  by simp
  have ld: ln (ε / |x - u|) = ln ε - ln |x - u| using d0 e
  by (simp add: ln-div)
  show ?thesis
proof (cases ε ≤ |x - u|)
  case True
  have ln ε ≤ ln |x - u| using True e d0
  by simp
  thus ?thesis unfolding g-def ld using True
  by (simp add: max-def)
next
  case False
  have ln |x - u| < ln ε using False e d0
  by simp
  thus ?thesis unfolding g-def ld using False
  by (simp add: max-def)
qed
qed
have gnn: AE u in rho. 0 ≤ g u using ane gpt
  by (auto elim: eventually-mono)
have nnI: ennreal (integralL rho g) =
  (∫+ u. ennreal (g u) ∂rho)
  using nn-integral-eq-integral[OF ig gnn]
  by simp
have nnE: (∫+ u. ennreal (g u) ∂rho) =
  (∫+ u. ennreal (max 0 (ln (ε / |x - u|))) ∂rho)
  using ane gpt by (auto intro: nn-integral-cong-AE elim: eventually-mono)
have ennreal (integralL rho g) ≤ ennreal (80 * sqrt ε) unfolding nnI nnE
  by (rule cross-bound[OF e])
hence b2: integralL rho g ≤ 80 * sqrt ε using e
  by (simp add: ennreal-le-iff)
show ?thesis unfolding dif using b2
  by simp
qed

```

The energies of a scaled sum of circle measures.

lemma *csum-scale-integral*:

fixes $t :: \text{nat} \Rightarrow \text{real}$ **and** $h :: \text{nat}$ **and** $\varepsilon r :: \text{real}$ **and** $f :: \text{complex} \Rightarrow \text{real}$
assumes $r: 0 \leq r$ **and** $f: f \in \text{borel-measurable borel}$ **and** $fi: \bigwedge i. i < h \Rightarrow \text{integrable } (\text{circ } (\text{of-real } (t \ i)) \ \varepsilon) \ f$
shows $\text{integral}^L (\text{scale-measure } (\text{ennreal } r) (\text{distr } (\text{count-space } \{..<h\} \otimes_M \text{cden}) \text{ borel } (\lambda(i, \vartheta). \text{of-real } (t \ i) + \text{of-real } \varepsilon * \text{cis } \vartheta))) \ f$
 $= r * (\sum i < h. \text{integral}^L (\text{circ } (\text{of-real } (t \ i)) \ \varepsilon) \ f)$
proof –
have $fi': \bigwedge i. i \in \{..<h\} \Rightarrow \text{integrable } (\text{circ } (\text{of-real } (t \ i)) \ \varepsilon) \ f$ **using** fi
by *simp*
have $fD: f \in \text{borel-measurable } (\text{distr } (\text{count-space } \{..<h\} \otimes_M \text{cden}) \text{ borel } (\lambda(i, \vartheta). \text{of-real } (t \ i) + \text{of-real } \varepsilon * \text{cis } \vartheta))$ **using** f
by *simp*
show $?thesis$ **using** $\text{scale-integral}(1)[OF \ r \ fD] \ \text{csum-integral}(2)[OF \ \text{finite-lessThan } f \ fi']$
by *simp*
qed

lemma *csum-energy-ge*:

fixes $t :: \text{nat} \Rightarrow \text{real}$ **and** $h :: \text{nat}$ **and** $\varepsilon r :: \text{real}$ **and** $S :: \text{complex measure}$
assumes $e: 0 < \varepsilon$ **and** $\text{inj}: \text{inj-on } t \ \{..<h\}$ **and** $r: 0 \leq r$
and $S: S = \text{scale-measure } (\text{ennreal } r) (\text{distr } (\text{count-space } \{..<h\} \otimes_M \text{cden}) \text{ borel } (\lambda(i, \vartheta). \text{of-real } (t \ i) + \text{of-real } \varepsilon * \text{cis } \vartheta))$
shows $r \wedge^2 * (\sum i < h. \sum j < h. (\text{if } i = j \text{ then } \ln \varepsilon \text{ else } \ln |t \ i - t \ j|)) \leq \text{logEnergyC } S \ S$
proof –
define $c :: \text{nat} \Rightarrow \text{complex}$ **where** $c \ i = \text{of-real } (t \ i)$ **for** i
define $\Phi :: \text{complex} \Rightarrow \text{real}$ **where** $\Phi \ z = (\sum j < h. \ln (\max \varepsilon (\text{norm } (c \ j - z))))$ **for** z
have $SI: \bigwedge f. f \in \text{borel-measurable borel} \Rightarrow (\bigwedge i. i < h \Rightarrow \text{integrable } (\text{circ } (c \ i) \ \varepsilon) \ f) \Rightarrow \text{integral}^L \ S \ f = r * (\sum i < h. \text{integral}^L (\text{circ } (c \ i) \ \varepsilon) \ f)$
unfolding $S \ c\text{-def}$ **using** $\text{csum-scale-integral}[OF \ r]$
by *simp*
have $\text{inner}: \bigwedge z. \text{integral}^L \ S \ (\lambda w. \ln (\text{norm } (z - w))) = r * \Phi \ z$
proof –
fix $z :: \text{complex}$
have $fe: (\lambda w. \ln (\text{norm } (z - w))) = (\lambda w. \ln (\text{norm } (w - z)))$
by (*simp add: norm-minus-commute*)
have $mf: (\lambda w :: \text{complex}. \ln (\text{norm } (w - z))) \in \text{borel-measurable borel}$
by *simp*
have $ii: \bigwedge i. i < h \Rightarrow \text{integrable } (\text{circ } (c \ i) \ \varepsilon) \ (\lambda w. \ln (\text{norm } (w - z)))$
by (*rule circ-log-integrable[OF e]*)
have $cl: \bigwedge j. \text{integral}^L (\text{circ } (c \ j) \ \varepsilon) \ (\lambda w. \ln (\text{norm } (w - z))) = \ln (\max \varepsilon (\text{norm } (c \ j - z)))$


```

    by (rule circ-log[OF e])
  show  $\text{integral}^L S (\lambda w. \ln (\text{norm } (z - w))) = r * \Phi z$ 
    unfolding fe  $\Phi$ -def using SI[OF mf ii] cl
    by simp
qed
have  $m\Phi j$ :  $\bigwedge j. (\lambda z. \ln (\max \varepsilon (\text{norm } (c j - z)))) \in$ 
  borel-measurable borel
  by simp
have  $m\Phi$ :  $\Phi \in \text{borel-measurable borel}$  unfolding  $\Phi$ -def[abs-def] using  $m\Phi j$ 
  by simp
have  $i\Phi j$ :  $\bigwedge i j. \text{integrable } (\text{circ } (c i) \varepsilon) (\lambda z. \ln (\max \varepsilon$ 
   $(\text{norm } (c j - z))))$ 
proof -
  fix i j
  show  $\text{integrable } (\text{circ } (c i) \varepsilon) (\lambda z. \ln (\max \varepsilon (\text{norm } (c j - z))))$ 
  proof (rule circ-integrable-bound[where B =
     $|\ln \varepsilon| + (\text{norm } (c j) + \text{norm } (c i) + |\varepsilon|)$ ])
    show  $(\lambda z. \ln (\max \varepsilon (\text{norm } (c j - z)))) \in \text{borel-measurable borel}$ 
      by (rule  $m\Phi j$ )
    fix z :: complex assume hz:  $\text{norm } (z - c i) = |\varepsilon|$ 
    have b:  $|\ln (\max \varepsilon (\text{norm } (c j - z)))| \leq$ 
       $|\ln \varepsilon| + \text{norm } (c j - z)$  by (rule abs-ln-max-le[OF e norm-ge-zero])
    have nz:  $\text{norm } (c j - z) \leq \text{norm } (c j) + \text{norm } (c i) + |\varepsilon|$ 
      using hz norm-triangle-ineq4[of c j z] norm-triangle-sub[of z c i]
      by simp
    show  $|\ln (\max \varepsilon (\text{norm } (c j - z)))| \leq$ 
       $|\ln \varepsilon| + (\text{norm } (c j) + \text{norm } (c i) + |\varepsilon|)$  using b nz
      by simp
  qed
qed
qed
have  $i\Phi$ :  $\bigwedge i. i < h \implies \text{integrable } (\text{circ } (c i) \varepsilon) \Phi$ 
  unfolding  $\Phi$ -def using  $i\Phi j$ 
  by simp
have outer:  $\logEnergyC S S = r * (r * (\sum i <$ 
   $h. \text{integral}^L (\text{circ } (c i) \varepsilon) \Phi))$ 
  unfolding logEnergyC-def inner using SI[OF m $\Phi$  i $\Phi$ ]
  by simp
have split:  $\bigwedge i. \text{integral}^L (\text{circ } (c i) \varepsilon) \Phi = (\sum j <$ 
   $h. \text{integral}^L (\text{circ } (c i) \varepsilon) (\lambda z. \ln (\max \varepsilon (\text{norm } (c j - z))))$ 
  unfolding  $\Phi$ -def using  $i\Phi j$ 
  by simp
have trm:  $\bigwedge i j. i < h \implies j < h \implies$ 
   $(\text{if } i = j \text{ then } \ln \varepsilon \text{ else } \ln |t i - t j|) \leq$ 
   $\text{integral}^L (\text{circ } (c i) \varepsilon) (\lambda z. \ln (\max \varepsilon (\text{norm } (c j - z))))$ 
proof -
  fix i j assume i:  $i < h$  and j:  $j < h$ 
  show  $(\text{if } i = j \text{ then } \ln \varepsilon \text{ else } \ln |t i - t j|) \leq$ 
     $\text{integral}^L (\text{circ } (c i) \varepsilon) (\lambda z. \ln (\max \varepsilon (\text{norm } (c j - z))))$ 
  proof (cases  $i = j$ )

```

```

case True
have ae: AE z in circ (c i) ε. ln (max ε (norm (c j - z))) =
  ln ε using circ-AE[of c i ε] True e
  by (auto simp: norm-minus-commute elim: eventually-mono)
have m1: (λz. ln (max ε (norm (c j - z)))) ∈
  borel-measurable (circ (c i) ε)
  using mΦj[of j] measurable-cong-sets[OF sets-circ refl]
  by blast
have ic1: integralL (circ (c i) ε) (λz. ln (max ε (norm (c j
  - z)))) = integralL (circ (c i) ε) (λz. ln ε)
  using integral-cong-AE[OF m1 - ae]
  by simp
have ic2: integralL (circ (c i) ε) (λz. ln ε) =
  ln ε using circ-univ[of c i ε] space-circ[of c i ε]
  by (simp add: measure-def)
show ?thesis using True ic1 ic2
  by auto
next
case False
have t i ≠ t j using inj i j False unfolding inj-on-def
  by auto
hence dpos: 0 < |t i - t j|
  by simp
have cne: norm (c i - c j) = |t i - t j| unfolding c-def
  by (metis norm-of-real of-real-hom.hom-minus)
have nul: {c j} ∈ null-sets (circ (c i) ε)
  using circ-singleton[OF e, of c i c j] sets-circ[of c i ε]
  by auto
have ane: AE z in circ (c i) ε. z ≠ c j
  using AE-I'[OF nul, of λz. z ≠ c j]
  by auto
have pt: ∧z. z ≠ c j ⇒
  ln (norm (z - c j)) ≤ ln (max ε (norm (c j - z)))
  by (simp add: ln-mono norm-minus-commute)
have aele: AE z in circ (c i) ε. ln (norm (z - c j)) ≤
  ln (max ε (norm (c j - z))) using ane pt
  by (auto elim: eventually-mono)
have i1: integrable (circ (c i) ε) (λz. ln (norm (z - c j)))
  by (rule circ-log-integrable[OF e])
have mono: integralL (circ (c i) ε) (λz. ln (norm (z - c j))) ≤
  integralL (circ (c i) ε) (λz. ln (max ε (norm (c j - z))))
  by (rule integral-mono-AE[OF i1 iΦj aele])
have cl: integralL (circ (c i) ε) (λz. ln (norm (z - c j))) =
  ln (max ε (norm (c i - c j))) by (rule circ-log[OF e])
have lm: ln |t i - t j| ≤ ln (max ε (norm (c i - c j)))
  unfolding cne using dpos e
  by simp
show ?thesis using False mono cl lm
  by simp

```

```

qed
qed
have sumle: ( $\sum i < h$ .  $\sum j < h$ . (if  $i = j$  then  $\ln \varepsilon$  else  $\ln |t i - t j|$ ))  $\leq (\sum i < h$ .  $\text{integral}^L (\text{circ } (c i) \varepsilon) \Phi$ )
  unfolding split using trm
  by (auto intro: sum-mono)
show ?thesis unfolding outer using sumle r mult-left-mono[OF sumle, of r * r]
  by (simp add: power2-eq-square mult.assoc)
qed

lemma csum-cross-le:
  fixes t :: nat  $\Rightarrow$  real and h :: nat and  $\varepsilon$  r :: real and S :: complex
    measure
  assumes e:  $0 < \varepsilon$  and e1:  $\varepsilon \leq 1$  and tnn:  $\bigwedge i. i < h \implies 0 \leq t i$  and r:  $0 \leq r$ 
    and S:  $S = \text{scale-measure } (\text{ennreal } r) (\text{distr } (\text{count-space } \{..<h\}) \otimes_M \text{cdem})$ 
  borel
    ( $\lambda(i, \vartheta). \text{of-real } (t i) + \text{of-real } \varepsilon * \text{cis } \vartheta$ )
  shows  $\log \text{EnergyC } S (\text{distr } \rho \text{ borel of-real}) \leq r * (\sum i < h. (\log \text{Pot } \rho (t i) + 80 * \text{sqrt } \varepsilon))$ 
proof -
  define c :: nat  $\Rightarrow$  complex where  $c i = \text{of-real } (t i)$  for i
  define  $\varrho C$  :: complex measure where  $\varrho C = \text{distr } \rho \text{ borel of-real}$ 
  define F :: complex  $\Rightarrow$  real where  $F z = \text{integral}^L \varrho C (\lambda w. \ln (\text{norm } (z - w)))$  for z
  have SI:  $\bigwedge f. f \in \text{borel-measurable borel} \implies (\bigwedge i. i < h \implies \text{integrable } (\text{circ } (c i) \varepsilon) f) \implies \text{integral}^L S f = r * (\sum i < h. \text{integral}^L (\text{circ } (c i) \varepsilon) f)$ 
    unfolding S c-def using csum-scale-integral[OF r]
    by simp
  have fr: finite-measure  $\varrho C$  unfolding  $\varrho C$ -def by (rule rhoC-finite)
  have sr: sets  $\varrho C = \text{sets borel}$  unfolding  $\varrho C$ -def by (rule sets-rhoC)
  have sfr: sigma-finite-measure  $\varrho C$  using fr
    by (simp add: finite-measure-def)
  have mp0:  $(\lambda p :: \text{complex} \times \text{complex}. \ln (\text{norm } (\text{fst } p - \text{snd } p))) \in \text{borel-measurable } (\text{borel } \otimes_M \text{borel})$ 
    by simp
  have sp: sets  $((\text{borel} :: \text{complex measure}) \otimes_M \varrho C) = \text{sets } (\text{borel } \otimes_M \text{borel})$  using sr
    using sets-pair-measure-cong by simp
  have mp:  $(\lambda(z, w). \ln (\text{norm } (z - w))) \in \text{borel-measurable } ((\text{borel} :: \text{complex measure}) \otimes_M \varrho C)$ 
    unfolding split-beta' using mp0 measurable-cong-sets[OF sp refl]
    by blast
  have mF:  $F \in \text{borel-measurable borel}$  unfolding F-def[abs-def]
    using sigma-finite-measure.borel-measurable-lebesgue-integral[OF sfr mp] .
  have aeR:  $AE w \text{ in } \varrho C. \text{norm } w \leq 2$  unfolding  $\varrho C$ -def by (rule rhoC-AE)

```

```

have lgR:  $\bigwedge z. (\int^+ w. \text{ennreal } (-\ln(\text{norm } (z - w))) \partial \varrho C) \leq$ 
   $\text{ennreal } 80$  unfolding  $\varrho C\text{-def}$  by (rule rhoC-neg-log)
have bound:  $\text{integral}^L (\text{circ } (c \ i) \ \varepsilon) \ F \leq$ 
   $\text{logPot rho } (t \ i) + 80 * \text{sqrt } \varepsilon$ 
  and  $iF: \text{integrable } (\text{circ } (c \ i) \ \varepsilon) \ F$  if  $i: i < h$  for  $i$ 
proof -
  have ci:  $\text{norm } (c \ i) = t \ i$  unfolding  $c\text{-def}$  using  $\text{tnn}[OF \ i]$ 
  by simp
  have aeC:  $AE \ z \text{ in } \text{circ } (c \ i) \ \varepsilon. \ \text{norm } z \leq t \ i + 3$ 
proof -
  have pt:  $\text{norm } z \leq t \ i + 3$  if  $hz: \text{norm } (z - c \ i) = |\varepsilon|$  for  $z$ 
    using  $hz \ ci \ e \ e1 \ \text{norm-triangle-sub}[of \ z \ c \ i]$ 
    by simp
  show ?thesis using  $\text{circ-AE}[of \ c \ i \ \varepsilon] \ pt$ 
    by (auto elim: eventually-mono)
qed
have aeR':  $AE \ w \text{ in } \varrho C. \ \text{norm } w \leq t \ i + 3$  using  $aeR \ \text{tnn}[OF \ i]$ 
  by (auto elim: eventually-mono)
have Pi:  $\text{integrable } (\text{circ } (c \ i) \ \varepsilon \otimes_M \varrho C) \ (\lambda p. \ln(\text{norm}$ 
   $(fst \ p - snd \ p)))$ 
  by (rule integrable-log-prod[ $OF \ \text{circ-finite}$  fr sets-circ sr aeC aeR' lgR])
have Pi':  $\text{integrable } (\text{circ } (c \ i) \ \varepsilon \otimes_M \varrho C) \ (\lambda(x,$ 
   $y). \ln(\text{norm } (x - y)))$ 
  unfolding split-beta' by (rule Pi)
have ps:  $\text{pair-sigma-finite } (\text{circ } (c \ i) \ \varepsilon) \ \varrho C$ 
  using  $\text{circ-finite}[of \ c \ i \ \varepsilon]$  fr
  unfolding  $\text{pair-sigma-finite-def}$  by (simp add: finite-measure-def)
show  $\text{integrable } (\text{circ } (c \ i) \ \varepsilon) \ F$  unfolding  $F\text{-def}[abs-def]$ 
  using  $\text{pair-sigma-finite.integrable-fst}[OF \ ps \ Pi']$  .
have s1:  $\text{integral}^L (\text{circ } (c \ i) \ \varepsilon) \ F =$ 
   $\text{integral}^L (\text{circ } (c \ i) \ \varepsilon \otimes_M \varrho C) \ (\lambda(x,$ 
   $y). \ln(\text{norm } (x - y)))$ 
  unfolding  $F\text{-def}[abs-def]$  using  $\text{pair-sigma-finite.integral-fst}[OF \ ps \ Pi']$  .
have s2:  $\text{integral}^L (\text{circ } (c \ i) \ \varepsilon \otimes_M \varrho C) \ (\lambda(x,$ 
   $y). \ln(\text{norm } (x - y))) = (\int y. (\int x. \ln(\text{norm } (x -$ 
   $y))) \partial \text{circ } (c \ i) \ \varepsilon) \ \partial \varrho C)$ 
  using  $\text{pair-sigma-finite.integral-snd}[OF \ ps \ Pi']$  by (rule sym)
have s3:  $\bigwedge y. (\int x. \ln(\text{norm } (x - y))) \ \partial \text{circ } (c \ i) \ \varepsilon =$ 
   $\ln(\max \varepsilon (\text{norm } (c \ i - y)))$  by (rule circ-log[ $OF \ e$ ])
have m4:  $(\lambda y :: \text{complex}. \ln(\max \varepsilon (\text{norm } (c \ i - y)))) \in$ 
   $\text{borel-measurable borel}$ 
  by simp
have s4:  $(\int y. \ln(\max \varepsilon (\text{norm } (c \ i - y))) \ \partial \varrho C) =$ 
   $(\int u. \ln(\max \varepsilon (\text{norm } (c \ i - of\text{-real } u))) \ \partial \rho)$ 
  unfolding  $\varrho C\text{-def}$  by (rule integral-distr[ $OF \ \text{rhoC-meas } m4$ ])
have s5:  $\bigwedge u. \text{norm } (c \ i - of\text{-real } u) = |t \ i - u|$  unfolding  $c\text{-def}$ 
  by (metis norm-of-real of-real-hom.hom-minus)
have eqF:  $\text{integral}^L (\text{circ } (c \ i) \ \varepsilon) \ F =$ 
   $(\int u. \ln(\max \varepsilon |t \ i - u|) \ \partial \rho)$ 

```

```

    unfolding s1 s2 s3 s4 s5 ..
  show  $\text{integral}^L (\text{circ } (c \ i) \ \varepsilon) \ F \leq \log \text{Pot } \rho (t \ i) + 80 * \text{sqrt } \varepsilon$ 
    unfolding eqF by (rule rho-logmax-le[OF e])
qed
have outer:  $\log \text{Energy} C \ S \ \varrho C = r * (\sum i <$ 
   $h. \text{integral}^L (\text{circ } (c \ i) \ \varepsilon) \ F)$ 
  unfolding logEnergyC-def F-def[symmetric] using SI[OF mF iF]
  by simp
have sle:  $(\sum i < h. \text{integral}^L (\text{circ } (c \ i) \ \varepsilon) \ F) \leq (\sum i <$ 
   $h. (\log \text{Pot } \rho (t \ i) + 80 * \text{sqrt } \varepsilon))$ 
  using bound by (blast intro: sum-mono)
show ?thesis unfolding  $\varrho C$ -def[symmetric] outer using mult-left-mono[OF sle
r] .
qed

```

theorem configuration-bound:

```

  fixes t :: nat  $\Rightarrow$  real
  assumes  $40 \text{ dvd } K \ 0 < K \ \wedge i. i < \text{hof } K \implies$ 
     $0 \leq t \ i \text{ inj-on } t \ \{..<\text{hof } K\}$ 
  shows  $2 * (\sum i < \text{hof } K. \sum j \in \{j. i < j \wedge$ 
     $j < \text{hof } K\}. \ln |t \ i - t \ j|)$ 
     $- \text{real } K * (\sum i < \text{hof } K. \text{Vf } (t \ i)) + (\sum i < \text{hof } K. \text{sqrt } (t \ i))$ 
     $\leq (\text{lam-r} * \text{of-rat } M0c - \log \text{Energy } \rho) * \text{real } K ^ 2$ 
     $+ (120 + \text{sqrt } 2) * \text{real } (\text{hof } K) + 2 * \text{real } (\text{hof } K) * \ln (\text{real } K)$ 
proof -
  define h where  $h = \text{hof } K$ 
  define  $\varepsilon$  where  $\varepsilon = (1 / (2 * \text{real } K)) ^ 2$ 
  have e0:  $0 < \varepsilon$  using assms(2) unfolding  $\varepsilon$ -def by simp
  have sqe:  $\text{sqrt } \varepsilon = 1 / (2 * \text{real } K)$  unfolding  $\varepsilon$ -def using assms(2)
    by simp
  have lne:  $\ln \varepsilon = - 2 * \ln 2 - 2 * \ln (\text{real } K)$  unfolding  $\varepsilon$ -def using assms(2)
    by (simp add: ln-realpow ln-div ln-mult)
  have hK:  $\text{real } h = \text{lam-r} * \text{real } K$  unfolding h-def using assms(1)
    by (auto elim!: dvdE simp: hof-def lam-r-def)

```

The regularised configuration $\sigma = K^{-1} \sum -i \omega(t-i, \varepsilon)$ as one push-forward measure, and ϱ on $\mathbb{C}$.

```

  define  $\sigma$  where  $\sigma = \text{scale-measure } (\text{ennreal } (1 / \text{real } K))$ 
     $(\text{distr } (\text{count-space } \{..<h\} \otimes_M \text{density lborel } (\lambda \vartheta. \text{ennreal}$ 
     $(\text{indicator } \{0..2 * \pi i\} \ \vartheta / (2 * \pi i)))) \text{ borel}$ 
     $(\lambda(i, \vartheta). \text{of-real } (t \ i) + \text{of-real } \varepsilon * \text{cis } \vartheta))$ 
  define  $\varrho C :: \text{complex measure}$  where  $\varrho C = \text{distr } \rho \text{ borel of-real}$ 
  have sig-univ:  $\text{emeasure } \sigma \text{ UNIV} = \text{ennreal } (\text{real } h / \text{real } K)$ 
proof -
  have D1:  $\text{emeasure } (\text{distr } (\text{count-space } \{..<h\} \otimes_M \text{density lborel}$ 
     $(\lambda \vartheta. \text{ennreal } (\text{indicator } \{0..2 * \pi i\} \ \vartheta / (2 * \pi i)))) \text{ borel } (\lambda(i,$ 
     $\vartheta). \text{of-real } (t \ i) + \text{of-real } \varepsilon * \text{cis } \vartheta)) \text{ UNIV} = \text{of-nat } h$ 
    using csum-emeasure[where  $I = \{..<h\}$  and  $A = \text{UNIV}$  and  $c =$ 
     $\lambda i. \text{of-real } (t \ i)$  and  $\varepsilon = \varepsilon]$  circ-univ

```

```

    by simp
  show ?thesis unfolding  $\sigma$ -def using D1
    by (simp add: ennreal-of-nat-eq-real-of-nat ennreal-mult'[symmetric])
qed
have rhoC-univ: emeasure  $\varrho C$  UNIV = ennreal lam-r
proof -
  have m: (of-real :: real  $\Rightarrow$  complex)  $\in$  measurable rho borel unfolding rho-def
    by simp
  have sp: space rho = UNIV unfolding rho-def
    by simp
  show ?thesis unfolding  $\varrho C$ -def using emeasure-distr[OF m, of UNIV] sp
rho-univ
    by simp
qed
have mass: emeasure  $\sigma$  UNIV = emeasure  $\varrho C$  UNIV unfolding sig-univ rhoC-univ
hK
  using assms(2) by simp
  have inj!: inj-on t {.. $h$ } using assms(4) unfolding h-def .
  have k0:  $0 \leq 1 / \text{real } K$ 
    by simp
  have energy-sig-ge:  $(1 / \text{real } K \wedge 2) * ((\sum i < h. \sum j < h. (\text{if } i = j \text{ then } \ln \varepsilon \text{ else } \ln |t\ i - t\ j|))) \leq \log \text{EnergyC } \sigma \sigma$ 
    using csum-energy-ge[OF e0 inj' k0  $\sigma$ -def] by (simp add: power-divide)
  have e1c:  $\varepsilon \leq 1$  unfolding  $\varepsilon$ -def using assms(2)
    by (simp add: power-le-one)
  have tnn:  $\bigwedge i. i < h \implies 0 \leq t\ i$  using assms(3) unfolding h-def
    by simp
  have energy-cross-le:  $\log \text{EnergyC } \sigma \varrho C \leq (1 / \text{real } K) * (\sum i < h. (\log \text{Pot } \rho (t\ i) + 80 * \text{sqrt } \varepsilon))$ 
    unfolding  $\varrho C$ -def by (rule csum-cross-le[OF e0 e1c tnn k0  $\sigma$ -def])
  have energy-rhoC:  $\log \text{EnergyC } \varrho C \varrho C = \log \text{EnergyC } \rho$ 
proof -
  have inner:  $\bigwedge z. \text{integral}^L \varrho C (\lambda w. \ln (\text{norm } (z - w))) = \text{integral}^L \rho (\lambda u. \ln (\text{norm } (z - \text{of-real } u)))$ 
    unfolding  $\varrho C$ -def using integral-distr[OF rhoC-meas] by force
  have sfr: sigma-finite-measure rho using rho-finite
    by (simp add: finite-measure-def)
  have mp0:  $(\lambda p :: \text{complex} \times \text{real}. \ln (\text{norm } (\text{fst } p - \text{of-real } (\text{snd } p)))) \in \text{borel-measurable } (\text{borel } \otimes_M \text{borel})$ 
    by simp
  have sp: sets ((borel :: complex measure)  $\otimes_M$  rho) = sets (borel  $\otimes_M$  borel) unfolding rho-def
    using sets-pair-measure-cong by simp
  have mp:  $(\lambda (z, u). \ln (\text{norm } (z - \text{of-real } u))) \in \text{borel-measurable } ((\text{borel} :: \text{complex measure}) \otimes_M \rho)$ 
    unfolding split-beta' using mp0 measurable-cong-sets[OF sp refl] by blast
  have mG:  $(\lambda z :: \text{complex}. \text{integral}^L \rho (\lambda u. \ln (\text{norm } (z - \text{of-real } u)))) \in \text{borel-measurable borel}$ 
    using sigma-finite-measure.borel-measurable-lebesgue-integral[OF sfr mp] .

```

```

have outer:  $\text{integral}^L \varrho C (\lambda z. \text{integral}^L \rho (\lambda u. \ln (\text{norm } (z - \text{of-real } u)))) = \text{integral}^L \rho (\lambda x. \text{integral}^L \rho (\lambda u. \ln (\text{norm } (\text{of-real } x - \text{of-real } u :: \text{complex}))))$ 
unfolding  $\varrho C\text{-def}$  by (rule integral-distr[OF rhoC-meas mG])
have pt:  $\bigwedge x u :: \text{real}. \ln (\text{norm } (\text{of-real } x - \text{of-real } u :: \text{complex})) = \ln |x - u|$ 
by (metis norm-of-real of-real-hom.hom-minus)
show ?thesis unfolding logEnergyC-def logEnergy-def logPot-def inner outer
pt ..
qed

```

The dominating measure $\eta = \sigma + \varrho C$ for Lemma 6.2 (Isabelle has no sum of measures: use *sup-measure'* or an explicit *measure-of*).

```

define  $\eta :: \text{complex measure}$  where  $\eta = \text{measure-of UNIV } (\text{sets borel}) (\lambda A. \text{emeasure } \sigma A + \text{emeasure } \varrho C A)$ 
have sets-sig:  $\text{sets } \sigma = \text{sets borel}$ 
unfolding  $\sigma\text{-def}$  by simp
have sets-rho:  $\text{sets } \varrho C = \text{sets borel}$ 
unfolding  $\varrho C\text{-def}$  by simp
have sets-eta:  $\text{sets } \eta = \text{sets borel}$ 
unfolding  $\eta\text{-def}$  by (rule summ-sets)
have em-eta:  $\bigwedge A. A \in \text{sets borel} \implies \text{emeasure } \eta A = \text{emeasure } \sigma A + \text{emeasure } \varrho C A$ 
unfolding  $\eta\text{-def}$  using summ-emeasure[OF sets-sig sets-rho] by simp
have fin:  $\text{finite-measure } \eta$ 
using em-eta[of UNIV] sig-univ rhoC-univ sets-eta sets-eq-imp-space-eq[OF sets-eta]
by (auto intro: finite-measureI)
have le1:  $\sigma \leq \eta$ 
using le-measure[of  $\sigma \eta$ ] sets-sig sets-eta em-eta by simp
have le2:  $\varrho C \leq \eta$ 
using le-measure[of  $\varrho C \eta$ ] sets-rho sets-eta em-eta by simp
have nn-sum:  $\bigwedge f. f \in \text{borel-measurable borel} \implies (\int^+ z. f z \partial \eta) \leq (\int^+ z. f z \partial \sigma) + (\int^+ z. f z \partial \varrho C)$ 
unfolding  $\eta\text{-def}$  using summ-nn[OF sets-sig sets-rho] by simp
have int-eta:  $\text{integrable } (\eta \otimes_M \eta) (\lambda zw. \ln (\text{norm } (\text{fst } zw - \text{snd } zw)))$ 
proof -
define  $R0 :: \text{real}$  where  $R0 = (\sum i < h. t i) + 3$ 
have tnn:  $\bigwedge i. i < h \implies 0 \leq t i$  using assms(3) unfolding h-def by simp
have tle:  $\bigwedge i. i < h \implies t i \leq (\sum j < h. t j)$  using tnn by (auto intro: member-le-sum)
have e1:  $\varepsilon \leq 1$  unfolding  $\varepsilon\text{-def}$  using assms(2) by (simp add: power-le-one)
have R3:  $3 \leq R0$  unfolding R0-def using tnn by (auto intro: sum-nonneg)

```

```

have cR:  $\bigwedge i. i < h \implies$ 
  norm (of-real (t i) :: complex) +  $\varepsilon \leq R0$  unfolding R0-def using tle tnn e1
  by force
have lne0:  $\ln \varepsilon \leq 0$  using e0 e1
  by simp
define B where  $B = 2 * R0 - \ln \varepsilon$ 
have B0:  $0 \leq B$  unfolding B-def using R3 lne0
  by simp
define N where  $N = \{z :: \text{complex}. R0 < \text{norm } z\}$ 
have Nm:  $N \in \text{sets borel}$  unfolding N-def
  by (auto intro: borel-open open-Collect-less continuous-intros)
have cN:  $\bigwedge i. i < h \implies \text{emeasure } (\text{circ } (\text{of-real } (t i)) \varepsilon) N = 0$ 
proof -
  fix i assume i:  $i < h$ 
  have pt:  $z \notin N$  if hz:  $\text{norm } (z - \text{of-real } (t i)) = |\varepsilon|$  for z
    using hz cR[OF i] e0 norm-triangle-sub[of z of-real (t i)] unfolding N-def
    by simp
  have ae:  $AE\ z\ \text{in}\ \text{circ } (\text{of-real } (t i))\ \varepsilon. z \notin N$ 
    using circ-AE[of of-real (t i)  $\varepsilon$ ] pt
    by (auto elim: eventually-mono)
  show  $\text{emeasure } (\text{circ } (\text{of-real } (t i)) \varepsilon) N = 0$ 
    using AE-iff-measurable[of N circ (of-real (t i))  $\varepsilon\ \lambda z. z \notin N$ ]
    ae Nm sets-circ space-circ
    by simp
qed
have sN:  $\text{emeasure } \sigma\ N = 0$  unfolding  $\sigma$ -def
  using csum-emeasure[where  $I = \{..<h\}$  and  $A = N$  and  $c =$ 
     $\lambda i. \text{of-real } (t i)$  and  $\varepsilon = \varepsilon$ ] Nm cN
  by simp
have rN:  $\text{emeasure } \rho C\ N = 0$ 
proof -
  have ae:  $AE\ w\ \text{in}\ \rho C. w \notin N$  using rhoC-AE R3 unfolding  $\rho C$ -def N-def
    by (auto elim: eventually-mono)
  show ?thesis using AE-iff-measurable[of N  $\rho C\ \lambda z. z \notin N$ ] ae Nm sets-rho
    sets-eq-imp-space-eq[OF sets-rho]
    by simp
qed
have eN:  $\text{emeasure } \eta\ N = 0$  using em-eta[OF Nm] sN rN
  by simp
have ae $\eta$ :  $AE\ z\ \text{in}\ \eta. \text{norm } z \leq R0$ 
  using AE-iff-measurable[of N  $\eta\ \lambda z. \text{norm } z \leq$ 
     $R0$ ] Nm sets-eta sets-eq-imp-space-eq[OF sets-eta] eN unfolding N-def
  by force
have lg:  $\bigwedge z. \text{norm } z \leq R0 \implies$ 
   $(\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w)))\ \partial \eta) \leq$ 
   $\text{ennreal } (\text{real } h / \text{real } K * B + 80)$ 
proof -
  fix z :: complex assume hz:  $\text{norm } z \leq R0$ 
  have mf:  $(\lambda w. \text{ennreal } (- \ln (\text{norm } (z - w)))) \in \text{borel-measurable borel}$ 

```



```

    by simp
  have mfD: ( $\lambda w. \text{ennreal } (- \ln (\text{norm } (z - w)))$ )  $\in$ 
    borel-measurable (distr (count-space  $\{..<h\}$ )  $\otimes_M$  density lborel
      ( $\lambda \vartheta. \text{ennreal } (\text{indicator } \{0..2 * \pi\} \vartheta / (2 * \pi))$ )) borel ( $\lambda(i, \vartheta). \text{of-real } (t \ i) + \text{of-real } \varepsilon * \text{cis } \vartheta$ )
    using mf
    by simp
  have a1: ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \eta$ )  $\leq$ 
    ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \sigma$ ) +
    ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \varrho C$ )
    by (rule nn-sum[OF mf])
  have a2: ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \sigma$ ) =
    ennreal (1 / real K) * ( $\sum i < h. \int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \text{circ } (\text{of-real } (t \ i)) \varepsilon$ )
    unfolding  $\sigma$ -def using nn-integral-scale-measure[OF mfD] csum-nn[OF
finite-lessThan
mf]
    by simp
  have a3: ( $\sum i < h. \int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \text{circ } (\text{of-real } (t \ i)) \varepsilon$ )  $\leq$  ( $\sum i < h. \text{ennreal } B$ )
    using circ-neg-log[OF e0 cR hz] unfolding B-def
    by (blast intro: sum-mono)
  have a4: ( $\sum i < h. \text{ennreal } B$ ) = ennreal (real h * B) using B0
    by (simp add: ennreal-of-nat-eq-real-of-nat ennreal-mult)
  have a5: ennreal (1 / real K) * ennreal (real h * B) = ennreal (real h / real
K * B)
    using B0
    by (simp add: ennreal-mult[symmetric])
  have a6: ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \varrho C$ )  $\leq$ 
    ennreal 80 unfolding  $\varrho C$ -def by (rule rhoC-neg-log)
  have a7: ennreal (real h / real K * B) + ennreal 80 = ennreal (real h / real
K * B + 80)
    using B0
    by (simp add: ennreal-plus)
  have ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \eta$ )  $\leq$ 
    ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \sigma$ ) +
    ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \varrho C$ )
    by (rule a1)
  also have ...  $\leq$  ennreal (1 / real K) * ( $\sum i < h. \text{ennreal } B$ ) + ennreal 80
    unfolding a2 using a3 a6
    by (force intro: add-mono mult-left-mono)
  also have ... = ennreal (real h / real K * B + 80) unfolding a4 a5 a7 ..
  finally show ( $\int^+ w. \text{ennreal } (- \ln (\text{norm } (z - w))) \partial \eta$ )  $\leq$ 
    ennreal (real h / real K * B + 80) .
qed
show ?thesis by (rule integrable-log-prod[OF fin fin sets-eta sets-eta aen aen
lg])
qed

```

```

have diag: emeasure ( $\eta \otimes_M \eta$ ) {zw. fst zw = snd zw} = 0
proof -
  have sfe: sigma-finite-measure  $\eta$  using fin
  by (simp add: finite-measure-def)
  have Dm0: {zw :: complex  $\times$  complex. fst zw = snd zw}  $\in$ 
    sets (borel  $\otimes_M$  borel)
  proof -
    have mf: ( $\lambda p :: \text{complex} \times \text{complex}.$  fst p - snd p)  $\in$ 
      borel-measurable (borel  $\otimes_M$  borel)
    by simp
    have eq: {p :: complex  $\times$  complex. fst p = snd p} =
      ( $\lambda p.$  fst p - snd p) - ' {0}  $\cap$  space (borel  $\otimes_M$  borel)
    by (auto simp: space-pair-measure)
    show ?thesis unfolding eq using measurable-sets[OF mf, of {0}]
    by simp
  qed
  have Dm: {zw :: complex  $\times$  complex. fst zw = snd zw}  $\in$ 
    sets ( $\eta \otimes_M \eta$ ) using Dm0 sets-pair-measure-cong[OF sets-eta sets-eta]
  by simp
  have e1: emeasure ( $\eta \otimes_M \eta$ ) {zw. fst zw = snd zw} =
    ( $\int^+ z. \int^+ w.$  indicator {zw :: complex  $\times$ 
      complex. fst zw = snd zw} (z, w)  $\partial\eta \partial\eta$ )
  by (rule sigma-finite-measure.emeasure-pair-measure[OF sfe Dm])
  have sz:  $\bigwedge z.$  emeasure  $\sigma$  {z} = 0
  proof -
    fix z :: complex
    have emeasure  $\sigma$  {z} = ennreal (1 / real K) * ( $\sum i <$ 
      h. emeasure (circ (of-real (t i))  $\varepsilon$ ) {z}) unfolding  $\sigma$ -def
    using csum-emeasure[where I = {.. $h$ } and A = {z} and c =
       $\lambda i.$  of-real (t i) and  $\varepsilon = \varepsilon$ ] by simp
    thus emeasure  $\sigma$  {z} = 0 using circ-singleton[OF e0]
    by simp
  qed
  have rz:  $\bigwedge z.$  emeasure  $\varrho C$  {z} = 0 unfolding  $\varrho C$ -def by (rule rhoC-singleton)
  have pt:  $\bigwedge z.$  ( $\int^+ w.$  indicator {zw :: complex  $\times$  complex. fst zw =
    snd zw} (z, w)  $\partial\eta$ ) = 0
  proof -
    fix z :: complex
    have ind: ( $\lambda w.$  indicator {zw :: complex  $\times$  complex. fst zw =
      snd zw} (z, w) :: ennreal) = indicator {z}
    by (simp add: fun-eq-iff indicator-def)
    show ( $\int^+ w.$  indicator {zw :: complex  $\times$  complex. fst zw =
      snd zw} (z, w)  $\partial\eta$ ) = 0 unfolding ind using em-eta[of {z}] sz rz sets-eta
    by simp
  qed
  show ?thesis unfolding e1 pt
  by simp
qed
have eta: finite-measure  $\eta$  sets  $\eta$  = sets borel  $\sigma \leq$ 

```

```

   $\eta \varrho C \leq \eta$ 
  integrable ( $\eta \otimes_M \eta$ ) ( $\lambda zw. \ln (\text{norm } (\text{fst } zw - \text{snd } zw)))$  emeasure ( $\eta \otimes_M \eta$ ) { $zw. \text{fst } zw = \text{snd } zw$ } = 0
  by fact+
  have L62:  $\log \text{Energy} C \sigma \sigma - 2 * \log \text{Energy} C \sigma \varrho C + \log \text{Energy} C \varrho C \varrho C \leq 0$ 
  by (rule log-energy-zero-mass-nonpos[OF eta(1) - - eta(2) eta(3,4) mass eta(5,6)]) (simp-all
    add:  $\sigma$ -def  $\varrho C$ -def)
  have fb:  $\bigwedge i. i < h \implies 2 * \log \text{Pot } \rho (t \ i) - \text{Vf } (t \ i) + \text{sqrt } (t \ i) / \text{real } K \leq \text{of-rat } M0c + \text{sqrt } 2 / \text{real } K$ 
  proof -
    fix i assume i:  $i < h$ 
    have K40:  $40 \leq \text{real } K$  using assms(1,2)
    by (auto elim: dvdE)
    have t0:  $0 \leq t \ i$  using assms(3) i unfolding h-def
    by simp
    show  $2 * \log \text{Pot } \rho (t \ i) - \text{Vf } (t \ i) + \text{sqrt } (t \ i) / \text{real } K \leq \text{of-rat } M0c + \text{sqrt } 2 / \text{real } K$ 
    using field-bound[OF K40 t0]
    by simp
  qed
  have pair:  $(\sum i < h. \sum j < h. (\text{if } i = j \text{ then } \ln \varepsilon \text{ else } \ln |t \ i - t \ j|)) = 2 * (\sum i < h. \sum j \in \{j. i < j \wedge j < h\}. \ln |t \ i - t \ j|) + \text{real } h * \ln \varepsilon$ 
  proof -
    define L where  $L \ i \ j = \ln |t \ i - t \ j|$  for  $i \ j$ 
    have sym:  $L \ i \ j = L \ j \ i$  for  $i \ j$  unfolding L-def
    by (simp add: abs-minus-commute)
    have row:  $(\sum j < h. (\text{if } i = j \text{ then } \ln \varepsilon \text{ else } L \ i \ j)) = \ln \varepsilon + (\sum j \in \{j. i < j \wedge j < h\}. L \ i \ j) + (\sum j \in \{j. j < i\}. L \ i \ j)$  if i:  $i < h$  for i
    proof -
      have r1:  $(\sum j < h. (\text{if } i = j \text{ then } \ln \varepsilon \text{ else } L \ i \ j)) = \ln \varepsilon + (\sum j \in \{..<h\} - \{i\}. L \ i \ j)$  using i
      by (simp add: sum.remove if-distrib sum.If-cases)
      have r2:  $\{..<h\} - \{i\} = \{j. i < j \wedge j < h\} \cup \{j. j < i\}$  using i
      by auto
      have r3:  $(\sum j \in \{j. i < j \wedge j < h\} \cup \{j. j < i\}. L \ i \ j) = (\sum j \in \{j. i < j \wedge j < h\}. L \ i \ j) + (\sum j \in \{j. j < i\}. L \ i \ j)$ 
      by (auto intro: sum.union-disjoint)
      show ?thesis unfolding r1 r2 r3
      by simp
    qed
  qed
  have rows:  $(\sum i < h. \sum j < h. (\text{if } i = j \text{ then } \ln \varepsilon \text{ else } L \ i \ j)) = \text{real } h * \ln \varepsilon + (\sum i < h. \sum j \in \{j. i < j \wedge j < h\}. L \ i \ j) + (\sum i < h. \sum j \in \{j. j < i\}. L \ i \ j)$ 

```

```

    using row
    by (simp add: sum.distrib)
  have sw1:  $(\sum i < h. \sum j \in \{j. j < i\}. L\ i\ j) = (\sum i < h. \sum j \in \{j. j \in \{..<h\} \wedge j < i\}. L\ i\ j)$ 
    by (force intro: sum.cong)
  have sw2:  $(\sum i < h. \sum j \in \{j. j \in \{..<h\} \wedge j < i\}. L\ i\ j) = (\sum j < h. \sum i \in \{i. i \in \{..<h\} \wedge j < i\}. L\ i\ j)$ 
    using sum.swap-restrict[of  $\{..<h\}$   $\{..<h\}$   $L\ \lambda i\ j. j < i$ ]
    by simp
  have sw3:  $(\sum j < h. \sum i \in \{i. i \in \{..<h\} \wedge j < i\}. L\ i\ j) = (\sum i < h. \sum j \in \{j. i < j \wedge j < h\}. L\ i\ j)$ 
    using sym by (auto simp: conj-commute intro: sum.cong)
  show ?thesis using rows sw1 sw2 sw3 unfolding L-def
    by linarith
qed
define S2 where  $S2 = (\sum i < h. \sum j \in \{j. i < j \wedge j < h\}. \ln |t\ i - t\ j|)$ 
define A where  $A = (\sum i < h. \logPot\ \rho\ (t\ i))$ 
define Vs where  $Vs = (\sum i < h. Vf\ (t\ i))$ 
define Ss where  $Ss = (\sum i < h. \sqrt{t\ i})$ 
define I where  $I = \logEnergy\ \rho$ 
define  $\kappa$  where  $\kappa = \text{real } K$ 
have k0:  $0 < \kappa$  unfolding  $\kappa$ -def using assms(2)
  by simp
have  $(\sum i < h. 2 * \logPot\ \rho\ (t\ i) - Vf\ (t\ i) + \sqrt{t\ i} / \text{real } K) \leq (\sum i < h. \text{of-rat } M0c + \sqrt{2} / \text{real } K)$ 
  using fb
  by (blast intro: sum-mono)
hence fbs:  $2 * A - Vs + Ss / \kappa \leq \text{real } h * (\text{of-rat } M0c + \sqrt{2} / \kappa)$ 
  unfolding A-def Vs-def Ss-def  $\kappa$ -def
  by (simp add: sum.distrib sum-subtractf sum-distrib-left sum-divide-distrib)
have fbK:  $2 * A * \kappa - Vs * \kappa + Ss \leq \text{real } h * \text{of-rat } M0c * \kappa + \text{real } h * \sqrt{2}$ 
proof -
  have  $(2 * A - Vs + Ss / \kappa) * \kappa \leq \text{real } h * (\text{of-rat } M0c + \sqrt{2} / \kappa) * \kappa$  using fbs k0
    by (simp add: mult-right-mono)
  thus ?thesis using k0
    by (simp add: algebra-simps)
qed
have cs:  $(\sum i < h. (\logPot\ \rho\ (t\ i) + 80 * \sqrt{\varepsilon})) = A + \text{real } h * (40 / \kappa)$ 
  unfolding A-def sqe  $\kappa$ -def
  by (simp add: sum.distrib)
have ec:  $\logEnergyC\ \sigma\ \varrho C \leq (A + \text{real } h * (40 / \kappa)) / \kappa$ 
  using energy-cross-le unfolding cs  $\kappa$ -def
  by simp
have  $(2 * S2 + \text{real } h * \ln \varepsilon) / \kappa^2 \leq \logEnergyC\ \sigma\ \sigma$ 
  using energy-sig-ge unfolding pair S2-def  $\kappa$ -def
  by (simp add: power2-eq-square)

```

```

hence comb:  $(2 * S2 + \text{real } h * \ln \varepsilon) / \kappa^2 \leq$ 
 $2 * ((A + \text{real } h * (40 / \kappa)) / \kappa) - I$ 
using ec L62 energy-rhoC unfolding I-def
by linarith
have combK:  $2 * S2 + \text{real } h * \ln \varepsilon \leq$ 
 $2 * A * \kappa + 80 * \text{real } h - I * \kappa^2$ 
proof -
have m:  $(2 * S2 + \text{real } h * \ln \varepsilon) / \kappa^2 * \kappa^2 \leq$ 
 $(2 * ((A + \text{real } h * (40 / \kappa)) / \kappa) - I) * \kappa^2$ 
using mult-right-mono[OF comb, of  $\kappa^2$ ] k0 by simp
have e1:  $(2 * S2 + \text{real } h * \ln \varepsilon) / \kappa^2 * \kappa^2 =$ 
 $2 * S2 + \text{real } h * \ln \varepsilon$  using k0
by simp
have e2:  $(2 * ((A + \text{real } h * (40 / \kappa)) / \kappa) - I) * \kappa^2 =$ 
 $2 * A * \kappa + 80 * \text{real } h - I * \kappa^2$ 
using k0 by (simp add: power2-eq-square algebra-simps)
show ?thesis using m e1 e2
by simp
qed
have hl:  $\text{real } h * \ln \varepsilon = -2 * \text{real } h * \ln 2 - 2 * \text{real } h * \ln \kappa$ 
unfolding lne  $\kappa$ -def by (simp add: algebra-simps)
have l2:  $\text{real } h * \ln 2 \leq \text{real } h$  using mult-left-le[of  $\ln 2$   $\text{real } h$ ] ln-2-less-1
by simp
have hM:  $\text{real } h * \text{of-rat } M0c * \kappa = \text{lam-r} * \text{of-rat } M0c * \kappa^2$ 
unfolding hK  $\kappa$ -def
by (simp add: power2-eq-square)
have fin:  $2 * S2 - \kappa * Vs + Ss \leq$ 
 $(\text{lam-r} * \text{of-rat } M0c - I) * \kappa^2 + (120 + \text{sqrt } 2) * \text{real } h + 2 * \text{real } h * \ln \kappa$ 
proof -
have r:  $(\text{lam-r} * \text{of-rat } M0c - I) * \kappa^2 + (120 + \text{sqrt } 2) * \text{real } h + 2 * \text{real } h * \ln \kappa =$ 
 $\text{lam-r} * \text{of-rat } M0c * \kappa^2 - I * \kappa^2 + 120 * \text{real } h + \text{real } h * \text{sqrt } 2 + 2 * \text{real } h * \ln \kappa$ 
by (simp add: algebra-simps)
have v:  $\kappa * Vs = Vs * \kappa$ 
by simp
have h0:  $0 \leq \text{real } h$ 
by simp
show ?thesis unfolding r v using fbK combK hl l2 hM h0
by simp
qed
show ?thesis using fin unfolding S2-def Vs-def Ss-def I-def  $\kappa$ -def h-def .
qed

```

30.4 (6.14), (6.15) and Proposition 6.3

$\int -0^\infty (1+y)^5 e^{-y/\kappa} dy \leq 3872 \kappa^6$ (Lean: *phi-bound*).

lemma *phi-aux*:

fixes $\kappa :: \text{real}$

```

assumes  $1 \leq \kappa$ 
shows set-integrable lborel  $\{0 < ..\}$   $(\lambda y. (1 + y) ^ 5 * \exp (- y / \kappa))$ 
  and set-lebesgue-integral lborel  $\{0 < ..\}$   $(\lambda y. (1 + y) ^ 5 * \exp (- y / \kappa)) \leq$ 
     $3872 * \kappa ^ 6$ 
proof -
  have  $k0: 0 < \kappa$  using assms
    by simp
  have  $c0: 0 < 1 / \kappa$  using  $k0$ 
    by simp
  define  $g0 :: real \Rightarrow real$  where  $g0\ y = y ^ 0 * \exp (- ((1 / \kappa) * y))$  for  $y$ 
  define  $g5 :: real \Rightarrow real$  where  $g5\ y = y ^ 5 * \exp (- ((1 / \kappa) * y))$  for  $y$ 
  define  $\psi :: real \Rightarrow real$  where  $\psi\ y = 32 * g0\ y + 32 * g5\ y$  for  $y$ 
  have  $i0: \text{set-integrable lborel } \{0 < ..\} g0$  unfolding g0-def using gam-integrable[OF c0, of 0]
    by simp
  have  $i5: \text{set-integrable lborel } \{0 < ..\} g5$  unfolding g5-def using gam-integrable[OF c0, of 5]
    by simp
  have  $v0: \text{set-lebesgue-integral lborel } \{0 < ..\} g0 = \kappa$  unfolding g0-def
    using gam[OF c0, of 0] k0
    by simp
  have  $v5: \text{set-lebesgue-integral lborel } \{0 < ..\} g5 = 120 * \kappa ^ 6$  unfolding g5-def
    using gam[OF c0, of 5] k0
    by (simp add: fact-numeral power-one-over)
  have  $ipsi: \text{set-integrable lborel } \{0 < ..\} \psi$  unfolding psi-def using  $i0\ i5$ 
    by simp
  have  $vpsi: \text{set-lebesgue-integral lborel } \{0 < ..\} \psi = 32 * \kappa + 3840 * \kappa ^ 6$ 
    unfolding psi-def using  $i0\ i5\ v0\ v5$ 
    by simp
  have  $p5: (1 + y) ^ 5 \leq 32 * (1 + y ^ 5)$  if  $y: 0 \leq y$  for  $y :: real$ 
proof (cases y ≤ 1)
  case True
    have  $(1 + y) ^ 5 \leq (2 :: real) ^ 5$  using power-mono[of 1 + y 2 5] True y
      by simp
    moreover have  $(2 :: real) ^ 5 = 32$ 
      by simp
    moreover have  $0 \leq y ^ 5$  using  $y$ 
      by simp
    ultimately show ?thesis
      by argo
  next
  case False
    have  $(1 + y) ^ 5 \leq (2 * y) ^ 5$  using power-mono[of 1 + y 2 * y 5] False y
      by simp
    also have  $\dots = 32 * y ^ 5$ 
      by (simp add: power-mult-distrib)
    finally show ?thesis
      by simp
qed

```

```

have bound: (1 + y) ^ 5 * exp (- y / κ) ≤ ψ y if y: 0 < y for y
proof -
  have e: exp (- y / κ) = exp (- ((1 / κ) * y))
    by simp
  have (1 + y) ^ 5 * exp (- y / κ) ≤ 32 * (1 + y ^ 5) * exp (- y / κ)
    using p5[of y] y
    by simp
  also have ... = ψ y unfolding ψ-def g0-def g5-def e
    by (simp add: algebra-simps)
  finally show ?thesis .
qed
have meas: set-borel-measurable lborel {0<..} (λy. (1 + y) ^ 5 * exp (- y / κ))
  unfolding set-borel-measurable-def by simp
have nn: 0 ≤ (1 + y) ^ 5 * exp (- y / κ) if 0 < y for y :: real using that
  by simp
have AE: AE y in lborel. y ∈ {0<..} →
  norm ((1 + y) ^ 5 * exp (- y / κ)) ≤ norm (ψ y)
  using bound nn
  by (force simp: AE-I2)
show I: set-integrable lborel {0<..} (λy. (1 + y) ^ 5 * exp (- y / κ))
  using set-integrable-bound[OF ipsi meas AE] .
have k6: κ ^ 1 ≤ κ ^ 6 using power-increasing[of 1 6 κ] assms
  by simp
have set-lebesgue-integral lborel {0<..} (λy. (1 + y) ^ 5 * exp (- y / κ)) ≤
  set-lebesgue-integral lborel {0<..} ψ
  using set-integral-mono[OF I ipsi] bound
  by simp
also have ... ≤ 3872 * κ ^ 6 unfolding vpsi using k6
  by simp
finally show set-lebesgue-integral lborel {0<..} (λy. (1 + y) ^ 5 * exp (- y /
  κ)) ≤ 3872 * κ ^ 6 .
qed

(6.14): log Δ-K(ζ(5)) ≤ 2h(h + 6N - K) log K + (λM-0 - I(ρ))K^2 +
18h log K + 160h.

theorem log-DeltaK-zeta5-bound:
  assumes 40 dvd K 0 < K
  shows ln (real-eval (DeltaK K) zeta5) ≤
    2 * real (hof K) * (real (hof K) + 6 * real (Nof K) - real K) * ln (real
K)
    + (lam-r * of-rat M0c - logEnergy rho) * real K ^ 2
    + 18 * real (hof K) * ln (real K) + 160 * real (hof K)
proof -
  define h where h = hof K
  define N where N = Nof K
  define E where E = (lam-r * of-rat M0c - logEnergy rho) * real K ^ 2 + (120
+ sqrt 2) * real h +
    2 * real h * ln (real K)
  define c0 where c0 = 8192 * exp 12 * real K ^ (12 * N + 12) / real K ^ (2

```

```

* K)
have hN: 1 ≤ N N ≤ K real N / real K = 3 / 40 0 < h
proof -
  obtain k where k: K = 40 * k using assms(1)
  by auto
  have k1: 1 ≤ k using assms(2) k
  by simp
  have Nk: N = 3 * k unfolding N-def Nof-def k
  by simp
  have hk: h = 37 * k unfolding h-def hof-def k
  by simp
  show 1 ≤ N using Nk k1
  by simp
  show N ≤ K using Nk k
  by simp
  show real N / real K = 3 / 40 using Nk k k1
  by simp
  show 0 < h using hk k1
  by simp
qed

```

Pointwise bound of the Andreief integrand (g -le and (6.9) with $t \cdot i = (y \cdot i / K)^{\wedge 2}$).

```

have pointwise:  $\bigwedge y. (\forall i < h. 0 < y \cdot i) \implies$ 
  ( $\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \cdot i^{\wedge 2} - y \cdot j^{\wedge 2})^{\wedge 2} * (\prod i <$ 
     $h. \text{real-eval } (Dm \ N) (y \cdot i^{\wedge 2})^{\wedge 6} / \text{real-eval } (Dm \ K) (y \cdot i^{\wedge 2}) * \text{weight } (y$ 
     $i))$ 
   $\leq (\text{real } K^{\wedge 4})^{\wedge (h * (h - 1) \text{ div } 2) * c0^{\wedge h} * \exp E * (\prod i <$ 
     $h. (1 + y \cdot i)^{\wedge 5} * \exp (- y \cdot i / \text{real } K))$ 
proof -
  fix y :: nat  $\Rightarrow$  real assume ypos:  $\forall i < h. 0 < y \cdot i$ 
  have k0: 0 < real K using assms(2)
  by simp
  define t where t i = (y i / real K)^ $\wedge 2$  for i
  define Q where Q = ( $\prod i < h. (1 + y \cdot i)^{\wedge 5} * \exp (- y \cdot i / \text{real } K)$ )
  have Q0: 0 ≤ Q unfolding Q-def using ypos
  by (fastforce intro: prod-nonneg)
  have c00: 0 ≤ c0 unfolding c0-def
  by simp
  have R0: 0 ≤ (real K^ $\wedge 4$ )^ $\wedge (h * (h - 1) \text{ div } 2) * c0^{\wedge h} * \exp E * Q$  using
    Q0 c00
  by simp
  have hyt: y i^ $\wedge 2$  = real K^ $\wedge 2$  * t i for i unfolding t-def using k0
  by (simp add: power-divide)
  have Dpos: 0 < real-eval (Dm m) (u^ $\wedge 2$ ) for m u
  proof -
    have D: Dm m = sqPoleProd {1..m}
    by (simp add: Dm-def sqPoleProd-def)
    show ?thesis unfolding D

```



```

    by (auto intro: real-eval-sqPoleProd-pos)
  qed
  have g0:  $0 \leq \text{real-eval } (Dm \ N) \ (u \wedge 2) \wedge 6 / \text{real-eval } (Dm \ K) \ (u \wedge 2) * \text{weight } u$ 
  if  $0 < u$  for  $u$ 
    using Dpos[of  $N \ u$ ] Dpos[of  $K \ u$ ] weight-pos[OF that]
    by (simp add: divide-nonneg-pos)
  show  $(\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \ i \wedge 2 - y \ j \wedge 2) \wedge 2) * (\prod i < h. \text{real-eval } (Dm \ N) \ (y \ i \wedge 2) \wedge 6 / \text{real-eval } (Dm \ K) \ (y \ i \wedge 2) * \text{weight } (y \ i))$ 
     $\leq (\text{real } K \wedge 4) \wedge (h * (h - 1) \text{ div } 2) * c0 \wedge h * \exp E * (\prod i < h. (1 + y \ i) \wedge 5 * \exp (-y \ i / \text{real } K))$ 
  proof (cases inj-on  $t \ \{..<h\}$ )
    case False
    obtain  $i \ j$  where  $ij: i < h \ j < h \ t \ i = t \ j \ i \neq j$  using False
      unfolding inj-on-def
      by auto
    obtain  $a \ b$  where  $ab: a < b \ b < h \ t \ a = t \ b$ 
    proof (cases  $i < j$ )
      case True
      show ?thesis using that[of  $i \ j$ ] True  $ij$ 
        by simp
    next
      case False
      show ?thesis using that[of  $j \ i$ ] False  $ij$ 
        by simp
    qed
  have  $(y \ a \wedge 2 - y \ b \wedge 2) \wedge 2 = 0$  using hyt[of  $a$ ] hyt[of  $b$ ]  $ab(3)$ 
    by (simp add: hyt)
  hence  $(\prod j \in \{j. a < j \wedge j < h\}. (y \ a \wedge 2 - y \ j \wedge 2) \wedge 2) = 0$  using  $ab$ 
    by (auto simp: prod-zero-iff)
  hence  $Z: (\prod i < h. \prod j \in \{j. i < j \wedge j < h\}. (y \ i \wedge 2 - y \ j \wedge 2) \wedge 2) = 0$  using  $ab$ 
    by (auto simp: prod-zero-iff)
  show ?thesis unfolding  $Z$  using  $R0$  unfolding  $Q\text{-def}$ 
    by (simp add: zero-le-mult-iff)
next
  case True
  define  $L$  where  $L \ i \ j = \ln |t \ i - t \ j|$  for  $i \ j$ 
  have hfac:  $(y \ i \wedge 2 - y \ j \wedge 2) \wedge 2 = \text{real } K \wedge 4 * \exp (2 * L \ i \ j)$  if  $ij: i < j$ 
   $j < h$ 
    for  $i \ j$ 
  proof -
    have  $ne: t \ i \neq t \ j$  using True  $ij$  unfolding inj-on-def
      by auto
    have  $l: 2 * \ln |t \ i - t \ j| = \ln (|t \ i - t \ j| \wedge 2)$ 
      using ln-realpow[of  $|t \ i - t \ j| \ 2$ ]
      by simp
    have  $e: \exp (2 * L \ i \ j) = (t \ i - t \ j) \wedge 2$  unfolding  $L\text{-def } l$  using  $ne$ 

```

```

    by simp
  have d: real K ^ 2 * t i - real K ^ 2 * t j = real K ^ 2 * (t i - t j)
    by (simp add: algebra-simps)
  show ?thesis unfolding e hyt d
    by (simp add: power-mult-distrib)
qed
have card: (∑ i<h. card {j. i < j ∧ j < h}) = h * (h - 1) div 2
proof -
  have c1: card {j. i < j ∧ j < h} = h - Suc i for i
  proof -
    have {j. i < j ∧ j < h} = {Suc i..<h}
    by auto
    thus ?thesis
    by simp
  qed
have (∑ i<h. card {j. i < j ∧ j < h}) = (∑ i<h. h - Suc i) unfolding
c1 ..
also have ... = (∑ i<h. i) using sum.nat-diff-reindex[of λi. i h]
  by simp
also have ... = h * (h - 1) div 2
  by (simp add: Sum-Ico-nat atLeast0LessThan[symmetric])
finally show ?thesis .
qed
have hV: (∏ i < h. ∏ j ∈ {j. i < j ∧
  j < h}. (y i ^ 2 - y j ^ 2) ^ 2) = (real K ^ 4) ^ (h * (h - 1) div 2) * exp
(2 * (∑ i <
  h. ∑ j ∈ {j. i < j ∧ j < h}. L i j))
proof -
  have (∏ i < h. ∏ j ∈ {j. i < j ∧
    j < h}. (y i ^ 2 - y j ^ 2) ^ 2) = (∏ i < h. ∏ j ∈ {j. i < j ∧
    j < h}. real K ^ 4 * exp (2 * L i j))
  using hfac
  by simp
  also have ... = (∏ i < h. (real K ^ 4) ^ card {j. i < j ∧
    j < h} * exp (2 * (∑ j ∈ {j. i < j ∧ j < h}. L i j)))
  by (simp add: prod.distrib exp-sum sum-distrib-left)
  also have ... = (real K ^ 4) ^ (∑ i<h. card {j. i < j ∧
    j < h}) * exp (2 * (∑ i < h. ∑ j ∈ {j. i < j ∧ j < h}. L i j))
  by (simp add: power-sum exp-sum sum-distrib-left)
  finally show ?thesis unfolding card .
qed
have hG: (∏ i < h. real-eval (Dm N) (y i ^ 2) ^ 6 / real-eval (Dm K) (y i
^ 2) *
  weight (y i)) ≤ (∏ i < h. c0 * ((1 + y i) ^ 5 * exp (- real K * Vf (t i))))
proof (rule prod-mono)
  fix i assume i ∈ {..<h}
  hence yi: 0 < y i using ypos
  by simp
  show 0 ≤ real-eval (Dm N) (y i ^ 2) ^ 6 / real-eval (Dm K) (y i ^ 2) *

```

$weight (y\ i) \wedge real\text{-}eval\ (Dm\ N)\ (y\ i \wedge 2) \wedge 6 / real\text{-}eval\ (Dm\ K)\ (y\ i \wedge 2) *$
 $weight (y\ i) \leq c0 * ((1 + y\ i) \wedge 5 * exp\ (-\ real\ K * Vf\ (t\ i)))$
using $g0[OF\ yi]\ g\text{-}le[OF\ hN(1,2,3)\ yi]$ **unfolding** $t\text{-}def\ c0\text{-}def$ **by** $simp$
qed
have hG' : $(\prod\ i < h. c0 * ((1 + y\ i) \wedge 5 * exp\ (-\ real\ K * Vf\ (t\ i)))) =$
 $c0 \wedge h * (\prod\ i < h. (1 + y\ i) \wedge 5) * exp\ (-\ real\ K * (\sum\ i < h. Vf\ (t\ i)))$
by $(simp\ add: prod.distrib\ exp\text{-}sum\ sum\text{-}distrib\text{-}left)$
have sq : $\sqrt{t\ i} = y\ i / real\ K$ **if** $i < h$ **for** i **unfolding** $t\text{-}def$ **using** $ypos$
that $k0$
by $auto$
have tnn : $0 \leq t\ i$ **for** i **unfolding** $t\text{-}def$
by $simp$
have $hc0$: $2 * (\sum\ i < h. \sum\ j \in \{j. i < j \wedge j < h\}. L\ i\ j) - real\ K * (\sum\ i < h. Vf\ (t\ i)) + (\sum\ i < h. \sqrt{t\ i}) \leq E$
using $configuration\text{-}bound[OF\ assms(1,2),\ of\ t]$ **unfolding** $h\text{-}def[symmetric]$
 $E\text{-}def\ L\text{-}def$
using $True\ tnn$
by $simp$
have ss : $(\sum\ i < h. \sqrt{t\ i}) = (\sum\ i < h. y\ i / real\ K)$ **using** sq
by $simp$
have $2 * (\sum\ i < h. \sum\ j \in \{j. i < j \wedge j < h\}. L\ i\ j) - real\ K * (\sum\ i < h. Vf\ (t\ i)) + (\sum\ i < h. y\ i / real\ K) \leq$
 E
using $hc0\ ss$
by $simp$
hence $hexp$: $exp\ (2 * (\sum\ i < h. \sum\ j \in \{j. i < j \wedge j < h\}. L\ i\ j)) * exp\ (-\ real\ K * (\sum\ i < h. Vf\ (t\ i))) \leq exp\ E * exp\ (-$
 $(\sum\ i < h. y\ i / real\ K))$
unfolding $mult\text{-}exp\text{-}exp$ **by** $simp$
have ex : $exp\ (-\ (\sum\ i < h. y\ i / real\ K)) = (\prod\ i < h. exp\ (-\ y\ i / real\ K))$
by $(simp\ add: exp\text{-}sum[symmetric]\ sum\text{-}negf[symmetric])$
have Qe : $Q = (\prod\ i < h. (1 + y\ i) \wedge 5) * exp\ (-\ (\sum\ i < h. y\ i / real\ K))$
unfolding $Q\text{-}def\ ex$
by $(simp\ add: prod.distrib)$
have pp : $0 \leq (\prod\ i < h. (1 + y\ i) \wedge 5)$ **using** $ypos$
by $(force\ intro: prod\text{-}nonneg)$
have Vn : $0 \leq (real\ K \wedge 4) \wedge (h * (h - 1) \text{div}\ 2) * exp\ (2 * (\sum\ i < h. \sum\ j$
 $\in$
 $\{j. i < j \wedge j < h\}. L\ i\ j))$
by $simp$
have $P0$: $0 \leq (real\ K \wedge 4) \wedge (h * (h - 1) \text{div}\ 2) * c0 \wedge h * (\prod\ i < h. (1 + y\ i) \wedge 5)$ **using** $c00\ pp$
by $simp$
have $(\prod\ i < h. \prod\ j \in \{j. i < j \wedge j < h\}. (y\ i \wedge 2 - y\ j \wedge 2) \wedge 2) * (\prod\ i < h. real\text{-}eval\ (Dm\ N)\ (y\ i \wedge 2) \wedge 6 / real\text{-}eval\ (Dm\ K)\ (y\ i \wedge 2) * weight\ (y\ i)) =$

```

    (real K ^ 4) ^ (h * (h - 1) div 2) * exp (2 * (∑ i < h. ∑ j ∈ {j. i < j ∧
    j < h}. L i j)) * (∏ i < h. real-eval (Dm N) (y i ^ 2) ^ 6 /
    real-eval (Dm K) (y i ^ 2) * weight (y i))
  unfolding hV ..
  also have ... ≤ (real K ^ 4) ^ (h * (h - 1) div 2) * exp (2 * (∑ i <
  h. ∑ j ∈ {j. i < j ∧
  j < h}. L i j)) * (c0 ^ h * (∏ i < h. (1 + y i) ^ 5) * exp (- real K * (∑
i <
  h. Vf (t i))))
  using mult-left-mono[OF hG[unfolded hG] Vn] by simp
  also have ... = (real K ^ 4) ^ (h * (h - 1) div 2) * c0 ^ h * (∏ i <
  h. (1 + y i) ^ 5) * (exp (2 * (∑ i < h. ∑ j ∈ {j. i < j ∧
  j < h}. L i j)) * exp (- real K * (∑ i < h. Vf (t i))))
  by (simp add: algebra-simps)
  also have ... ≤ (real K ^ 4) ^ (h * (h - 1) div 2) * c0 ^ h * (∏ i <
  h. (1 + y i) ^ 5) * (exp E * exp (- (∑ i < h. y i / real K)))
  using hexp P0
  by (simp add: mult-left-mono)
  also have ... = (real K ^ 4) ^ (h * (h - 1) div 2) * c0 ^ h * exp E * Q
unfolding Qe
  by (simp add: algebra-simps)
  finally show ?thesis unfolding Q-def .
qed
qed
have phi-bound: set-lebesgue-integral lborel {0<..} (λy. (1 + y) ^ 5 * exp (- y
/
  real K)) ≤ 3872 * real K ^ 6
  using phi-aux(2)[of real K] assms(2) by simp
have A: real-eval (DeltaK K) zeta5 = 1 / fact h *
  set-lebesgue-integral (PiM {..<h} (λ-. lborel)) {y. ∀ i < h. 0 < y i}
  (λy. (∏ i < h. ∏ j ∈ {j. i < j ∧
  j < h}. (y i ^ 2 - y j ^ 2) ^ 2) * (∏ i <
  h. real-eval (Dm N) (y i ^ 2) ^ 6 / real-eval (Dm K) (y i ^ 2) * weight
(y i)))
  unfolding h-def N-def by (rule DeltaK-zeta5-integral[OF assms(1)])
have B: set-lebesgue-integral (PiM {..<h} (λ-. lborel)) {y. ∀ i < h. 0 < y i}
  (λy. (∏ i < h. ∏ j ∈ {j. i < j ∧
  j < h}. (y i ^ 2 - y j ^ 2) ^ 2) * (∏ i <
  h. real-eval (Dm N) (y i ^ 2) ^ 6 / real-eval (Dm K) (y i ^ 2) * weight
(y i)))
  ≤ (real K ^ 4) ^ (h * (h - 1) div 2) * c0 ^ h * exp E * (3872 * real K ^
6) ^ h
proof -
  define φ :: real ⇒ real where φ y = (1 + y) ^ 5 * exp (- y / real K)
  for y
  define ψ :: real ⇒ real where ψ y = indicator {0<..} y * φ y
  for y
  define C :: real where C = (real K ^ 4) ^ (h * (h - 1) div 2) * c0 ^ h * exp
E
E

```

```

have c00:  $0 \leq c0$  unfolding c0-def
  by simp
have C0:  $0 \leq C$  unfolding C-def using c00
  by simp
have set-integrable lborel  $\{0 < ..\}$   $\varphi$  unfolding  $\varphi$ -def
  using phi-aux(1)[of real K] assms(2)
  by simp
hence psi-int: integrable lborel  $\psi$  unfolding set-integrable-def  $\psi$ -def
  by simp
have psi-eq:  $\text{integral}^L \text{lborel } \psi = \text{set-lebesgue-integral lborel } \{0 < ..\} \varphi$ 
  unfolding set-lebesgue-integral-def  $\psi$ -def
  by simp
have psi-nn:  $0 \leq \psi \ y$  for  $y$  unfolding  $\psi$ -def  $\varphi$ -def
  by (simp add: indicator-def)
have int-nn:  $0 \leq \text{integral}^L \text{lborel } \psi$  using psi-nn
  by (simp add: integral-nonneg)
have int-le:  $\text{integral}^L \text{lborel } \psi \leq 3872 * \text{real } K \wedge 6$ 
  unfolding psi-eq  $\varphi$ -def using phi-bound
  by simp
interpret PS: product-sigma-finite  $\lambda :: \text{nat. lborel} :: \text{real measure}$ 
  by (auto intro: product-sigma-finite.intro sigma-finite-lborel)
have prodint: integrable (PiM  $\{.. < h\} (\lambda -. \text{lborel})$ )  $(\lambda y. \prod_{i < h.} \psi (y \ i))$  using psi-int PS.product-integrable-prod[of  $\{.. < h\} \lambda -. \psi$ ]
  by simp
have prodeg:  $\text{integral}^L (\text{PiM } \{.. < h\} (\lambda -. \text{lborel})) (\lambda y. \prod_{i < h.} \psi (y \ i)) = \text{integral}^L \text{lborel } \psi \wedge h$ 
  using psi-int PS.product-integral-prod[of  $\{.. < h\} \lambda -. \psi$ ]
  by simp
have intmaj: integrable (PiM  $\{.. < h\} (\lambda -. \text{lborel})$ )  $(\lambda y. C * (\prod_{i < h.} \psi (y \ i)))$  using prodint
  by simp
have nn:  $0 \leq C * (\prod_{i < h.} \psi (y \ i))$  for  $y$  using C0 psi-nn
  by (simp add: prod-nonneg)
have maj: indicator  $\{y. \forall i < h. 0 < y \ i\} y *_R ((\prod_{i < h.} \prod_{j \in \{j. i < j \wedge j < h\}. (y \ i \wedge 2 - y \ j \wedge 2) \wedge 2) * (\prod_{i < h.} \text{real-eval } (Dm \ N) (y \ i \wedge 2) \wedge 6 / \text{real-eval } (Dm \ K) (y \ i \wedge 2) * \text{weight } (y \ i))) \leq$ 
   $C * (\prod_{i < h.} \psi (y \ i))$  for  $y$ 
proof (cases  $\forall i < h. 0 < y \ i$ )
  case True
  have ps:  $(\prod_{i < h.} \psi (y \ i)) = (\prod_{i < h.} (1 + y \ i) \wedge 5 * \exp (- y \ i / \text{real } K))$ 
    using True unfolding  $\psi$ -def  $\varphi$ -def
    by simp
  show ?thesis unfolding ps C-def using pointwise[OF True] True
    by simp
next
case False
show ?thesis using False nn[of y]
  by simp

```

```

qed
have set-lebesgue-integral (PiM {.. $h$ } ( $\lambda \cdot$ . lborel)) { $y$ .  $\forall i < h$ .  $0 <$ 
 $y\ i$ } ( $\lambda y$ . ( $(\prod i < h$ .  $\prod j \in \{j. i < j \wedge$ 
 $j < h\}$ .  $(y\ i^2 - y\ j^2)^2$ ) *  $(\prod i <$ 
 $h$ .  $\text{real-eval } (Dm\ N) (y\ i^2)^6 / \text{real-eval } (Dm\ K) (y\ i^2) * \text{weight } (y$ 
 $i))))$ 
 $\leq \text{integral}^L (PiM \{.. $h$ \} (\lambda \cdot$ . lborel)) ( $\lambda y$ .  $C *$ 
 $(\prod i < h$ .  $\psi (y\ i)))$ 
unfolding set-lebesgue-integral-def using integral-mono'[OF intmaj maj nn]
by simp
also have ... =  $C * \text{integral}^L \text{lborel } \psi \wedge h$  using prodeq
by simp
also have ...  $\leq C * (3872 * \text{real } K^6) \wedge h$  using int-nn int-le C0
by (meson mult-left-mono power-mono)
finally show ?thesis unfolding C-def .
qed
have pos:  $0 < \text{real-eval } (DeltaK\ K) \text{ zeta5}$  by (rule Delta-zeta5-pos)
have fact1:  $1 / \text{fact } h \leq (1 :: \text{real})$ 
by (simp add: divide-le-eq-1)
define I where  $I = \text{set-lebesgue-integral } (PiM \{.. $h$ \} (\lambda \cdot$ . lborel)) { $y$ .  $\forall i < h$ .  $0$ 
 $<$ 
 $y\ i$ }
 $(\lambda y$ . ( $\prod i < h$ .  $\prod j \in \{j. i < j \wedge$ 
 $j < h\}$ .  $(y\ i^2 - y\ j^2)^2$ ) *  $(\prod i <$ 
 $h$ .  $\text{real-eval } (Dm\ N) (y\ i^2)^6 / \text{real-eval } (Dm\ K) (y\ i^2) * \text{weight}$ 
 $(y\ i)))$ 
define LK :: real where  $LK = \ln (\text{real } K)$ 
define P where  $P = h * (h - 1) \text{ div } 2$ 
define R where  $R = (\text{real } K^4) \wedge P * c0 \wedge h * \exp E * (3872 * \text{real } K^6)$ 
 $\wedge h$ 
have A':  $\text{real-eval } (DeltaK\ K) \text{ zeta5} = 1 / \text{fact } h * I$  using A unfolding I-def
.
have B':  $I \leq R$  using B unfolding I-def R-def P-def .
have K1:  $1 \leq \text{real } K$  using assms(2)
by simp
have c0pos:  $0 < c0$  unfolding c0-def using K1
by simp
have I00:  $0 < 1 / \text{fact } h * I$  using pos unfolding A'
by simp
have I0:  $0 < I$  using zero-less-mult-pos[OF I00]
by simp
have DI:  $1 / \text{fact } h * I \leq I$  using mult-right-mono[OF fact1, of I] I0
by simp
have lnR:  $\ln (\text{real-eval } (DeltaK\ K) \text{ zeta5}) \leq \ln R$  using A' DI B' pos
by simp
have lnR-eq:  $\ln R = \text{real } P * (4 * LK) + \text{real } h * \ln c0 + E + \text{real } h * (\ln 3872$ 
 $+ 6 * LK)$ 
unfolding R-def LK-def using assms(2) c0pos
by (simp add: ln-mult ln-realpow distrib-left)

```

```

have p8192: (8192 :: real) = 2 ^ 13
  by simp
have l8192: ln (8192 :: real) = 13 * ln 2
  unfolding p8192 ln-realpow by simp
have lnc0: ln c0 = 13 * ln 2 + 12 + real (12 * N + 12) * LK - real (2 * K)
* LK
  unfolding c0-def LK-def using assms(2) l8192
  by (simp add: ln-div ln-mult ln-realpow)
have ev: even (h * (h - 1))
  by auto
have P * 2 = h * (h - 1) unfolding P-def using ev
  by simp
moreover have real (h * (h - 1)) = real h * real h - real h using hN(4)
  by (simp add: of-nat-diff algebra-simps)
ultimately have hP: real P * 2 = real h * real h - real h
  by linarith
have e4: real P * (4 * LK) = 2 * (real h * real h - real h) * LK
  by (simp add: hP[symmetric])
have l2: ln 2 < (1::real)
  by (simp add: ln-2-less-1)
have p4096: (4096 :: real) = 2 ^ 12
  by simp
have ln 3872 ≤ ln (4096::real)
  by simp
moreover have ln (4096::real) = 12 * ln 2
  unfolding p4096 ln-realpow by simp
ultimately have l3872: ln 3872 ≤ 12 * ln (2::real)
  by simp
have s2: sqrt 2 ≤ (3 / 2 :: real) using real-le-lsqrt[where x = 2 and y = 3 /
2]
  by (simp add: power2-eq-square)
have key: (2 * real h * (real h + 6 * real N - real K) * LK + (lam-r * of-rat
M0c -
logEnergy rho) * real K ^ 2 + 18 * real h * LK + 160 * real h)
- (real P * (4 * LK) + real h * ln c0 + E + real h * (ln 3872 + 6 * LK)) =
real h * (28 - 13 * ln 2 - sqrt 2 - ln 3872)
  unfolding e4 lnc0 E-def unfolding LK-def[symmetric]
  by (simp add: algebra-simps)
have nonneg: 0 ≤ real h * (28 - 13 * ln 2 - sqrt 2 - ln 3872) using l2 l3872
s2
  by simp
have ln (real-eval (DeltaK K) zeta5) ≤
2 * real h * (real h + 6 * real N - real K) * LK + (lam-r * of-rat M0c -
logEnergy rho) *
real K ^ 2 + 18 * real h * LK + 160 * real h
  using lnR lnR-eq key nonneg
  by simp
thus ?thesis unfolding h-def N-def LK-def .
qed

```

(6.15): $\log S-K \leq (2\lambda - 12\alpha\lambda - 2\lambda^2)K^2 \log K + C \cdot K^2 + 6h \log K + 6h$ (factorial bounds; *Stirling-Formula.ln-fact-bounds* is available).

theorem *log-SK-bound*:

assumes $40 \text{ dvd } K$ $0 < K$

shows $\ln (\text{of-rat } (SK \ K)) \leq$

$(2 * \text{lam-r} - 12 * \text{alpha-r} * \text{lam-r} - 2 * \text{lam-r}^2) * \text{real } K^2 * \ln (\text{real } K)$
 $+ Cstar * \text{real } K^2 + 6 * \text{real } (\text{hof } K) * \ln (\text{real } K) + 6 * \text{real } (\text{hof } K)$

proof –

obtain n **where** $K: K = 40 * n$ **and** $n1: 1 \leq n$ **using** *assms* **by** (*auto elim: dvdE*)

have $hh: \text{hof } K = 37 * n$ **and** $hN: \text{Nof } K = 3 * n$ **unfolding** K *hof-def Nof-def* **by** *simp-all*

have *ln-fact-le*: $\bigwedge m. 1 \leq m \implies$

$\ln (\text{fact } m) \leq \text{real } m * \ln (\text{real } m) - \text{real } m + \ln (\text{real } m) + 2$

proof –

fix $m :: \text{nat}$ **assume** $1 \leq m$

thus $\ln (\text{fact } m) \leq \text{real } m * \ln (\text{real } m) - \text{real } m + \ln (\text{real } m) + 2$

proof (*induction m rule: nat-induct-at-least*)

case *base*

show *?case*

by *simp*

next

case (*Suc m*)

define $A :: \text{real}$ **where** $A = \ln (\text{real } m)$

define $B :: \text{real}$ **where** $B = \ln (\text{real } (\text{Suc } m))$

have $m1: 1 \leq \text{real } m$ **using** *Suc.hyps*

by *simp*

have *IH*: $\ln (\text{fact } m) \leq \text{real } m * A - \text{real } m + A + 2$ **using** *Suc.IH* **unfolding** *A-def* .

have $d1: \ln (\text{real } m / \text{real } (\text{Suc } m)) \leq \text{real } m / \text{real } (\text{Suc } m) - 1$

using $m1$ *ln-le-minus-one*[*of real m / real (Suc m)*]

by *simp*

have $d2: \ln (\text{real } m / \text{real } (\text{Suc } m)) = A - B$ **using** $m1$ **unfolding** *A-def B-def*

by (*simp add: ln-div*)

have $d3: \text{real } m / \text{real } (\text{Suc } m) - 1 = -1 / \text{real } (\text{Suc } m)$ **using** $m1$

by (*simp add: field-simps*)

have $d4a: A - B \leq -1 / \text{real } (\text{Suc } m)$ **using** $d1 \ d2 \ d3$

by *simp*

have $d4: \text{real } (\text{Suc } m) * (A - B) \leq -1$ **using** *mult-left-mono*[*OF d4a, of real (Suc m)*] $m1$

by *simp*

have $f: \ln (\text{fact } (\text{Suc } m)) = B + \ln (\text{fact } m)$ **unfolding** *B-def*

by (*simp add: fact-Suc ln-mult*)

have $e: \text{real } (\text{Suc } m) * (A - B) = \text{real } m * A + A - \text{real } (\text{Suc } m) * B$

by (*simp add: algebra-simps*)

have $s: \text{real } (\text{Suc } m) = \text{real } m + 1$

by *simp*


```

    have g:  $\ln (\text{fact } (\text{Suc } m)) \leq \text{real } (\text{Suc } m) * B - \text{real } (\text{Suc } m) + B + 2$  using
e f d4 IH s
    by simp
    show ?case using g unfolding B-def .
  qed
qed
have le-ln-fact:  $\bigwedge m. 1 \leq m \implies$ 
   $\text{real } m * \ln (\text{real } m) - \text{real } m + 1 \leq \ln (\text{fact } m)$ 
proof -
  fix m :: nat assume  $1 \leq m$ 
  thus  $\text{real } m * \ln (\text{real } m) - \text{real } m + 1 \leq \ln (\text{fact } m)$ 
  proof (induction m rule: nat-induct-at-least)
    case base
    show ?case
    by auto
  next
    case (Suc m)
    define A :: real where  $A = \ln (\text{real } m)$ 
    define B :: real where  $B = \ln (\text{real } (\text{Suc } m))$ 
    have m1:  $1 \leq \text{real } m$  using Suc.hyps
    by simp
    have IH:  $\text{real } m * A - \text{real } m + 1 \leq \ln (\text{fact } m)$  using Suc.IH unfolding
A-def .
    have d1:  $\ln (\text{real } (\text{Suc } m) / \text{real } m) \leq \text{real } (\text{Suc } m) / \text{real } m - 1$ 
    using m1 ln-le-minus-one[of  $\text{real } (\text{Suc } m) / \text{real } m$ ]
    by simp
    have d2:  $\ln (\text{real } (\text{Suc } m) / \text{real } m) = B - A$  using m1 unfolding A-def
B-def
    by (simp add: ln-div)
    have d3:  $\text{real } (\text{Suc } m) / \text{real } m - 1 = 1 / \text{real } m$  using m1
    by (simp add: field-simps)
    have d4a:  $B - A \leq 1 / \text{real } m$  using d1 d2 d3
    by simp
    have d4:  $\text{real } m * (B - A) \leq 1$  using mult-left-mono[OF d4a, of  $\text{real } m$ ] m1
    by simp
    have f:  $\ln (\text{fact } (\text{Suc } m)) = B + \ln (\text{fact } m)$  unfolding B-def
    by (simp add: fact-Suc ln-mult)
    have e:  $\text{real } (\text{Suc } m) * B - \text{real } (\text{Suc } m) + 1 = \text{real } m * (B - A) + \text{real } m$ 
* A + B - real m
    by (simp add: algebra-simps)
    have g:  $\text{real } (\text{Suc } m) * B - \text{real } (\text{Suc } m) + 1 \leq \ln (\text{fact } (\text{Suc } m))$  using e f
d4 IH
    by simp
    show ?case using g unfolding B-def .
  qed
qed
have sum-ln-fact-ge:  $\bigwedge h. 1 \leq h \implies$ 
 $\text{real } h ^ 2 * \ln (2 * \text{real } h) - 3 * \text{real } h ^ 2 / 2 - 2 * \text{real } h * \ln (2 * \text{real } h)$ 
 $\leq$ 

```

```

    (∑ i = 1..h - 1. ln (fact (2 * i)))
  proof -
    fix h :: nat assume 1 ≤ h
    thus real h ^ 2 * ln (2 * real h) - 3 * real h ^ 2 / 2 - 2 * real h * ln (2 *
real h) ≤
    (∑ i = 1..h - 1. ln (fact (2 * i)))
  proof (induction h rule: nat-induct-at-least)
    case base
    have l2: 0 ≤ ln (2::real)
    by simp
    have b1: (real 1)2 * ln (2 * real 1) - 3 * (real 1)2 / 2 - 2 * real 1 * ln (2
*
    real 1) = - ln (2::real) - 3 / 2
    by simp
    have b2: (∑ i = 1..1 - 1. ln (fact (2 * i))) = (0::real) by simp
    show ?case unfolding b1 b2 using l2 by linarith
  next
  case (Suc h)
  define A :: real where A = ln (2 * real h)
  define B :: real where B = ln (2 * real (Suc h))
  have h1: 1 ≤ real h using Suc.hyps
  by simp
  have IH: real h ^ 2 * A - 3 * real h ^ 2 / 2 - 2 * real h * A ≤ (∑ i =
1..h - 1. ln (fact (2 * i)))
  using Suc.IH unfolding A-def .
  have hf: 2 * real h * A - 2 * real h + 1 ≤ ln (fact (2 * h))
  using le-ln-fact[of 2 * h] Suc.hyps unfolding A-def
  by simp
  have d1: ln ((2 * real (Suc h)) / (2 * real h)) ≤ (2 * real (Suc h)) / (2 *
real h) - 1
  using h1 ln-le-minus-one[of (2 * real (Suc h)) / (2 * real h)] by simp
  have d2: ln ((2 * real (Suc h)) / (2 * real h)) = B - A using h1 unfolding
A-def B-def
  by (simp add: ln-div)
  have d3: (2 * real (Suc h)) / (2 * real h) - 1 = 1 / real h using h1
  by (simp add: field-simps)
  have d4a: B - A ≤ 1 / real h using d1 d2 d3
  by simp
  have d4: real h * (B - A) ≤ 1 using mult-left-mono[OF d4a, of real h] h1
  by simp
  have d6: 0 ≤ B unfolding B-def
  by simp
  have hh: real h * (real h * (B - A)) ≤ real h using d4 h1
  by (simp add: mult-left-mono)
  obtain k where k: h = Suc k using not0-implies-Suc[of h] Suc.hyps
  by auto
  have split: (∑ i = 1..Suc h - 1. ln (fact (2 * i))) = (∑ i =
1..h - 1. ln (fact (2 * i))) + ln (fact (2 * h))
  unfolding k

```

```

    by simp
    have e: real (Suc h) ^ 2 * B - 3 * real (Suc h) ^ 2 / 2 - 2 * real (Suc h)
* B
      = real h * (real h * (B - A)) + real h ^ 2 * A - B - 3 * real h ^ 2 / 2
- 3 * real h - 3 /
      2
    by (simp add: algebra-simps power2-eq-square)
    have g: real (Suc h) ^ 2 * B - 3 * real (Suc h) ^ 2 / 2 - 2 * real (Suc h)
* B ≤
      (∑ i = 1..Suc h - 1. ln (fact (2 * i)))
    unfolding e split
    proof -
      show real h * (real h * (B - A)) + real h ^ 2 * A - B - 3 * real h ^ 2 /
2 - 3 * real h -
      3 / 2 ≤ (∑ i = 1..h - 1. ln (fact (2 * i))) + ln (fact (2 * h))
      using IH hf hh d6 by simp
    qed
    show ?case using g unfolding B-def .
  qed
qed
have SK: ln (of-rat (SK K)) = 2 * real (hof K) * ln (fact K) + real (hof K -
1) * ln 4 - 12 *
  real (hof K) * ln (fact (Nof K)) - 2 * (∑ i = 1..hof K - 1. ln (fact (2 * i)))
proof -
  have fr: (of-rat (fact m) :: real) = fact m for m using of-rat-of-nat-eq[of fact
m] by simp
  have q: (of-rat (SK K) :: real) = (fact K ^ (2 * hof K) * 4 ^ (hof K - 1)) /
(fact (Nof K) ^ (12 * hof K) * (∏ i = 1..hof K - 1. fact (2 * i) ^ 2))
  unfolding SK-def by (simp add: of-rat-divide of-rat-mult of-rat-power of-rat-prod
fr)
  have pos: 0 < (∏ i = 1..hof K - 1. (fact (2 * i) :: real) ^ 2) by (simp add:
prod-pos)
  have ln (of-rat (SK K)) = ln ((fact K :: real) ^ (2 * hof K)) + ln ((4 :: real)
^ (hof K -
1)) - ln ((fact (Nof K) :: real) ^ (12 * hof K)) - ln (∏ i =
1..hof K - 1. (fact (2 * i) :: real) ^ 2)
  unfolding q using pos by (simp add: ln-div ln-mult)
  also have ln (∏ i = 1..hof K - 1. (fact (2 * i) :: real) ^ 2) = 2 * (∑ i =
1..hof K - 1. ln (fact (2 * i)))
  by (simp add: ln-prod ln-realpow sum-distrib-left)
  finally show ?thesis by (simp add: ln-realpow)
qed
define x :: real where x = real n
define L :: real where L = ln (40 * x)
define F40 :: real where F40 = ln (fact (40 * n))
define F3 :: real where F3 = ln (fact (3 * n))
define S :: real where S = (∑ i = 1..37 * n - 1. ln (fact (2 * i)))
define a :: real where a = ln (3 / 40)
define b :: real where b = ln (37 / 20)

```

```

define l4 :: real where l4 = ln 4
have x1: 1 ≤ x using n1 unfolding x-def by simp
have eN: ln (3 * x) = a + L using x1 ln-mult-pos[of 3 / 40 40 * x] unfolding
a-def L-def
by simp
have eh: ln (2 * (37 * x)) = b + L using x1 ln-mult-pos[of 37 / 20 40 * x]
unfolding b-def L-def by simp
have hA: F40 ≤ 40 * x * L - 40 * x + L + 2 using ln-fact-le[of 40 * n] n1
unfolding F40-def L-def x-def by simp
have hB: 3 * x * (a + L) - 3 * x + 1 ≤ F3 using le-ln-fact[of 3 * n] n1 eN
unfolding F3-def x-def by simp
have hC: (37 * x) ^ 2 * (b + L) - 3 * (37 * x) ^ 2 / 2 - 2 * (37 * x) * (b
+ L) ≤ S
using sum-ln-fact-ge[of 37 * n] n1 eh unfolding S-def x-def L-def by simp
have l4u: l4 ≤ 3 using ln-le-minus-one[of 4 :: real] unfolding l4-def by simp
have l4l: 0 ≤ l4 unfolding l4-def by simp
have bu: b ≤ 1 using ln-le-minus-one[of 37 / 20 :: real] unfolding b-def by
simp
have ca: Cstar = - 2 * (37 / 40) + 12 * (3 / 40) * (37 / 40) * (1 - a) + 3
* (37 / 40) ^ 2 - 2 *
(37 / 40) ^ 2 * b
unfolding Cstar-def lam-r-def alpha-r-def a-def b-def by simp
have key: ((2 * lam-r - 12 * alpha-r * lam-r - 2 * lam-r ^ 2) * (40 * x) ^ 2
* L + Cstar * (40 *
x) ^ 2 + 6 * (37 * x) * L + 6 * (37 * x))
- (2 * (37 * x) * F40 + (37 * x - 1) * l4 - 12 * (37 * x) * F3 - 2 * S)
= 74 * x * (40 * x * L - 40 * x + L + 2 - F40) + 444 * x * (F3 - (3 *
x * (a + L) - 3 * x +
1))
+ 2 * (S - ((37 * x) ^ 2 * (b + L) - 3 * (37 * x) ^ 2 / 2 - 2 * (37 *
x) * (b + L))) + (x *
(518 - 37 * l4 - 148 * b) + l4)
unfolding ca lam-r-def alpha-r-def by (simp add: field-simps power2-eq-square)
have nn1: 0 ≤ 74 * x * (40 * x * L - 40 * x + L + 2 - F40) using hA x1
by simp
have nn2: 0 ≤ 444 * x * (F3 - (3 * x * (a + L) - 3 * x + 1)) using hB x1
by simp
have nn3: 0 ≤ x * (518 - 37 * l4 - 148 * b) using l4l l4u bu x1 by simp
have main: 2 * (37 * x) * F40 + (37 * x - 1) * l4 - 12 * (37 * x) * F3 - 2
* S ≤
(2 * lam-r - 12 * alpha-r * lam-r - 2 * lam-r ^ 2) * (40 * x) ^ 2 * L +
Cstar * (40 * x) ^ 2 +
6 * (37 * x) * L + 6 * (37 * x)
using key nn1 nn2 nn3 hC l4l by simp
have sub: real (37 * n - 1) = 37 * x - 1
using n1 unfolding x-def by simp
show ?thesis
unfolding SK unfolding hh hN unfolding K
using main sub by (simp add: x-def L-def F40-def F3-def S-def l4-def)

```

qed

Proposition 6.3 is at the end of this theory (it needs (6.4) from Appendix A).

lemma *real-eval-FK-eq*: $\text{real-eval } (FK\ K)\ \text{zeta5} = \text{of-rat } (SK\ K) * \text{real-eval } (DeltaK\ K)\ \text{zeta5}$

by (*simp add: FK-def real-eval-def map-poly-smult of-rat-mult*)

Proposition 6.3: $0 < F\text{-}K(\zeta(5))$ and $\log F\text{-}K(\zeta(5)) \leq UbarK^2 + 24K \log K + 200K$. (Needs (6.4) from Appendix A, which is therefore imported.)

theorem *log-FK-zeta5-bound*:

assumes $40 \text{ dvd } K$ $0 < K$

shows $0 < \text{real-eval } (FK\ K)\ \text{zeta5} \wedge$

$\ln (\text{real-eval } (FK\ K)\ \text{zeta5}) \leq$

$\text{of-rat } Ubar * \text{real } K^2 + 24 * \text{real } K * \ln (\text{real } K) + 200 * \text{real } K$

proof –

have $S0$: $0 < (\text{of-rat } (SK\ K) :: \text{real})$ **using** *SK-pos[of K]* **by** *simp*

have $D0$: $0 < \text{real-eval } (DeltaK\ K)\ \text{zeta5}$ **by** (*rule Delta-zeta5-pos*)

have pos : $0 < \text{real-eval } (FK\ K)\ \text{zeta5}$ **unfolding** *real-eval-FK-eq* **using** $S0\ D0$

by *simp*

obtain k **where** k : $K = 40 * k$ **using** *assms(1)* **by** (*elim dvdE*)

have hk : $\text{real } (\text{hof } K) = 37 * \text{real } k$ **and** Nk : $\text{real } (\text{Nof } K) = 3 * \text{real } k$ **and** Kk : $\text{real } K =$

$40 * \text{real } k$

unfolding k *hof-def Nof-def* **by** *simp-all*

define LK **where** $LK = \ln (\text{real } K)$

have $LK0$: $0 \leq LK$ **using** *assms(2)* **by** (*simp add: LK-def*)

have hK : $\text{real } (\text{hof } K) \leq \text{real } K$ **using** $hk\ Kk$ **by** *simp*

define A **where** $A = 2 * \text{real } (\text{hof } K) * (\text{real } (\text{hof } K) + 6 * \text{real } (\text{Nof } K) - \text{real } K) * LK$

define B **where** $B = (2 * \text{lam-r} - 12 * \text{alpha-r} * \text{lam-r} - 2 * \text{lam-r}^2) * \text{real } K^2 * LK$

have AB : $A + B = 0$ **unfolding** $A\text{-def } B\text{-def } hk\ Nk\ Kk$

by (*simp add: lam-r-def alpha-r-def power2-eq-square algebra-simps*)

have L : $\ln (\text{real-eval } (FK\ K)\ \text{zeta5}) = \ln (\text{of-rat } (SK\ K)) + \ln (\text{real-eval } (DeltaK\ K)\ \text{zeta5})$

unfolding *real-eval-FK-eq* **using** $S0\ D0$ **by** (*simp add: ln-mult*)

have $\ln (\text{real-eval } (FK\ K)\ \text{zeta5})$

$\leq (A + (\text{lam-r} * \text{of-rat } M0c - \text{logEnergy } \rho) * \text{real } K^2 + 18 * \text{real } (\text{hof } K) * LK + 160 * \text{real } (\text{hof } K))$

$+ (B + Cstar * \text{real } K^2 + 6 * \text{real } (\text{hof } K) * LK + 6 * \text{real } (\text{hof } K))$

unfolding $L\ A\text{-def } B\text{-def } LK\text{-def}$ **using** *log-DeltaK-zeta5-bound[OF assms]* *log-SK-bound[OF assms]*

by *linarith*

also have $\dots = (A + B) + (\text{lam-r} * \text{of-rat } M0c - \text{logEnergy } \rho + Cstar) * \text{real } K^2$

$+ 24 * (\text{real } (\text{hof } K) * LK) + 166 * \text{real } (\text{hof } K)$

by (*simp add: algebra-simps*)

```

also have ... ≤ of-rat Ubar * real K ^ 2 + 24 * (real K * LK) + 200 * real K
proof -
  have e: (lam-r * of-rat M0c - logEnergy rho + Cstar) * real K ^ 2 ≤
    of-rat Ubar * real K ^ 2
  by (rule mult-right-mono[OF rho-energy-bound]) simp
  have l: real (hof K) * LK ≤ real K * LK by (rule mult-right-mono[OF hK
LK0])
  have h: 166 * real (hof K) ≤ 200 * real K using hK by simp
  show ?thesis unfolding AB using e l h by linarith
qed
finally show ?thesis using pos by (simp add: LK-def mult.assoc)
qed

end

```

```

theory Zeta5-Irrational
imports Zeta5-Archimedean-Bound Zeta5-Exact-Constants HOL-Real-Asymp.Real-Asymp
  Liouville-Numbers.Liouville-Numbers-Misc
begin

```

31 The main theorem: $\zeta(5)$ is irrational

Section 7 of the paper: Theorem 2.1 ($Q_{K,M} \in \mathbb{Z}[X]$ of degree h with $Q_{K,M}(\zeta(5)) > 0$), the limit (7.1), the decay (2.7) for $M = 200$, the elementary irrationality criterion (*irrational-of-int-poly-approx*: integer polynomials Q_n of degree $\leq d$ n with $0 < Q_n(x) < \exp(-c n^2)$ force $x \notin \mathbb{Q}$), and the main theorem *zeta5-irrational*. Lean theories *Section7*, *Main*, *Claim*.

```

lemma real-eval-smult: real-eval (Polynomial.smult c P) x = of-rat c * real-eval
P x
by (simp add: real-eval-def map-poly-smult of-rat-mult)

```

```

lemma real-eval-QKM: real-eval (QKM K M) zeta5 = of-rat (mKM K M) *
real-eval (FK K) zeta5
by (simp add: QKM-def real-eval-smult)

```

```

lemma real-eval-FK: real-eval (FK K) zeta5 = of-rat (SK K) * real-eval (DeltaK
K) zeta5
by (simp add: FK-def real-eval-smult)

```

Theorem 2.1, first part: $Q_{\{K,M\}} \in \mathbb{Z}[X]$, $\deg Q_{\{K,M\}} = h$, $Q_{\{K,M\}}(\zeta(5)) > 0$.

```

theorem QKM-int-degree-pos:
  assumes StdHyp K M
  shows ( $\exists P :: \text{int poly. map-poly of-int } P = QKM K M$ )  $\wedge$ 
    degree (QKM K M) = hof K  $\wedge$ 
    0 < real-eval (QKM K M) zeta5
proof (intro conjI)

```

show $\exists P :: \text{int poly. map-poly of-int } P = \text{QKM } K \ M$ **by** (rule QKM-int-poly[OF assms])
have $hK: 40 \text{ dvd } K$ **using** assms **by** (simp add: StdHyp-def)
have $\text{QKM } K \ M = \text{Polynomial.smult } (mKM \ K \ M * SK \ K) \ (\text{Delta} \ K \ K)$
by (simp add: QKM-def FK-def mult-ac)
thus $\text{degree } (\text{QKM } K \ M) = \text{hof } K$
using $mKM\text{-pos[of } K \ M] \ SK\text{-pos[of } K] \ \text{Delta-degree-lead-coeff[OF } hK]$
by (auto simp: degree-smult-eq)
show $0 < \text{real-eval } (\text{QKM } K \ M) \ \text{zeta5}$
using $mKM\text{-pos } SK\text{-pos } \text{Delta-zeta5-pos}$ **by** (simp add: real-eval-QKM real-eval-FK)
qed

$(24 \ K \log K + 200 \ K) / K^2 \rightarrow 0$ along $K = 40n$ (real-asymp).

lemma error-term-small:

assumes $0 < \varepsilon$
shows $\forall_F n$ in at-top.
 $(24 * \text{real } (40 * n) * \ln (\text{real } (40 * n)) + 200 * \text{real } (40 * n)) / \text{real } (40 * n)^2 \leq \varepsilon$
proof –
define f **where** $f = (\lambda n. (24 * \text{real } (40 * n) * \ln (\text{real } (40 * n)) + 200 * \text{real } (40 * n)) / \text{real } (40 * n)^2)$
have $(f \longrightarrow 0)$ at-top **unfolding** $f\text{-def}$ **by** real-asymp
from tendstoD[OF this assms] **have** $\forall_F n$ in at-top. $f \ n \leq \varepsilon$
by (elim eventually-mono) (auto simp: dist-real-def abs-less-iff)
thus ?thesis **by** (simp add: f-def)
qed

(7.1): $\limsup K^{-2} \log Q\text{-}\{K, M\}(\zeta(5)) \leq A \cdot M + Ubar$ for $M \in 40\mathbb{Z}$, $M \geq 40$.

theorem log-QKM-zeta5-limsup:

assumes $40 \leq M$ $40 \text{ dvd } M$ $0 < \varepsilon$
shows $\forall_F n$ in at-top.
 $\ln (\text{real-eval } (\text{QKM } (40 * n) \ M) \ \text{zeta5}) / \text{real } (40 * n)^2 \leq \text{of-rat } (AM \ M) + \text{of-rat } Ubar + \varepsilon$
proof –
have $A: \forall_F n$ in at-top. $\ln (\text{of-rat } (mKM \ (40 * n) \ M)) / \text{real } (40 * n)^2 \leq \text{of-rat } (AM \ M) + \varepsilon / 2$
by (rule log-mKM-limsup-AM) (use assms in simp-all)
have $B: \forall_F n$ in at-top.
 $(24 * \text{real } (40 * n) * \ln (\text{real } (40 * n)) + 200 * \text{real } (40 * n)) / \text{real } (40 * n)^2 \leq \varepsilon / 2$
by (rule error-term-small) (use assms in simp)
have $C: \forall_F n$ in at-top. $(0 :: \text{nat}) < n$ **by** (rule eventually-gt-at-top)
from $A \ B \ C$ **show** ?thesis
proof eventually-elim
case (elim n)

```

have hK: 40 dvd 40 * n 0 < 40 * n using elim by simp-all
have K0: 0 < real (40 * n) using elim by simp
have Fpos: 0 < real-eval (FK (40 * n)) zeta5
  and Flog: ln (real-eval (FK (40 * n)) zeta5)
    ≤ of-rat Ubar * real (40 * n) ^ 2 + 24 * real (40 * n) * ln (real (40 * n))
+ 200 *
  real (40 * n)
  using log-FK-zeta5-bound[OF hK] by blast+
have ne: mKM (40 * n) M ≠ 0 real-eval (FK (40 * n)) zeta5 ≠ 0
  using mKM-pos[of 40 * n M] Fpos by auto
have ln (real-eval (QKM (40 * n) M) zeta5) =
  ln (of-rat (mKM (40 * n) M)) + ln (real-eval (FK (40 * n)) zeta5)
  unfolding real-eval-QKM by (simp add: ln-mult ne)
hence eq1: ln (real-eval (QKM (40 * n) M) zeta5) / real (40 * n) ^ 2 =
  ln (of-rat (mKM (40 * n) M)) / real (40 * n) ^ 2 + ln (real-eval (FK
(40 * n)) zeta5) /
  real (40 * n) ^ 2
  by (simp add: add-divide-distrib)
have ln (real-eval (FK (40 * n)) zeta5) / real (40 * n) ^ 2
  ≤ (of-rat Ubar * real (40 * n) ^ 2 + 24 * real (40 * n) * ln (real (40 *
n)) + 200 *
  real (40 * n)) / real (40 * n) ^ 2
  by (intro divide-right-mono Flog) simp
also have ... = of-rat Ubar + (24 * real (40 * n) * ln (real (40 * n)) + 200
* real (40 *
  n)) / real (40 * n) ^ 2
  using K0 by (simp add: add-divide-distrib)
finally show ?case using elim eq1 by linarith
qed
qed

```

(2.7): $0 < Q\text{-}\{40n, 200\}(\zeta(5)) < \exp(-139n^2/5)$ for all large n .

theorem QKM-zeta5-decay:

$\forall_F n$ in at-top.

$0 < \text{real-eval} (QKM (40 * n) 200) \text{ zeta5} \wedge$

$\text{real-eval} (QKM (40 * n) 200) \text{ zeta5} < \exp (- (139 / 5 * \text{real } n ^ 2))$

proof –

define dq :: rat **where** dq = 3089837638249482469 / 58872425992515135000

have dpos: $0 < (\text{of-rat } dq :: \text{real})$ **by** (simp add: dq-def of-rat-divide)

have A: $\forall_F n$ in at-top. $\ln (\text{real-eval} (QKM (40 * n) 200) \text{ zeta5}) / \text{real} (40 * n) ^ 2$

$\leq \text{of-rat} (AM 200) + \text{of-rat } Ubar + \text{of-rat } dq / 3200$

by (rule log-QKM-zeta5-limsup) (use dpos in simp-all)

from A eventually-ge-at-top[of 200000 :: nat] **show** ?thesis

proof eventually-elim

case (elim n)

have h: StdHyp (40 * n) 200 **using** elim **by** (auto simp: StdHyp-def)

have pos: $0 < \text{real-eval} (QKM (40 * n) 200) \text{ zeta5}$ **using** QKM-int-degree-pos[OF h] **by** blast


```

have n0: 0 < real n using elim by simp
have K2: real (40 * n) ^ 2 = 1600 * real n ^ 2 by (simp add: power-mult-distrib)
have ln (real-eval (QKM (40 * n) 200) zeta5)
  ≤ (of-rat (AM 200) + of-rat Ubar + of-rat dq / 3200) * real (40 * n) ^
2
  using elim by (subst (asm) pos-divide-le-eq) simp-all
  also have ... = (of-rat (AM 200) + of-rat Ubar + of-rat dq / 3200) * (1600
* real n ^ 2)
  by (simp add: K2)
  also have ... = (- (139 / 5) - of-rat dq / 2) * real n ^ 2
  proof -
    have c: (of-rat (AM 200) + of-rat Ubar + of-rat dq / 3200) * 1600 =
      - (139 / 5) - (of-rat dq :: real) / 2
    unfolding dq-def AM-def Astar-def lam-q-def Ubar-def
    by (simp add: of-rat-divide of-rat-add of-rat-diff of-rat-minus of-rat-mult)
    show ?thesis by (subst mult.assoc[symmetric], subst c, rule refl)
  qed
  also have ... < - (139 / 5 * real n ^ 2)
  proof -
    have 0 < of-rat dq / 2 * real n ^ 2 using dpos n0 by simp
    thus ?thesis by (simp add: algebra-simps)
  qed
  finally have ln (real-eval (QKM (40 * n) 200) zeta5) < - (139 / 5 * real n
^ 2) .
  hence real-eval (QKM (40 * n) 200) zeta5 < exp (- (139 / 5 * real n ^ 2))
  using pos by (metis exp-less-cancel-iff exp-ln)
  thus ?case using pos by simp
qed
qed

```

The abstract top-level argument: if integer polynomials Q - n of degree at most dn satisfy $0 < Q$ - $n(x) < \exp(-cn^2)$ for large n , then x is irrational ($b \nmid dn$ Q - $n(a/b)$ is a positive integer that is eventually < 1).

theorem *irrational-of-int-poly-approx*:

```

fixes x c :: real and Qs :: nat ⇒ int poly
assumes c: 0 < c and deg: ⋀n. degree (Qs n) ≤ d * n
  and ev: ∀F n in at-top. 0 < poly (map-poly of-int (Qs n)) x ∧
  poly (map-poly of-int (Qs n)) x < exp (- (c * real n ^ 2))
shows x ∉ ℚ
proof
  assume x ∈ ℚ
  then obtain a b where b: b > 0 and x: x = of-int a / of-int b by (rule
Rats-cases')
  have b1: (1::real) ≤ of-int b using b by simp
  define N where N = nat ⌈real d * ln (of-int b) / c⌉
  have ∀F n in at-top. N ≤ n by (rule eventually-ge-at-top)
  from this ev have ∀F n in at-top. N ≤ n ∧
    (0 < poly (map-poly of-int (Qs n)) x ∧
    poly (map-poly of-int (Qs n)) x < exp (- (c * real n ^ 2))) by (rule eventu-

```

$ally-conj$)
then obtain M **where** $M: \forall n \geq M. N \leq n \wedge$
 $(0 < poly (map-poly of-int (Qs n)) x \wedge$
 $poly (map-poly of-int (Qs n)) x < exp (- (c * real n ^ 2)))$
by $(auto simp: eventually-at-top-linorder)$
define n **where** $n = M$
have $n: N \leq n$ **and** $pos: 0 < poly (map-poly of-int (Qs n)) x$
and $lt: poly (map-poly of-int (Qs n)) x < exp (- (c * real n ^ 2))$
using M **by** $(auto simp: n-def)$
define Q **where** $Q = (map-poly of-int (Qs n) :: real poly)$
have $Qcoeff: Polynomial.coeff Q i \in \mathbb{Z}$ **for** i **by** $(simp add: Q-def coeff-map-poly)$
have $ne: poly Q (of-int a / of-int b) \neq 0$ **using** pos **by** $(simp add: Q-def x)$
have $1 / of-int b ^ degree Q \leq |poly Q (of-int a / of-int b)|$
using $int-poly-rat-no-root-ge[OF Qcoeff b ne]$ **by** $simp$
also have $|poly Q (of-int a / of-int b)| = poly Q x$ **using** pos **by** $(simp add: Q-def x)$
finally have $lower: 1 / of-int b ^ degree Q \leq poly Q x .$
have $(of-int b :: real) ^ degree Q \leq of-int b ^ (d * n)$
using $b1 deg[of n]$ **by** $(intro power-increasing) (simp-all add: Q-def degree-map-poly)$
hence $lower': (1 :: real) / of-int b ^ (d * n) \leq 1 / of-int b ^ degree Q$
using b **by** $(intro divide-left-mono) (auto intro!: mult-pos-pos)$
have $real d * \ln (of-int b) / c \leq real N$ **by** $(simp add: N-def real-nat-ceiling-ge)$
also have $\dots \leq real n$ **using** n **by** $simp$
finally have $real d * \ln (of-int b) \leq c * real n$ **using** c **by** $(simp add: field-simps)$
hence $real d * \ln (of-int b) * real n \leq c * real n * real n$
by $(intro mult-right-mono) simp-all$
hence $exp (- (c * real n ^ 2)) \leq exp (- (real d * \ln (of-int b) * real n))$
by $(simp add: power2-eq-square)$
also have $e: exp (real d * \ln (of-int b) * real n) = (of-int b :: real) ^ (d * n)$
proof –
have $exp (real d * \ln (of-int b) * real n) = exp (real (d * n) * \ln (of-int b))$
by $(simp add: algebra-simps)$
also have $\dots = exp (\ln (of-int b)) ^ (d * n)$ **by** $(rule exp-of-nat-mult)$
also have $\dots = of-int b ^ (d * n)$ **using** b **by** $simp$
finally show $?thesis .$
qed
have $exp (- (real d * \ln (of-int b) * real n)) = (1 :: real) / of-int b ^ (d * n)$
by $(simp add: exp-minus e inverse-eq-divide)$
finally have $upper: exp (- (c * real n ^ 2)) \leq (1 :: real) / of-int b ^ (d * n) .$
show $False$ **using** $lower lower' lt upper$ **unfolding** $Q-def$ **by** $linarith$
qed

$\zeta(5)$ is real: its Dirichlet series has real terms ($Zeta-Function.sums-zeta-nat$).

lemma $zeta5-real: zeta 5 \in \mathbb{R}$

proof –

have $(\lambda k. Im (1 / of-nat (Suc k) ^ 5 :: complex)) sums Im (zeta 5)$

using $sums-zeta-nat[of 5]$ **by** $(intro sums-Im) simp$

moreover have $Im (1 / of-nat (Suc k) ^ 5 :: complex) = 0$ **for** k

proof –

```

  have (1 / of-nat (Suc k) ^ 5 :: complex) = of-real (1 / real (Suc k) ^ 5) by
simp
  thus ?thesis by (simp only:) simp
qed
ultimately have (λk. 0 :: real) sums Im (zeta 5) by simp
hence Im (zeta 5) = 0 using sums-unique2 sums-zero by blast
thus ?thesis by (simp add: complex-is-Real-iff)
qed

```

```

lemma zeta5-eq: zeta 5 = of-real zeta5
  unfolding zeta5-def using zeta5-real by simp

```

The integer polynomials $Q_{\{40n, 200\}}$ (and 0 for small n).

```

definition Qint :: nat ⇒ int poly where
  Qint n = (if StdHyp (40 * n) 200 then (SOME P. map-poly of-int P = QKM
(40 * n) 200) else 0)

```

```

lemma Qint-spec:
  assumes StdHyp (40 * n) 200
  shows map-poly of-int (Qint n) = QKM (40 * n) 200
  using assms someI-ex[OF conjunct1[OF QKM-int-degree-pos[OF assms]]] by
(simp add: Qint-def)

```

Theorem 1.1 for the real number zeta5.

```

theorem zeta5-irrational-real: zeta5 ∉ ℚ
proof (rule irrational-of-int-poly-approx[where c = 139 / 5 and d = 37 and Qs
= Qint])
  show 0 < (139 / 5 :: real) by simp
  show degree (Qint n) ≤ 37 * n for n
  proof (cases StdHyp (40 * n) 200)
    case True
      have degree (QKM (40 * n) 200) = hof (40 * n) using QKM-int-degree-pos[OF
True] by blast
      moreover have degree (map-poly of-int (Qint n) :: rat poly) = degree (Qint n)
        by (rule degree-map-poly) simp
      ultimately show ?thesis using Qint-spec[OF True] by (simp add: hof-def)
    next
      case False
      thus ?thesis by (simp add: Qint-def)
  qed
  show ∀F n in at-top. 0 < poly (map-poly of-int (Qint n)) zeta5 ∧
    poly (map-poly of-int (Qint n)) zeta5 < exp (− (139 / 5 * real n ^ 2))
  proof −
    have ∀F n in at-top. (200000 :: nat) ≤ n by (rule eventually-ge-at-top)
    with QKM-zeta5-decay show ?thesis
  proof eventually-elim
    case (elim n)
    hence h: StdHyp (40 * n) 200 by (auto simp: StdHyp-def)
    have real-eval (QKM (40 * n) 200) zeta5 = poly (map-poly of-int (Qint n))

```

```

zeta5
  unfolding Quint-spec[OF h, symmetric] real-eval-def
  by (simp add: map-poly-map-poly o-def)
  thus ?case using elim by simp
qed
qed
qed

```

Theorem 1.1, stated for the AFP’s zeta: $\zeta(5) \notin \mathbb{Q}$.

```

theorem zeta5-irrational: zeta 5  $\notin \mathbb{Q}$ 
  using zeta5-irrational-real by (simp add: zeta5-eq)

```

```

corollary Re-zeta5-irrational: Re (zeta 5)  $\notin \mathbb{Q}$ 
  using zeta5-irrational-real by (simp add: zeta5-def)

```

Sanity check of the statement: *zeta 5* is the familiar series $\sum_{n \geq 1} n^{-5}$.

```

lemma zeta5-series: zeta 5 =  $(\sum n. 1 / \text{of-nat } (\text{Suc } n) ^ 5)$ 
  using zeta-nat-conv-suminf[of 5] by simp

```

```

end

```

References

- [1] domino14. zeta5: a Lean 4 formalisation of “ $\zeta(5)$ is irrational”. <https://github.com/dominol4/zeta5>, 2026.
- [2] A. Fauzan. $\zeta(5)$ is irrational. Zenodo preprint, Sept. 2026. <https://zenodo.org/records/22826419>, doi:10.5281/zenodo.22826419.