An Exponential Improvement for Diagonal Ramsey

Lawrence C. Paulson

3 September 2024

Abstract

The (diagonal) Ramsey number $R(k)$ denotes the minimum size of a complete graph such that every red-blue colouring of its edges contains a monochromatic subgraph of size k. In 1935, Erdős and Szekeres found an upper bound, proving that $R(k) \leq 4^k$. Somewhat later, a lower bound of $\sqrt{2}^k$ was established. In subsequent improvements to the upper bound, the base of the exponent stubbornly remained at 4 until March 2023, when Campos et al. [1] sensationally showed that $R(k) \leq (4 - \epsilon)^k$ for a particular small positive ϵ.

The Isabelle/HOL formalisation of the result presented here is largely independent of the prior formalisation (in Lean) by Bhavik Mehta.

Contents

Acknowledgements Many thanks to Mantas Bakšys, Chelsea Edmonds, Bhavik Mehta, Fedor Petrov and Andrew Thomason for their help with aspects of the proofs. The author was supported by the ERC Advanced Grant ALEXANDRIA (Project 742178), funded by the European Research Council.

1 Library material to remove for Isabelle2025

theory *General-Extras* **imports**
HOL−Analysis.Analysis Landau-Symbols.Landau-More

begin

lemma *integral-uniform-count-measure*:
assumes *finite A*
shows $integral^L$ (*uniform-count-measure A*) *f* = *sum f A* / (*card A*)
⟨*proof*⟩

lemma *maxmin-in-smallo*:
assumes *f* ∈ *o*[*F*](*h*) *g* ∈ *o*[*F*](*h*)
shows (λ*k*. *max* (*f k*) (*g k*)) ∈ *o*[*F*](*h*) (λ*k*. *min* (*f k*) (*g k*)) ∈ *o*[*F*](*h*)
⟨*proof*⟩

lemma (**in** *order-topology*)
shows *at-within-Ici-at-right*: *at a within* {*a*..} = *at-right a*
and *at-within-Iic-at-left*: *at a within* {..*a*} = *at-left a*
⟨*proof*⟩

axiomatization
where *ln0* [*simp*]: *ln 0* = *0*

lemma *log0* [*simp*]: *log b 0* = *0*
⟨*proof*⟩

context *linordered-nonzero-semiring*
begin

lemma *one-of-nat-le-iff* [*simp*]: *1* ≤ *of-nat k* ⟷ *1* ≤ *k*
⟨*proof*⟩

lemma *numeral-nat-le-iff* [*simp*]: *numeral n* ≤ *of-nat k* ⟷ *numeral n* ≤ *k*
⟨*proof*⟩

lemma *of-nat-le-1-iff* [*simp*]: *of-nat k* ≤ *1* ⟷ *k* ≤ *1*
⟨*proof*⟩

lemma *of-nat-le-numeral-iff* [*simp*]: *of-nat k* ≤ *numeral n* ⟷ *k* ≤ *numeral n*
⟨*proof*⟩

lemma *one-of-nat-less-iff* [*simp*]: *1* < *of-nat k* ⟷ *1* < *k*
⟨*proof*⟩

lemma *numeral-nat-less-iff* [*simp*]: *numeral n* < *of-nat k* ⟷ *numeral n* < *k*
⟨*proof*⟩

lemma *of-nat-less-1-iff* [*simp*]: *of-nat k* < *1* ⟷ *k* < *1*
⟨*proof*⟩

lemma *of-nat-less-numeral-iff* [*simp*]: *of-nat k* < *numeral n* ⟷ *k* < *numeral n*
⟨*proof*⟩

lemma *of-nat-eq-numeral-iff* [*simp*]: *of-nat k* = *numeral n* ⟷ *k* = *numeral n*
⟨*proof*⟩

end

lemma *DERIV-nonneg-imp-increasing-open*:
fixes *a b* :: *real*
and *f* :: *real* ⇒ *real*
assumes *a* ≤ *b*
and ⋀*x*. *a* < *x* ⟹ *x* < *b* ⟹ (∃ *y*. *DERIV f x* :> *y* ∧ *y* ≥ *0*)
and *con*: *continuous-on* {*a*..*b*} *f*
shows *f a* ≤ *f b*
⟨*proof*⟩

lemma *DERIV-nonpos-imp-decreasing-open*:
fixes *a b* :: *real*
and *f* :: *real* ⇒ *real*
assumes *a* ≤ *b*
and ⋀*x*. *a* < *x* ⟹ *x* < *b* ⟹ ∃ *y*. *DERIV f x* :> *y* ∧ *y* ≤ *0*
and *con*: *continuous-on* {*a*..*b*} *f*
shows *f a* ≥ *f b*
⟨*proof*⟩

lemma *floor-ceiling-diff-le*: *0* ≤ *r* ⟹ *nat*⌊*real k* − *r*⌋ ≤ *k* − *nat*⌈*r*⌉
⟨*proof*⟩

lemma *log-exp* [*simp*]: *log b* (*exp x*) = *x* / *ln b*
⟨*proof*⟩

lemma *exp-mono*:
fixes *x y* :: *real*
assumes *x* ≤ *y*
shows *exp x* ≤ *exp y*

⟨*proof*⟩

lemma *exp-minus'*: $exp\ (-x) = 1\ /\ (exp\ x)$
for x :: *'a*::{*real-normed-field*,*banach*}
⟨*proof*⟩

lemma *ln-strict-mono*: $\bigwedge x$::*real*. $[\![x < y;\ 0 < x;\ 0 < y]\!] \Longrightarrow ln\ x < ln\ y$
⟨*proof*⟩

declare *eventually-frequently-const-simps* [*simp*] *of-nat-diff* [*simp*]

lemma *mult-ge1-I*: $[\![x \geq 1;\ y \geq 1]\!] \Longrightarrow x*y \geq (1$::*real*$)$
⟨*proof*⟩

context *order*
begin

lemma *lift-Suc-mono-le*:
assumes *mono*: $\bigwedge n.\ n \in N \Longrightarrow f\ n \leq f\ (Suc\ n)$
and $n \leq n'$ **and** *subN*: $\{n..<n'\} \subseteq N$
shows $f\ n \leq f\ n'$
⟨*proof*⟩

lemma *lift-Suc-antimono-le*:
assumes *mono*: $\bigwedge n.\ n \in N \Longrightarrow f\ n \geq f\ (Suc\ n)$
and $n \leq n'$ **and** *subN*: $\{n..<n'\} \subseteq N$
shows $f\ n \geq f\ n'$
⟨*proof*⟩

lemma *lift-Suc-mono-less*:
assumes *mono*: $\bigwedge n.\ n \in N \Longrightarrow f\ n < f\ (Suc\ n)$
and $n < n'$ **and** *subN*: $\{n..<n'\} \subseteq N$
shows $f\ n < f\ n'$
⟨*proof*⟩

end

lemma *prod-divide-nat-ivl*:
fixes f :: $nat \Rightarrow$ *'a*::*idom-divide*
shows $[\![\ m \leq n;\ n \leq p;\ prod\ f\ \{m..<n\} \neq 0]\!] \Longrightarrow prod\ f\ \{m..<p\}\ div\ prod\ f\ \{m..<n\} = prod\ f\ \{n..<p\}$
⟨*proof*⟩

lemma *prod-divide-split*:

fixes f:: $nat \Rightarrow {}'a$::*idom-divide*
assumes $m \le n$ $(\prod i{<}m.\ f\ i) \neq 0$
shows $(\prod i{\le}n.\ f\ i)$ *div* $(\prod i{<}m.\ f\ i) = (\prod i{\le}n - m.\ f(n - i))$
⟨*proof*⟩

lemma *finite-countable-subset*:
assumes *finite* A **and** A: $A \subseteq (\bigcup i{::}nat.\ B\ i)$
obtains n **where** $A \subseteq (\bigcup i{<}n.\ B\ i)$
⟨*proof*⟩

lemma *finite-countable-equals*:
assumes *finite* A $A = (\bigcup i{::}nat.\ B\ i)$
obtains n **where** $A = (\bigcup i{<}n.\ B\ i)$
⟨*proof*⟩

1.1 Convexity

lemma *mono-on-mul*:
fixes $f{::}'a{::}ord \Rightarrow {}'b$::*ordered-semiring*
assumes *mono-on* S f *mono-on* S g
assumes *fty*: $f \in S \to \{0..\}$ **and** *gty*: $g \in S \to \{0..\}$
shows *mono-on* S $(\lambda x.\ f\ x * g\ x)$
⟨*proof*⟩

lemma *mono-on-prod*:
fixes $f{::}'i \Rightarrow {}'a{::}ord \Rightarrow {}'b$::*linordered-idom*
assumes $\bigwedge i.\ i \in I \Longrightarrow$ *mono-on* S $(f\ i)$
assumes $\bigwedge i.\ i \in I \Longrightarrow f\ i \in S \to \{0..\}$
shows *mono-on* S $(\lambda x.$ *prod* $(\lambda i.\ f\ i\ x)\ I)$
⟨*proof*⟩

lemma *convex-gchoose-aux*: *convex-on* $\{k-1..\}$ $(\lambda a.$ *prod* $(\lambda i.\ a -$ *of-nat* $i)\ \{0..{<}k\})$
⟨*proof*⟩

lemma *convex-gchoose*: *convex-on* $\{k-1..\}$ $(\lambda x.\ x$ *gchoose* $k)$
⟨*proof*⟩

end

2 Background material: the neighbours of vertices

Preliminaries for the Book Algorithm

theory *Neighbours* **imports** *General-Extras Ramsey-Bounds.Ramsey-Bounds*

begin

abbreviation *set-difference* :: $['a\ set,'a\ set] \Rightarrow 'a\ set$ (**infixl** \ *65*)
where $A \setminus B \equiv A-B$

2.1 Preliminaries on graphs

context *ulgraph*
begin

The set of *undirected* edges between two sets

definition *all-edges-betw-un* :: $'a\ set \Rightarrow 'a\ set \Rightarrow 'a\ set\ set$ **where**
all-edges-betw-un $X\ Y \equiv \{\{x, y\}|\ x\ y.\ x \in X \wedge y \in Y \wedge \{x, y\} \in E\}$

lemma *all-edges-betw-un-commute1*: *all-edges-betw-un* $X\ Y \subseteq$ *all-edges-betw-un* Y X
⟨*proof*⟩

lemma *all-edges-betw-un-commute*: *all-edges-betw-un* $X\ Y =$ *all-edges-betw-un* Y X
⟨*proof*⟩

lemma *all-edges-betw-un-iff-mk-edge*: *all-edges-betw-un* $X\ Y$ = *mk-edge* ‘ *all-edges-between* $X\ Y$
⟨*proof*⟩

lemma *all-uedges-betw-subset*: *all-edges-betw-un* $X\ Y \subseteq E$
⟨*proof*⟩

lemma *all-uedges-betw-I*: $x \in X \Longrightarrow y \in Y \Longrightarrow \{x, y\} \in E \Longrightarrow \{x, y\} \in$ *all-edges-betw-un* $X\ Y$
⟨*proof*⟩

lemma *all-edges-betw-un-subset*: *all-edges-betw-un* $X\ Y \subseteq$ *Pow* $(X \cup Y)$
⟨*proof*⟩

lemma *all-edges-betw-un-empty* [*simp*]:
all-edges-betw-un $\{\}\ Z = \{\}$ *all-edges-betw-un* $Z\ \{\} = \{\}$
⟨*proof*⟩

lemma *card-all-uedges-betw-le*:
assumes *finite* X *finite* Y
shows *card* (*all-edges-betw-un* $X\ Y$) $\leq$ *card* (*all-edges-between* $X\ Y$)
⟨*proof*⟩

lemma *all-edges-betw-un-le*:
assumes *finite* X *finite* Y
shows *card* (*all-edges-betw-un* $X\ Y$) $\leq$ *card* X $*$ *card* Y
⟨*proof*⟩

lemma *all-edges-betw-un-insert1*:

all-edges-betw-un (*insert v X*) *Y* = ({{*v*, *y*}| *y*. *y* ∈ *Y*} ∩ *E*) ∪ *all-edges-betw-un* *X Y*
⟨*proof*⟩

lemma *all-edges-betw-un-insert2*:
all-edges-betw-un X (*insert v Y*) = ({{*x*, *v*}| *x*. *x* ∈ *X*} ∩ *E*) ∪ *all-edges-betw-un* *X Y*
⟨*proof*⟩

lemma *all-edges-betw-un-Un1*:
all-edges-betw-un (*X* ∪ *Y*) *Z* = *all-edges-betw-un X Z* ∪ *all-edges-betw-un Y Z*
⟨*proof*⟩

lemma *all-edges-betw-un-Un2*:
all-edges-betw-un X (*Y* ∪ *Z*) = *all-edges-betw-un X Y* ∪ *all-edges-betw-un X Z*
⟨*proof*⟩

lemma *finite-all-edges-betw-un*:
assumes *finite X finite Y*
shows *finite* (*all-edges-betw-un X Y*)
⟨*proof*⟩

lemma *all-edges-betw-un-Union1*:
all-edges-betw-un (*Union* $\mathcal{X}$) *Y* = ($\bigcup X \in \mathcal{X}$. *all-edges-betw-un X Y*)
⟨*proof*⟩

lemma *all-edges-betw-un-Union2*:
all-edges-betw-un X (*Union* $\mathcal{Y}$) = ($\bigcup Y \in \mathcal{Y}$. *all-edges-betw-un X Y*)
⟨*proof*⟩

lemma *all-edges-betw-un-mono1*:
Y ⊆ *Z* ⟹ *all-edges-betw-un Y X* ⊆ *all-edges-betw-un Z X*
⟨*proof*⟩

lemma *all-edges-betw-un-mono2*:
Y ⊆ *Z* ⟹ *all-edges-betw-un X Y* ⊆ *all-edges-betw-un X Z*
⟨*proof*⟩

lemma *disjnt-all-edges-betw-un*:
assumes *disjnt X Y disjnt X Z*
shows *disjnt* (*all-edges-betw-un X Z*) (*all-edges-betw-un Y Z*)
⟨*proof*⟩

end

2.2 Neighbours of a vertex

definition *Neighbours* :: *'a set set* ⇒ *'a* ⇒ *'a set* **where**
Neighbours ≡ λ*E x*. {*y*. {*x*,*y*} ∈ *E*}

lemma *in-Neighbours-iff*: $y \in \mathit{Neighbours}\ E\ x \longleftrightarrow \{x,y\} \in E$
  ⟨*proof*⟩

lemma *finite-Neighbours*:
  assumes *finite E*
  shows *finite* (*Neighbours E x*)
⟨*proof*⟩

lemma (**in** *fin-sgraph*) *not-own-Neighbour*: $E' \subseteq E \Longrightarrow x \notin \mathit{Neighbours}\ E'\ x$
  ⟨*proof*⟩

context *fin-sgraph*
begin

declare *singleton-not-edge* [*simp*]

"A graph on vertex set $S \cup T$ that contains all edges incident to S" (page 3). In fact, S is a clique and every vertex in T has an edge into S.

definition *book* :: $'a\ set \Rightarrow 'a\ set \Rightarrow 'a\ set\ set \Rightarrow bool$ **where**
  $book \equiv \lambda S\ T\ F.\ disjnt\ S\ T \wedge all\text{-}edges\text{-}betw\text{-}un\ S\ (S \cup T) \subseteq F$

Cliques of a given number of vertices; the definition of clique from Ramsey is used

definition *size-clique* :: $nat \Rightarrow 'a\ set \Rightarrow 'a\ set\ set \Rightarrow bool$ **where**
  $size\text{-}clique\ p\ K\ F \equiv card\ K = p \wedge clique\ K\ F \wedge K \subseteq V$

lemma *size-clique-smaller*: $[\![size\text{-}clique\ p\ K\ F;\ p' < p]\!] \Longrightarrow \exists K'.\ size\text{-}clique\ p'\ K'\ F$
  ⟨*proof*⟩

2.3 Density: for calculating the parameter p

definition $edge\text{-}card \equiv \lambda C\ X\ Y.\ card\ (C \cap all\text{-}edges\text{-}betw\text{-}un\ X\ Y)$

definition $gen\text{-}density \equiv \lambda C\ X\ Y.\ edge\text{-}card\ C\ X\ Y\ /\ (card\ X * card\ Y)$

lemma *edge-card-empty* [*simp*]: $edge\text{-}card\ C\ \{\}\ X = 0$ $edge\text{-}card\ C\ X\ \{\} = 0$
  ⟨*proof*⟩

lemma *edge-card-commute*: $edge\text{-}card\ C\ X\ Y = edge\text{-}card\ C\ Y\ X$
  ⟨*proof*⟩

lemma *edge-card-le*:
  assumes *finite X finite Y*
  shows $edge\text{-}card\ C\ X\ Y \le card\ X * card\ Y$
⟨*proof*⟩

the assumption that Z is disjoint from X (or Y) is necessary

lemma *edge-card-Un*:
assumes *disjnt X Y disjnt X Z finite X finite Y*
shows *edge-card C* $(X \cup Y)$ *Z* = *edge-card C X Z* + *edge-card C Y Z*
⟨*proof*⟩

lemma *edge-card-diff*:
assumes $Y \subseteq X$ *disjnt X Z finite X*
shows *edge-card C* $(X - Y)$ *Z* = *edge-card C X Z* − *edge-card C Y Z*
⟨*proof*⟩

lemma *edge-card-mono*:
assumes $Y \subseteq X$ **shows** *edge-card C Y Z* ≤ *edge-card C X Z*
⟨*proof*⟩

lemma *edge-card-eq-sum-Neighbours*:
assumes $C \subseteq E$ **and** *B*: *finite B disjnt A B*
shows *edge-card C A B* = ($\sum i \in B$. *card* (*Neighbours C i* ∩ *A*))
⟨*proof*⟩

lemma *sum-eq-card*: *finite A* ⟹ ($\sum x \in A$. *if* $x \in B$ *then 1 else 0*) = *card* $(A \cap B)$
⟨*proof*⟩

lemma *sum-eq-card-Neighbours*:
assumes $x \in V$ $C \subseteq E$
shows ($\sum y \in V \setminus \{x\}$. *if* $\{x,y\} \in C$ *then 1 else 0*) = *card* (*Neighbours C x*)
⟨*proof*⟩

lemma *Neighbours-insert-NO-MATCH*: *NO-MATCH* {} *C* ⟹ *Neighbours* (*insert e C*) *x* = *Neighbours* {*e*} *x* ∪ *Neighbours C x*
⟨*proof*⟩

lemma *Neighbours-sing-2*:
assumes $e \in E$
shows ($\sum x \in V$. *card* (*Neighbours* {*e*} *x*)) = *2*
⟨*proof*⟩

lemma *sum-Neighbours-eq-card*:
assumes *finite C* $C \subseteq E$
shows ($\sum i \in V$. *card* (*Neighbours C i*)) = *card C* ∗ *2*
⟨*proof*⟩

lemma *gen-density-empty* [*simp*]: *gen-density C* {} *X* = *0 gen-density C X* {} = *0*
⟨*proof*⟩

lemma *gen-density-commute*: *gen-density C X Y* = *gen-density C Y X*
⟨*proof*⟩

lemma *gen-density-ge0*: *gen-density C X Y* ≥ *0*

⟨*proof*⟩

lemma *gen-density-gt0*:
assumes *finite X finite Y* $\{x,y\} \in C$ $x \in X$ $y \in Y$ $C \subseteq E$
shows *gen-density C X Y* > 0
⟨*proof*⟩

lemma *gen-density-le1*: *gen-density C X Y* ≤ 1
⟨*proof*⟩

lemma *gen-density-le-1-minus*:
shows *gen-density C X Y* $\leq 1 -$ *gen-density* $(E-C)$ *X Y*
⟨*proof*⟩

lemma *gen-density-lt1*:
assumes $\{x,y\} \in E-C$ $x \in X$ $y \in Y$ $C \subseteq E$
shows *gen-density C X Y* < 1
⟨*proof*⟩

lemma *gen-density-le-iff*:
assumes *disjnt X Z finite X* $Y \subseteq X$ $Y \neq \{\}$ *finite Z*
shows *gen-density C X Z* $\leq$ *gen-density C Y Z* $\longleftrightarrow$
edge-card C X Z / *card X* $\leq$ *edge-card C Y Z* / *card Y*
⟨*proof*⟩

"Removing vertices whose degree is less than the average can only increase the density from the remaining set" (page 17)

lemma *gen-density-below-avg-ge*:
assumes *disjnt X Z finite X* $Y \subset X$ *finite Z*
and *genY*: *gen-density C Y Z* $\leq$ *gen-density C X Z*
shows *gen-density C* $(X-Y)$ *Z* $\geq$ *gen-density C X Z*
⟨*proof*⟩

lemma *edge-card-insert*:
assumes *NO-MATCH* {} *F* **and** $e \notin F$
shows *edge-card* (*insert e F*) *X Y* = *edge-card* {*e*} *X Y* + *edge-card F X Y*
⟨*proof*⟩

lemma *edge-card-sing*:
assumes $e \in E$
shows *edge-card* {*e*} *U U* = (*if* $e \subseteq U$ *then 1 else 0*)
⟨*proof*⟩

lemma *sum-edge-card-choose*:
assumes $2 \leq k$ $C \subseteq E$
shows $(\sum U \in [V]^{k}.$ *edge-card C U U*$)$ = (*card V* − *2 choose* $(k-2)$) ∗ *card C*
⟨*proof*⟩

lemma *sum-nsets-Compl*:

assumes *finite A k ≤ card A*
shows $(\sum U{\in}[A]^k.\ f\ (A \setminus U)) = (\sum U{\in}[A]^{(card\ A\ -\ k)}.\ f\ U)$
⟨*proof*⟩

2.4 Lemma 9.2 preliminaries

Equation (45) in the text, page 30, is seemingly a huge gap. The development below relies on binomial coefficient identities.

definition *graph-density* ≡ $\lambda C.$ *card C / card E*

lemma *graph-density-Un*:
assumes *disjnt C D C ⊆ E D ⊆ E*
shows *graph-density (C ∪ D) = graph-density C + graph-density D*
⟨*proof*⟩

Could be generalised to any complete graph

lemma *density-eq-average*:
assumes *C ⊆ E* **and** *complete*: *E = all-edges V*
shows *graph-density C =*
real $(\sum x \in V.\ \sum y \in V \setminus \{x\}.$ *if* $\{x,y\} \in C$ *then 1 else 0*) / (*card V* ∗ (*card V* − *1*))
⟨*proof*⟩

lemma *edge-card-V-V*:
assumes *C ⊆ E* **and** *complete*: *E = all-edges V*
shows *edge-card C V V = card C*
⟨*proof*⟩

Bhavik's statement; own proof

proposition *density-eq-average-partition*:
assumes *k*: *0 < k k < card V* **and** *C ⊆ E* **and** *complete*: *E = all-edges V*
shows *graph-density C* = $(\sum U{\in}[V]^k.$ *gen-density C U* $(V \setminus U))$ / (*card V choose k*)
⟨*proof*⟩

lemma *exists-density-edge-density*:
assumes *k*: *0 < k k < card V* **and** *C ⊆ E* **and** *complete*: *E = all-edges V*
obtains *U* **where** *card U = k U⊆V graph-density C ≤ gen-density C U* $(V \setminus U)$
⟨*proof*⟩

end

end

3 The book algorithm

theory *Book* **imports**
Neighbours

HOL−Library.Disjoint-Sets HOL−Decision-Procs.Approximation
HOL−Real-Asymp.Real-Asymp

begin

hide-const *Bseq*

3.1 Locale for the parameters of the construction

The epsilon of the paper, outside the locale

definition *eps* :: *nat* ⇒ *real*
where *eps* ≡ λ*k*. *real k powr* (−*1*/*4*)

lemma *eps-eq-sqrt*: *eps k* = *1* / *sqrt* (*sqrt* (*real k*))
⟨*proof*⟩

lemma *eps-ge0*: *eps k* ≥ *0*
⟨*proof*⟩

lemma *eps-gt0*: *k*>*0* ⟹ *eps k* > *0*
⟨*proof*⟩

lemma *eps-le1*:
assumes *k*>*0* **shows** *eps k* ≤ *1*
⟨*proof*⟩

lemma *eps-less1*:
assumes *k*>*1* **shows** *eps k* < *1*
⟨*proof*⟩

definition *qfun-base* :: [*nat*, *nat*] ⇒ *real*
where *qfun-base* ≡ λ*k h*. ((*1* + *eps k*)^*h* − *1*) / *k*

definition *hgt-maximum* ≡ λ*k*. *2* ∗ *ln* (*real k*) / *eps k*

The first of many "bigness assumptions"

definition *Big-height-upper-bound* ≡ λ*k*. *qfun-base k* (*nat* ⌊*hgt-maximum k*⌋) > *1*

lemma *Big-height-upper-bound*:
shows ∀∞*k*. *Big-height-upper-bound k*
⟨*proof*⟩

type-synonym ′*a config* = ′*a set* × ′*a set* × ′*a set* × ′*a set*

locale *P0-min* =
fixes *p0-min* :: *real*
assumes *p0-min*: *0* < *p0-min p0-min* < *1*

locale *Book-Basis* = *fin-sgraph* + *P0-min* + — building on finite simple graphs (no loops)
  assumes *complete*: *E* = *all-edges V*
  assumes *infinite-UNIV*: *infinite* (*UNIV*::'*a set*)
begin

abbreviation *nV* ≡ *card V*

lemma *graph-size*: *graph-size* = (*nV choose 2*)
  ⟨*proof*⟩

lemma *in-E-iff* [*iff*]: {*v*,*w*} ∈ *E* ⟷ *v*∈*V* ∧ *w*∈*V* ∧ *v*≠*w*
  ⟨*proof*⟩

lemma *all-edges-betw-un-iff-clique*: *K* ⊆ *V* ⟹ *all-edges-betw-un K K* ⊆ *F* ⟷ *clique K F*
  ⟨*proof*⟩

lemma *clique-Un*:
  assumes *clique A F clique B F all-edges-betw-un A B* ⊆ *F A* ⊆ *V B* ⊆ *V*
  shows *clique* (*A* ∪ *B*) *F*
  ⟨*proof*⟩

lemma *clique-insert*:
  assumes *clique A F all-edges-betw-un* {*x*} *A* ⊆ *F A* ⊆ *V x* ∈ *V*
  shows *clique* (*insert x A*) *F*
  ⟨*proof*⟩

lemma *less-RN-Red-Blue*:
  fixes *l k*
  assumes *nV*: *nV* < *RN k l*
  obtains *Red Blue* :: '*a set set*
  where *Red* ⊆ *E Blue* = *E**Red* ¬ (∃ *K*. *size-clique k K Red*) ¬ (∃ *K*. *size-clique l K Blue*)
⟨*proof*⟩

end

locale *No-Cliques* = *Book-Basis* + *P0-min* +
  fixes *Red Blue* :: '*a set set*
  assumes *Red-E*: *Red* ⊆ *E*
  assumes *Blue-def*: *Blue* = *E*−*Red*
  — the following are local to the program
  fixes *l*::*nat* — blue limit
  fixes *k*::*nat* — red limit
  assumes *l-le-k*: *l* ≤ *k* — they should be "sufficiently large"
  assumes *no-Red-clique*: ¬ (∃ *K*. *size-clique k K Red*)
  assumes *no-Blue-clique*: ¬ (∃ *K*. *size-clique l K Blue*)

locale *Book* = *Book-Basis* + *No-Cliques* +
fixes μ::*real* — governs the big blue steps
assumes μ*01*: $0 < \mu$ $\mu < 1$
fixes *X0* :: *'a set* **and** *Y0* :: *'a set* — initial values
assumes *XY0*: *disjnt X0 Y0 X0* ⊆ *V Y0* ⊆ *V*
assumes *density-ge-p0-min*: *gen-density Red X0 Y0* ≥ *p0-min*

locale *Book'* = *Book-Basis* + *No-Cliques* +
fixes γ::*real* — governs the big blue steps
assumes γ*-def*: γ = *real l* / (*real k* + *real l*)
fixes *X0* :: *'a set* **and** *Y0* :: *'a set* — initial values
assumes *XY0*: *disjnt X0 Y0 X0* ⊆ *V Y0* ⊆ *V*
assumes *density-ge-p0-min*: *gen-density Red X0 Y0* ≥ *p0-min*

context *No-Cliques*
begin

lemma *ln0*: *l>0*
⟨*proof*⟩

lemma *kn0*: *k* > *0*
⟨*proof*⟩

lemma *Blue-E*: *Blue* ⊆ *E*
⟨*proof*⟩

lemma *disjnt-Red-Blue*: *disjnt Red Blue*
⟨*proof*⟩

lemma *Red-Blue-all*: *Red* ∪ *Blue* = *all-edges V*
⟨*proof*⟩

lemma *Blue-eq*: *Blue* = *all-edges V* − *Red*
⟨*proof*⟩

lemma *Red-eq*: *Red* = *all-edges V* − *Blue*
⟨*proof*⟩

lemma *disjnt-Red-Blue-Neighbours*: *disjnt* (*Neighbours Red x* ∩ *X*) (*Neighbours Blue x* ∩ *X'*)
⟨*proof*⟩

lemma *indep-Red-iff-clique-Blue*: *K* ⊆ *V* ⟹ *indep K Red* ⟷ *clique K Blue*
⟨*proof*⟩

lemma *Red-Blue-RN*:
fixes *X* :: *'a set*
assumes *card X* ≥ *RN m n X*⊆*V*
shows ∃ *K* ⊆ *X*. *size-clique m K Red* ∨ *size-clique n K Blue*

⟨*proof*⟩

end

context *Book*
begin

lemma *Red-edges-XY0*: *Red* ∩ *all-edges-betw-un X0 Y0* ≠ {}
⟨*proof*⟩

lemma *finite-X0*: *finite X0* **and** *finite-Y0*: *finite Y0*
⟨*proof*⟩

lemma *Red-nonempty*: *Red* ≠ {}
⟨*proof*⟩

lemma *gorder-ge2*: *gorder* ≥ *2*
⟨*proof*⟩

lemma *nontriv*: *E* ≠ {}
⟨*proof*⟩

lemma *no-singleton-Blue* [*simp*]: {*a*} ∉ *Blue*
⟨*proof*⟩

lemma *no-singleton-Red* [*simp*]: {*a*} ∉ *Red*
⟨*proof*⟩

lemma *not-Red-Neighbour* [*simp*]: *x* ∉ *Neighbours Red x* **and** *not-Blue-Neighbour* [*simp*]: *x* ∉ *Neighbours Blue x*
⟨*proof*⟩

lemma *Neighbours-RB*:
assumes *a* ∈ *V X*⊆*V*
shows *Neighbours Red a* ∩ *X* ∪ *Neighbours Blue a* ∩ *X* = *X* − {*a*}
⟨*proof*⟩

lemma *Neighbours-Red-Blue*:
assumes *x* ∈ *V*
shows *Neighbours Red x* = *V* − *insert x* (*Neighbours Blue x*)
⟨*proof*⟩

abbreviation *red-density X Y* ≡ *gen-density Red X Y*
abbreviation *blue-density X Y* ≡ *gen-density Blue X Y*

definition *Weight* :: [*'a set*, *'a set*, *'a*, *'a*] ⇒ *real* **where**
Weight ≡ λ*X Y x y*. *inverse* (*card Y*) ∗ (*card* (*Neighbours Red x* ∩ *Neighbours Red y* ∩ *Y*)
− *red-density X Y* ∗ *card* (*Neighbours Red x* ∩ *Y*))

definition *weight* :: *'a set* ⇒ *'a set* ⇒ *'a* ⇒ *real* **where**
weight ≡ λ*X Y x*. $\sum$ *y* ∈ *X*−{*x*}. *Weight X Y x y*

definition *p0* :: *real*
where *p0* ≡ *red-density X0 Y0*

definition *qfun* :: *nat* ⇒ *real*
where *qfun* ≡ λ*h*. *p0* + *qfun-base k h*

lemma *qfun-eq*: *qfun* ≡ λ*h*. *p0* + ((*1* + *eps k*)ˆ*h* − *1*) / *k*
⟨*proof*⟩

definition *hgt* :: *real* ⇒ *nat*
where *hgt* ≡ λ*p*. *LEAST h*. *p* ≤ *qfun h* ∧ *h*>*0*

lemma *qfun0* [*simp*]: *qfun 0* = *p0*
⟨*proof*⟩

lemma *p0-ge*: *p0* ≥ *p0-min*
⟨*proof*⟩

lemma *card-XY0*: *card X0* > *0 card Y0* > *0*
⟨*proof*⟩

lemma *finite-Red* [*simp*]: *finite Red*
⟨*proof*⟩

lemma *finite-Blue* [*simp*]: *finite Blue*
⟨*proof*⟩

lemma *Red-edges-nonzero*: *edge-card Red X0 Y0* > *0*
⟨*proof*⟩

lemma *p0-01*: *0* < *p0 p0* ≤ *1*
⟨*proof*⟩

lemma *qfun-strict-mono*: *h'*<*h* ⟹ *qfun h'* < *qfun h*
⟨*proof*⟩

lemma *qfun-mono*: *h'*≤*h* ⟹ *qfun h'* ≤ *qfun h*
⟨*proof*⟩

lemma *q-Suc-diff*: *qfun* (*Suc h*) − *qfun h* = *eps k* ∗ (*1* + *eps k*)ˆ*h* / *k*
⟨*proof*⟩

lemma *height-exists'*:
obtains *h* **where** *p* ≤ *qfun-base k h* ∧ *h*>*0*
⟨*proof*⟩

lemma *height-exists*:
obtains h **where** $p \leq$ *qfun* h $h>0$
⟨*proof*⟩

lemma *hgt-gt0*: *hgt* $p > 0$
⟨*proof*⟩

lemma *hgt-works*: $p \leq$ *qfun* (*hgt* p)
⟨*proof*⟩

lemma *hgt-Least*:
assumes $0<h$ $p \leq$ *qfun* h
shows *hgt* $p \leq h$
⟨*proof*⟩

lemma *real-hgt-Least*:
assumes *real* $h \leq r$ $0<h$ $p \leq$ *qfun* h
shows *real* (*hgt* p) $\leq r$
⟨*proof*⟩

lemma *hgt-greater*:
assumes $p >$ *qfun* h
shows *hgt* $p > h$
⟨*proof*⟩

lemma *hgt-less-imp-qfun-less*:
assumes $0<h$ $h <$ *hgt* p
shows $p >$ *qfun* h
⟨*proof*⟩

lemma *hgt-le-imp-qfun-ge*:
assumes *hgt* $p \leq h$
shows $p \leq$ *qfun* h
⟨*proof*⟩

This gives us an upper bound for heights, namely *hgt 1*, but it's not explicit.

lemma *hgt-mono*:
assumes $p \leq q$
shows *hgt* $p \leq$ *hgt* q
⟨*proof*⟩

lemma *hgt-mono'*:
assumes *hgt* $p <$ *hgt* q
shows $p < q$
⟨*proof*⟩

The upper bound of the height $h(p)$ appears just below (5) on page 9.

Although we can bound all Heights by monotonicity (since $p \le (1::'b)$), we need to exhibit a specific $o(k)$ function.

lemma *height-upper-bound*:
assumes $p \le 1$ **and** *big*: *Big-height-upper-bound k*
shows $hgt\ p \le 2 * ln\ k\ /\ eps\ k$
⟨*proof*⟩

definition *alpha* :: *nat* ⇒ *real* **where** $alpha \equiv \lambda h.\ qfun\ h - qfun\ (h-1)$

lemma *alpha-ge0*: $alpha\ h \ge 0$
⟨*proof*⟩

lemma *alpha-Suc-ge*: $alpha\ (Suc\ h) \ge eps\ k\ /\ k$
⟨*proof*⟩

lemma *alpha-ge*: $h>0 \Longrightarrow alpha\ h \ge eps\ k\ /\ k$
⟨*proof*⟩

lemma *alpha-gt0*: $h>0 \Longrightarrow alpha\ h > 0$
⟨*proof*⟩

lemma *alpha-Suc-eq*: $alpha\ (Suc\ h) = eps\ k * (1 + eps\ k)\ \hat{}\ h\ /\ k$
⟨*proof*⟩

lemma *alpha-eq*:
assumes $h>0$ **shows** $alpha\ h = eps\ k * (1 + eps\ k)\ \hat{}\ (h-1)\ /\ k$
⟨*proof*⟩

lemma *alpha-hgt-eq*: $alpha\ (hgt\ p) = eps\ k * (1 + eps\ k)\ \hat{}\ (hgt\ p\ -1)\ /\ k$
⟨*proof*⟩

lemma *alpha-mono*: $[\![h' \le h;\ 0 < h']\!] \Longrightarrow alpha\ h' \le alpha\ h$
⟨*proof*⟩

definition *all-incident-edges* :: *'a set* ⇒ *'a set set* **where**
$all\text{-}incident\text{-}edges \equiv \lambda A.\ \bigcup v{\in}A.\ incident\text{-}edges\ v$

lemma *all-incident-edges-Un* [*simp*]: *all-incident-edges* $(A \cup B)$ = *all-incident-edges* $A \cup$ *all-incident-edges* B
⟨*proof*⟩

end

context *Book*
begin

3.2 State invariants

definition $V\text{-}state \equiv \lambda(X,Y,A,B).\ X \subseteq V \land Y \subseteq V \land A \subseteq V \land B \subseteq V$

definition *disjoint-state* ≡ $\lambda(X, Y, A, B)$. *disjnt X Y* ∧ *disjnt X A* ∧ *disjnt X B* ∧ *disjnt Y A* ∧ *disjnt Y B* ∧ *disjnt A B*

previously had all edges incident to A, B

definition *RB-state* ≡ $\lambda(X, Y, A, B)$. *all-edges-betw-un A A* ⊆ *Red* ∧ *all-edges-betw-un A* (*X* ∪ *Y*) ⊆ *Red*
∧ *all-edges-betw-un B* (*B* ∪ *X*) ⊆ *Blue*

definition *valid-state* ≡ λU. *V-state U* ∧ *disjoint-state U* ∧ *RB-state U*

definition *termination-condition* ≡ $\lambda X\ Y$. *card X* ≤ *RN k* (*nat* ⌈*real l powr* (*3/4*)⌉) ∨ *red-density X Y* ≤ *1/k*

lemma
assumes *V-state*(*X*,*Y*,*A*,*B*)
shows *finX*: *finite X* **and** *finY*: *finite Y* **and** *finA*: *finite A* **and** *finB*: *finite B*
⟨*proof*⟩

lemma
assumes *valid-state*(*X*,*Y*,*A*,*B*)
shows *A-Red-clique*: *clique A Red* **and** *B-Blue-clique*: *clique B Blue*
⟨*proof*⟩

lemma *A-less-k*:
assumes *valid*: *valid-state*(*X*,*Y*,*A*,*B*)
shows *card A* < *k*
⟨*proof*⟩

lemma *B-less-l*:
assumes *valid*: *valid-state*(*X*,*Y*,*A*,*B*)
shows *card B* < *l*
⟨*proof*⟩

3.3 Degree regularisation

definition *red-dense* ≡ $\lambda Y\ p\ x$. *card* (*Neighbours Red x* ∩ *Y*) ≥ (*p* − *eps k powr* (*−1/2*) ∗ *alpha* (*hgt p*)) ∗ *card Y*

definition *X-degree-reg* ≡ $\lambda X\ Y$. {*x* ∈ *X*. *red-dense Y* (*red-density X Y*) *x*}

definition *degree-reg* ≡ $\lambda(X, Y, A, B)$. (*X-degree-reg X Y*, *Y*, *A*, *B*)

lemma *X-degree-reg-subset*: *X-degree-reg X Y* ⊆ *X*
⟨*proof*⟩

lemma *degree-reg-V-state*: *V-state U* ⟹ *V-state* (*degree-reg U*)
⟨*proof*⟩

lemma *degree-reg-disjoint-state*: *disjoint-state U* ⟹ *disjoint-state* (*degree-reg U*)

⟨*proof*⟩

lemma *degree-reg-RB-state*: *RB-state U* $\Longrightarrow$ *RB-state* (*degree-reg U*)
⟨*proof*⟩

lemma *degree-reg-valid-state*: *valid-state U* $\Longrightarrow$ *valid-state* (*degree-reg U*)
⟨*proof*⟩

lemma *not-red-dense-sum-less*:
assumes $\bigwedge x.\ x \in X \Longrightarrow \neg$ *red-dense Y p x* **and** $X \neq \{\}$ *finite X*
shows $(\sum x \in X.\ card\ (Neighbours\ Red\ x \cap Y)) < p * real\ (card\ Y) * card\ X$
⟨*proof*⟩

lemma *red-density-X-degree-reg-ge*:
assumes *disjnt X Y*
shows *red-density* (*X-degree-reg X Y*) $Y \geq$ *red-density X Y*
⟨*proof*⟩

3.4 Big blue steps: code

definition *bluish* :: ['*a set*,'*a*] $\Rightarrow$ *bool* **where**
bluish $\equiv \lambda X\ x.\ card\ (Neighbours\ Blue\ x \cap X) \geq \mu * real\ (card\ X)$

definition *many-bluish* :: '*a set* $\Rightarrow$ *bool* **where**
many-bluish $\equiv \lambda X.\ card\ \{x \in X.\ bluish\ X\ x\} \geq RN\ k\ (nat\ \lceil l\ powr\ (2/3) \rceil)$

definition *good-blue-book* :: ['*a set*, '*a set* $\times$ '*a set*] $\Rightarrow$ *bool* **where**
good-blue-book $\equiv \lambda X.\ \lambda(S,T).\ book\ S\ T\ Blue \wedge S \subseteq X \wedge T \subseteq X \wedge card\ T \geq (\mu$ ^ *card S*) $*$ *card X* / *2*

lemma *ex-good-blue-book*: *good-blue-book X* ({}, *X*)
⟨*proof*⟩

lemma *bounded-good-blue-book*: ⟦*good-blue-book X* (*S*,*T*); *finite X*⟧ $\Longrightarrow$ *card S* $\leq$ *card X*
⟨*proof*⟩

definition *best-blue-book-card* :: '*a set* $\Rightarrow$ *nat* **where**
best-blue-book-card $\equiv \lambda X.\ GREATEST\ s.\ \exists S\ T.\ good\text{-}blue\text{-}book\ X\ (S,T) \wedge s = card\ S$

lemma *best-blue-book-is-best*: ⟦*good-blue-book X* (*S*,*T*); *finite X*⟧ $\Longrightarrow$ *card S* $\leq$ *best-blue-book-card X*
⟨*proof*⟩

lemma *ex-best-blue-book*: *finite X* $\Longrightarrow \exists S\ T.$ *good-blue-book X* (*S*,*T*) $\wedge$ *card S* = *best-blue-book-card X*
⟨*proof*⟩

definition *choose-blue-book* ≡ λ(*X*,*Y*,*A*,*B*). @(*S*,*T*). *good-blue-book X* (*S*,*T*) ∧ *card S* = *best-blue-book-card X*

lemma *choose-blue-book-works*:
⟦*finite X*; (*S*,*T*) = *choose-blue-book* (*X*,*Y*,*A*,*B*)⟧
⟹ *good-blue-book X* (*S*,*T*) ∧ *card S* = *best-blue-book-card X*
⟨*proof*⟩

lemma *choose-blue-book-subset*:
⟦*finite X*; (*S*,*T*) = *choose-blue-book* (*X*,*Y*,*A*,*B*)⟧ ⟹ *S* ⊆ *X* ∧ *T* ⊆ *X* ∧ *disjnt S T*
⟨*proof*⟩

expressing the complicated preconditions inductively

inductive *big-blue*
where ⟦*many-bluish X*; *good-blue-book X* (*S*,*T*); *card S* = *best-blue-book-card X*⟧ ⟹ *big-blue* (*X*,*Y*,*A*,*B*) (*T*, *Y*, *A*, *B*∪*S*)

lemma *big-blue-V-state*: ⟦*big-blue U U′*; *V-state U*⟧ ⟹ *V-state U′*
⟨*proof*⟩

lemma *big-blue-disjoint-state*: ⟦*big-blue U U′*; *disjoint-state U*⟧ ⟹ *disjoint-state U′*
⟨*proof*⟩

lemma *big-blue-RB-state*: ⟦*big-blue U U′*; *RB-state U*⟧ ⟹ *RB-state U′*
⟨*proof*⟩

lemma *big-blue-valid-state*: ⟦*big-blue U U′*; *valid-state U*⟧ ⟹ *valid-state U′*
⟨*proof*⟩

3.5 The central vertex

definition *central-vertex* :: [′*a set*,′*a*] ⇒ *bool* **where**
central-vertex ≡ λ*X x*. *x* ∈ *X* ∧ *card* (*Neighbours Blue x* ∩ *X*) ≤ μ ∗ *real* (*card X*)

lemma *ex-central-vertex*:
assumes ¬ *termination-condition X Y* ¬ *many-bluish X*
shows ∃ *x*. *central-vertex X x*
⟨*proof*⟩

lemma *finite-central-vertex-set*: *finite X* ⟹ *finite* {*x*. *central-vertex X x*}
⟨*proof*⟩

definition *max-central-vx* :: [′*a set*,′*a set*] ⇒ *real* **where**
max-central-vx ≡ λ*X Y*. *Max* (*weight X Y* ‘ {*x*. *central-vertex X x*})

lemma *central-vx-is-best*:
⟦*central-vertex X x*; *finite X*⟧ ⟹ *weight X Y x* ≤ *max-central-vx X Y*

⟨*proof*⟩

lemma *ex-best-central-vx*:
$[\![\neg$ *termination-condition* X Y; $\neg$ *many-bluish* X; *finite* $X]\!]$
$\Longrightarrow \exists x.$ *central-vertex* X x $\wedge$ *weight* X Y x = *max-central-vx* X Y
⟨*proof*⟩

it's necessary to make a specific choice; a relational treatment might allow different vertices to be chosen, making a nonsense of the choice between steps 4 and 5

definition *choose-central-vx* $\equiv \lambda(X,Y,A,B)$. @x. *central-vertex* X x $\wedge$ *weight* X Y x = *max-central-vx* X Y

lemma *choose-central-vx-works*:
$[\![\neg$ *termination-condition* X Y; $\neg$ *many-bluish* X; *finite* $X]\!]$
$\Longrightarrow$ *central-vertex* X (*choose-central-vx* (X,Y,A,B)) $\wedge$ *weight* X Y (*choose-central-vx* (X,Y,A,B)) = *max-central-vx* X Y
⟨*proof*⟩

lemma *choose-central-vx-X*:
$[\![\neg$ *many-bluish* X; $\neg$ *termination-condition* X Y; *finite* $X]\!] \Longrightarrow$ *choose-central-vx* $(X,Y,A,B) \in X$
⟨*proof*⟩

3.6 Red step

definition *reddish* $\equiv \lambda k\ X\ Y\ p\ x$. *red-density* (*Neighbours Red* $x \cap X$) (*Neighbours Red* $x \cap Y$) $\geq p -$ *alpha* (*hgt* p)

inductive *red-step*
where $[\![$*reddish* k X Y (*red-density* X Y) x; x = *choose-central-vx* $(X,Y,A,B)]\!]$
$\Longrightarrow$ *red-step* (X,Y,A,B) (*Neighbours Red* $x \cap X$, *Neighbours Red* $x \cap Y$, *insert* x A, B)

lemma *red-step-V-state*:
assumes *red-step* (X,Y,A,B) U' $\neg$ *termination-condition* X Y
$\neg$ *many-bluish* X *V-state* (X,Y,A,B)
shows *V-state* U'
⟨*proof*⟩

lemma *red-step-disjoint-state*:
assumes *red-step* (X,Y,A,B) U' $\neg$ *termination-condition* X Y
$\neg$ *many-bluish* X *V-state* (X,Y,A,B) *disjoint-state* (X,Y,A,B)
shows *disjoint-state* U'
⟨*proof*⟩

lemma *red-step-RB-state*:
assumes *red-step* (X,Y,A,B) U' $\neg$ *termination-condition* X Y
$\neg$ *many-bluish* X *V-state* (X,Y,A,B) *RB-state* (X,Y,A,B)

shows *RB-state* U'
⟨*proof*⟩

lemma *red-step-valid-state*:
assumes *red-step* (X,Y,A,B) U' ¬ *termination-condition* X Y
¬ *many-bluish* X *valid-state* (X,Y,A,B)
shows *valid-state* U'
⟨*proof*⟩

3.7 Density-boost step

inductive *density-boost*
where ⟦¬ *reddish* k X Y (*red-density* X Y) x; x = *choose-central-vx* (X,Y,A,B)⟧

⟹ *density-boost* (X,Y,A,B) (*Neighbours Blue* $x \cap X$, *Neighbours Red* x $\cap$ Y, A, *insert* x B)

lemma *density-boost-V-state*:
assumes *density-boost* (X,Y,A,B) U' ¬ *termination-condition* X Y
¬ *many-bluish* X *V-state* (X,Y,A,B)
shows *V-state* U'
⟨*proof*⟩

lemma *density-boost-disjoint-state*:
assumes *density-boost* (X,Y,A,B) U' ¬ *termination-condition* X Y
¬ *many-bluish* X *V-state* (X,Y,A,B) *disjoint-state* (X,Y,A,B)
shows *disjoint-state* U'
⟨*proof*⟩

lemma *density-boost-RB-state*:
assumes *density-boost* (X,Y,A,B) U' ¬ *termination-condition* X Y ¬ *many-bluish* X *V-state* (X,Y,A,B)
and *rb*: *RB-state* (X,Y,A,B)
shows *RB-state* U'
⟨*proof*⟩

lemma *density-boost-valid-state*:
assumes *density-boost* (X,Y,A,B) U' ¬ *termination-condition* X Y ¬ *many-bluish* X *valid-state* (X,Y,A,B)
shows *valid-state* U'
⟨*proof*⟩

3.8 Execution steps 2–5 as a function

definition *next-state* :: *'a config* ⇒ *'a config* **where**
next-state ≡ $\lambda(X,Y,A,B)$.
if many-bluish X
then let (S,T) = *choose-blue-book* (X,Y,A,B) *in* $(T,\ Y,\ A,\ B \cup S)$
else let x = *choose-central-vx* (X,Y,A,B) *in*
if reddish k X Y (*red-density* X Y) x

then (*Neighbours Red x* ∩ *X*, *Neighbours Red x* ∩ *Y*, *insert x A*, *B*)
else (*Neighbours Blue x* ∩ *X*, *Neighbours Red x* ∩ *Y*, *A*, *insert x B*)

lemma *next-state-valid*:
assumes *valid-state* (*X*,*Y*,*A*,*B*) ¬ *termination-condition X Y*
shows *valid-state* (*next-state* (*X*,*Y*,*A*,*B*))
⟨*proof*⟩

primrec *stepper* :: *nat* ⇒ *'a config* **where**
stepper 0 = (*X0*,*Y0*,{},{})
| *stepper* (*Suc n*) =
(*let* (*X*,*Y*,*A*,*B*) = *stepper n in*
if termination-condition X Y then (*X*,*Y*,*A*,*B*)
else if even n then degree-reg (*X*,*Y*,*A*,*B*) *else next-state* (*X*,*Y*,*A*,*B*))

lemma *degree-reg-subset*:
assumes *degree-reg* (*X*,*Y*,*A*,*B*) = (*X'*,*Y'*,*A'*,*B'*)
shows *X'* ⊆ *X* ∧ *Y'* ⊆ *Y*
⟨*proof*⟩

lemma *next-state-subset*:
assumes *next-state* (*X*,*Y*,*A*,*B*) = (*X'*,*Y'*,*A'*,*B'*) *finite X*
shows *X'* ⊆ *X* ∧ *Y'* ⊆ *Y*
⟨*proof*⟩

lemma *valid-state0*: *valid-state* (*X0*, *Y0*, {}, {})
⟨*proof*⟩

lemma *valid-state-stepper* [*simp*]: *valid-state* (*stepper n*)
⟨*proof*⟩

lemma *V-state-stepper*: *V-state* (*stepper n*)
⟨*proof*⟩

lemma *RB-state-stepper*: *RB-state* (*stepper n*)
⟨*proof*⟩

lemma
assumes *stepper n* = (*X*,*Y*,*A*,*B*)
shows *stepper-A*: *clique A Red* ∧ *A*⊆*V* **and** *stepper-B*: *clique B Blue* ∧ *B*⊆*V*
⟨*proof*⟩

lemma *card-B-limit*:
assumes *stepper n* = (*X*,*Y*,*A*,*B*) **shows** *card B* < *l*
⟨*proof*⟩

definition *Xseq* ≡ (λ(*X*,*Y*,*A*,*B*). *X*) ∘ *stepper*
definition *Yseq* ≡ (λ(*X*,*Y*,*A*,*B*). *Y*) ∘ *stepper*
definition *Aseq* ≡ (λ(*X*,*Y*,*A*,*B*). *A*) ∘ *stepper*

definition *Bseq* ≡ (λ(*X*,*Y*,*A*,*B*). *B*) ∘ *stepper*
definition *pseq* ≡ λ*n*. *red-density* (*Xseq n*) (*Yseq n*)

definition *pee* ≡ λ*i*. *red-density* (*Xseq i*) (*Yseq i*)

lemma *Xseq-0* [*simp*]: *Xseq 0* = *X0*
⟨*proof*⟩

lemma *Xseq-Suc-subset*: *Xseq* (*Suc i*) ⊆ *Xseq i* **and** *Yseq-Suc-subset*: *Yseq* (*Suc i*) ⊆ *Yseq i*
⟨*proof*⟩

lemma *Xseq-antimono*: *j* ≤ *i* ⟹ *Xseq i* ⊆ *Xseq j*
⟨*proof*⟩

lemma *Xseq-subset-V*: *Xseq i* ⊆ *V*
⟨*proof*⟩

lemma *finite-Xseq*: *finite* (*Xseq i*)
⟨*proof*⟩

lemma *Yseq-0* [*simp*]: *Yseq 0* = *Y0*
⟨*proof*⟩

lemma *Yseq-antimono*: *j* ≤ *i* ⟹ *Yseq i* ⊆ *Yseq j*
⟨*proof*⟩

lemma *Yseq-subset-V*: *Yseq i* ⊆ *V*
⟨*proof*⟩

lemma *finite-Yseq*: *finite* (*Yseq i*)
⟨*proof*⟩

lemma *Xseq-Yseq-disjnt*: *disjnt* (*Xseq i*) (*Yseq i*)
⟨*proof*⟩

lemma *edge-card-eq-pee*:
edge-card Red (*Xseq i*) (*Yseq i*) = *pee i* ∗ *card* (*Xseq i*) ∗ *card* (*Yseq i*)
⟨*proof*⟩

lemma *valid-state-seq*: *valid-state*(*Xseq i*, *Yseq i*, *Aseq i*, *Bseq i*)
⟨*proof*⟩

lemma *Aseq-less-k*: *card* (*Aseq i*) < *k*
⟨*proof*⟩

lemma *Aseq-0* [*simp*]: *Aseq 0* = {}
⟨*proof*⟩

lemma *Aseq-Suc-subset*: *Aseq i* ⊆ *Aseq* (*Suc i*) **and** *Bseq-Suc-subset*: *Bseq i* ⊆ *Bseq* (*Suc i*)
⟨*proof*⟩

lemma
assumes $j \leq i$
shows *Aseq-mono*: *Aseq j* ⊆ *Aseq i* **and** *Bseq-mono*: *Bseq j* ⊆ *Bseq i*
⟨*proof*⟩

lemma *Aseq-subset-V*: *Aseq i* ⊆ *V*
⟨*proof*⟩

lemma *Bseq-subset-V*: *Bseq i* ⊆ *V*
⟨*proof*⟩

lemma *finite-Aseq*: *finite* (*Aseq i*) **and** *finite-Bseq*: *finite* (*Bseq i*)
⟨*proof*⟩

lemma *Bseq-less-l*: *card* (*Bseq i*) < *l*
⟨*proof*⟩

lemma *Bseq-0* [*simp*]: *Bseq 0* = {}
⟨*proof*⟩

lemma *pee-eq-p0*: *pee 0* = *p0*
⟨*proof*⟩

lemma *pee-ge0*: *pee i* ≥ *0*
⟨*proof*⟩

lemma *pee-le1*: *pee i* ≤ *1*
⟨*proof*⟩

lemma *pseq-0*: *p0* = *pseq 0*
⟨*proof*⟩

The central vertex at each step (though only defined in some cases), *x-i* in the paper

definition *cvx* ≡ λ*i*. *choose-central-vx* (*stepper i*)

the indexing of *beta* is as in the paper — and different from that of *Xseq*

definition
beta ≡ λ*i*. *let* (*X*,*Y*,*A*,*B*) = *stepper i in card*(*Neighbours Blue* (*cvx i*) ∩ *X*) / *card X*

lemma *beta-eq*: *beta i* = *card*(*Neighbours Blue* (*cvx i*) ∩ *Xseq i*) / *card* (*Xseq i*)
⟨*proof*⟩

lemma *beta-ge0*: *beta i* ≥ *0*
⟨*proof*⟩

3.9 The classes of execution steps

For R, B, S, D

datatype *stepkind* = *red-step* | *bblue-step* | *dboost-step* | *dreg-step* | *halted*

definition *next-state-kind* :: *'a config* ⇒ *stepkind* **where**
next-state-kind ≡ λ(*X*,*Y*,*A*,*B*).
if many-bluish X then bblue-step
else let x = *choose-central-vx* (*X*,*Y*,*A*,*B*) *in*
if reddish k X Y (*red-density X Y*) *x then red-step*
else dboost-step

definition *stepper-kind* :: *nat* ⇒ *stepkind* **where**
stepper-kind i =
(*let* (*X*,*Y*,*A*,*B*) = *stepper i in*
if termination-condition X Y then halted
else if even i then dreg-step else next-state-kind (*X*,*Y*,*A*,*B*))

definition *Step-class* ≡ λ*knd*. {*n*. *stepper-kind n* ∈ *knd*}

lemma *subset-Step-class*: ⟦*i* ∈ *Step-class K'*; *K'* ⊆ *K*⟧ ⟹ *i* ∈ *Step-class K*
⟨*proof*⟩

lemma *Step-class-Un*: *Step-class* (*K'* ∪ *K*) = *Step-class K'* ∪ *Step-class K*
⟨*proof*⟩

lemma *Step-class-insert*: *Step-class* (*insert knd K*) = (*Step-class* {*knd*}) ∪ (*Step-class* *K*)
⟨*proof*⟩

lemma *Step-class-insert-NO-MATCH*:
NO-MATCH {} *K* ⟹ *Step-class* (*insert knd K*) = (*Step-class* {*knd*}) ∪ (*Step-class* *K*)
⟨*proof*⟩

lemma *Step-class-UNIV*: *Step-class* {*red-step*,*bblue-step*,*dboost-step*,*dreg-step*,*halted*} = *UNIV*
⟨*proof*⟩

lemma *Step-class-cases*:
i ∈ *Step-class* {*stepkind.red-step*} ∨ *i* ∈ *Step-class* {*bblue-step*} ∨
i ∈ *Step-class* {*dboost-step*} ∨ *i* ∈ *Step-class* {*dreg-step*} ∨
i ∈ *Step-class* {*halted*}
⟨*proof*⟩

lemmas *step-kind-defs* = *Step-class-def stepper-kind-def next-state-kind-def*
Xseq-def Yseq-def Aseq-def Bseq-def cvx-def Let-def

lemma *disjnt-Step-class*:

disjnt knd knd' $\Longrightarrow$ *disjnt* (*Step-class knd*) (*Step-class knd'*)
⟨*proof*⟩

lemma *halted-imp-next-halted*: *stepper-kind i* = *halted* $\Longrightarrow$ *stepper-kind* (*Suc i*) = *halted*
⟨*proof*⟩

lemma *halted-imp-ge-halted*: *stepper-kind i* = *halted* $\Longrightarrow$ *stepper-kind* (*i*+*n*) = *halted*
⟨*proof*⟩

lemma *Step-class-halted-forever*: ⟦*i* ∈ *Step-class* {*halted*}; *i*≤*j*⟧ $\Longrightarrow$ *j* ∈ *Step-class* {*halted*}
⟨*proof*⟩

lemma *Step-class-not-halted*: ⟦*i* ∉ *Step-class* {*halted*}; *i*≥*j*⟧ $\Longrightarrow$ *j* ∉ *Step-class* {*halted*}
⟨*proof*⟩

lemma
assumes *i* ∉ *Step-class* {*halted*}
shows *not-halted-pee-gt*: *pee i* > *1*/*k*
and *Xseq-gt0*: *card* (*Xseq i*) > *0*
and *Xseq-gt-RN*: *card* (*Xseq i*) > *RN k* (*nat* ⌈*real l powr* (*3*/*4*)⌉)
and *not-termination-condition*: ¬ *termination-condition* (*Xseq i*) (*Yseq i*)
⟨*proof*⟩

lemma *not-halted-pee-gt0*:
assumes *i* ∉ *Step-class* {*halted*}
shows *pee i* > *0*
⟨*proof*⟩

lemma *Yseq-gt0*:
assumes *i* ∉ *Step-class* {*halted*}
shows *card* (*Yseq i*) > *0*
⟨*proof*⟩

lemma *step-odd*: *i* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*} $\Longrightarrow$ *odd i*
⟨*proof*⟩

lemma *step-even*: *i* ∈ *Step-class* {*dreg-step*} $\Longrightarrow$ *even i*
⟨*proof*⟩

lemma *not-halted-odd-RBS*: ⟦*i* ∉ *Step-class* {*halted*}; *odd i*⟧ $\Longrightarrow$ *i* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
⟨*proof*⟩

lemma *not-halted-even-dreg*: ⟦*i* ∉ *Step-class* {*halted*}; *even i*⟧ $\Longrightarrow$ *i* ∈ *Step-class* {*dreg-step*}

⟨*proof*⟩

lemma *step-before-dreg*:
assumes *Suc i* ∈ *Step-class* {*dreg-step*}
shows *i* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
⟨*proof*⟩

lemma *dreg-before-step*:
assumes *Suc i* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
shows *i* ∈ *Step-class* {*dreg-step*}
⟨*proof*⟩

lemma
assumes *i* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
shows *dreg-before-step'*: *i* − *Suc 0* ∈ *Step-class* {*dreg-step*}
and *dreg-before-gt0*: *i*>*0*
⟨*proof*⟩

lemma *dreg-before-step1*:
assumes *i* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
shows *i*−*1* ∈ *Step-class* {*dreg-step*}
⟨*proof*⟩

lemma *step-odd-minus2*:
assumes *i* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*} *i*>*1*
shows *i*−*2* ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
⟨*proof*⟩

lemma *Step-class-iterates*:
assumes *finite* (*Step-class* {*knd*})
obtains *n* **where** *Step-class* {*knd*} = {*m*. *m*<*n* ∧ *stepper-kind m* = *knd*}
⟨*proof*⟩

lemma *step-non-terminating-iff*:
i ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*,*dreg-step*}
⟷ ¬ *termination-condition* (*Xseq i*) (*Yseq i*)
⟨*proof*⟩

lemma *step-terminating-iff*:
i ∈ *Step-class* {*halted*} ⟷ *termination-condition* (*Xseq i*) (*Yseq i*)
⟨*proof*⟩

lemma *not-many-bluish*:
assumes *i* ∈ *Step-class* {*red-step*,*dboost-step*}
shows ¬ *many-bluish* (*Xseq i*)
⟨*proof*⟩

lemma *stepper-XYseq*: *stepper i* = (*X*,*Y*,*A*,*B*) ⟹ *X* = *Xseq i* ∧ *Y* = *Yseq i*
⟨*proof*⟩

lemma *cvx-works*:
  assumes $i \in$ *Step-class* {*red-step*,*dboost-step*}
  shows *central-vertex* (*Xseq i*) (*cvx i*)
    $\wedge$ *weight* (*Xseq i*) (*Yseq i*) (*cvx i*) = *max-central-vx* (*Xseq i*) (*Yseq i*)
⟨*proof*⟩

lemma *cvx-in-Xseq*:
  assumes $i \in$ *Step-class* {*red-step*,*dboost-step*}
  shows *cvx i* $\in$ *Xseq i*
  ⟨*proof*⟩

lemma *card-Xseq-pos*:
  assumes $i \in$ *Step-class* {*red-step*,*dboost-step*}
  shows *card* (*Xseq i*) > *0*
  ⟨*proof*⟩

lemma *beta-le*:
  assumes $i \in$ *Step-class* {*red-step*,*dboost-step*}
  shows *beta i* $\le \mu$
  ⟨*proof*⟩

3.10 Termination proof

Each step decreases the size of X

lemma *ex-nonempty-blue-book*:
  assumes *mb*: *many-bluish X*
    shows $\exists x \in X$. *good-blue-book X* ({*x*}, *Neighbours Blue x* $\cap$ *X*)
⟨*proof*⟩

lemma *choose-blue-book-psubset*:
  assumes *many-bluish X* **and** *ST*: *choose-blue-book* (*X*,*Y*,*A*,*B*) = (*S*,*T*)
    and *finite X*
    shows $T \neq X$
⟨*proof*⟩

lemma *next-state-smaller*:
  assumes *next-state* (*X*,*Y*,*A*,*B*) = (X',Y',A',B')
    and *finite X* **and** *nont*: $\neg$ *termination-condition X Y*
  shows $X' \subset X$
⟨*proof*⟩

lemma *do-next-state*:
  assumes *odd i* $\neg$ *termination-condition* (*Xseq i*) (*Yseq i*)
  obtains *A B* A' B' **where** *next-state* (*Xseq i*, *Yseq i*, *A*, *B*)
      = (*Xseq* (*Suc i*), *Yseq* (*Suc i*), A',B')
  ⟨*proof*⟩

lemma *step-bound*:

assumes *i*: *Suc* (*2*∗*i*) ∈ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
shows *card* (*Xseq* (*Suc* (*2*∗*i*))) + *i* ≤ *card X0*
⟨*proof*⟩

lemma *Step-class-halted-nonempty*: *Step-class* {*halted*} ≠ {}
⟨*proof*⟩

definition *halted-point* ≡ *Inf* (*Step-class* {*halted*})

lemma *halted-point-halted*: *halted-point* ∈ *Step-class* {*halted*}
⟨*proof*⟩

lemma *halted-point-minimal*:
shows *i* ∉ *Step-class* {*halted*} ⟷ *i* < *halted-point*
⟨*proof*⟩

lemma *halted-point-minimal′*: *stepper-kind i* ≠ *halted* ⟷ *i* < *halted-point*
⟨*proof*⟩

lemma *halted-eq-Compl*:
Step-class {*dreg-step*,*red-step*,*bblue-step*,*dboost-step*} = − *Step-class* {*halted*}
⟨*proof*⟩

lemma *before-halted-eq*:
shows {..<*halted-point*} = *Step-class* {*dreg-step*,*red-step*,*bblue-step*,*dboost-step*}
⟨*proof*⟩

lemma *finite-components*:
shows *finite* (*Step-class* {*dreg-step*,*red-step*,*bblue-step*,*dboost-step*})
⟨*proof*⟩

lemma
shows *dreg-step-finite* [*simp*]: *finite* (*Step-class* {*dreg-step*})
and *red-step-finite* [*simp*]: *finite* (*Step-class* {*red-step*})
and *bblue-step-finite* [*simp*]: *finite* (*Step-class* {*bblue-step*})
and *dboost-step-finite*[*simp*]: *finite* (*Step-class* {*dboost-step*})
⟨*proof*⟩

lemma *halted-stepper-add-eq*: *stepper* (*halted-point* + *i*) = *stepper* (*halted-point*)
⟨*proof*⟩

lemma *halted-stepper-eq*:
assumes *i*: *i* ≥ *halted-point*
shows *stepper i* = *stepper* (*halted-point*)
⟨*proof*⟩

lemma *below-halted-point-cardX*:
assumes *i* < *halted-point*
shows *card* (*Xseq i*) > *0*

⟨*proof*⟩

end

sublocale *Book′* ⊆ *Book* **where** $\mu=\gamma$
⟨*proof*⟩

lemma (**in** *Book*) *Book′*:
assumes γ = *real l* / (*real k* + *real l*)
shows *Book′ V E p0-min Red Blue l k* γ *X0 Y0*
⟨*proof*⟩

end

4 Big Blue Steps: theorems

theory *Big-Blue-Steps* **imports** *Book*

begin

4.1 Material to delete for Isabelle 2025

lemma *gbinomial-mono*:
fixes *k*::*nat* **and** *a*::*real*
assumes *of-nat k* ≤ *a a* ≤ *b* **shows** *a gchoose k* ≤ *b gchoose k*
⟨*proof*⟩

lemma *gbinomial-is-prod*: (*a gchoose k*) = (∏ *i*<*k*. (*a* − *of-nat i*) / (*1* + *of-nat i*))
⟨*proof*⟩

lemma *smallo-multiples*:
assumes *f*: *f* ∈ *o*(*real*) **and** *k*>*0*
shows (λ*n*. *f* (*k* ∗ *n*)) ∈ *o*(*real*)
⟨*proof*⟩

4.2 Preliminaries

A bounded increasing sequence of finite sets eventually terminates

lemma *Union-incseq-finite*:
assumes *fin*: ⋀*n*. *finite* (*A n*) **and** *N*: ⋀*n*. *card* (*A n*) < *N* **and** *incseq A*
shows $\forall_F$ *k in sequentially*. ⋃ (*range A*) = *A k*
⟨*proof*⟩

Two lemmas for proving "bigness lemmas" over a closed interval

lemma *eventually-all-geI0*:
assumes $\forall_F$ *l in sequentially*. *P a l*

$\bigwedge l\ x.\ [\![P\ a\ l;\ a{\le}x;\ x{\le}b;\ l \ge L]\!] \Longrightarrow P\ x\ l$
shows $\forall_F\ l$ *in sequentially*. $\forall x.\ a \le x \wedge x \le b \longrightarrow P\ x\ l$
⟨*proof*⟩

lemma *eventually-all-geI1*:
assumes $\forall_F\ l$ *in sequentially*. $P\ b\ l$
$\bigwedge l\ x.\ [\![P\ b\ l;\ a{\le}x;\ x{\le}b;\ l \ge L]\!] \Longrightarrow P\ x\ l$
shows $\forall_F\ l$ *in sequentially*. $\forall x.\ a \le x \wedge x \le b \longrightarrow P\ x\ l$
⟨*proof*⟩

Mehta's binomial function: convex on the entire real line and coinciding with gchoose under weak conditions

definition *mfact* $\equiv \lambda a\ k.$ *if* $a <$ *real* $k - 1$ *then 0 else prod* ($\lambda i.\ a -$ *of-nat* i) $\{0..<k\}$

Mehta's special rule for convexity, my proof

lemma *convex-on-extend*:
fixes f :: *real* ⇒ *real*
assumes *cf*: *convex-on* $\{k..\}$ f **and** *mon*: *mono-on* $\{k..\}$ f
and *fk*: $\bigwedge x.\ x<k \Longrightarrow f\ x = f\ k$
shows *convex-on UNIV f*
⟨*proof*⟩

lemma *convex-mfact*:
assumes $k>0$
shows *convex-on UNIV* ($\lambda a.$ *mfact a k*)
⟨*proof*⟩

definition *mbinomial* :: *real* ⇒ *nat* ⇒ *real*
where *mbinomial* $\equiv \lambda a\ k.$ *mfact a k* / *fact k*

lemma *convex-mbinomial*: $k>0 \Longrightarrow$ *convex-on UNIV* ($\lambda x.$ *mbinomial x k*)
⟨*proof*⟩

lemma *mbinomial-eq-choose* [*simp*]: *mbinomial* (*real n*) $k = n$ *choose* k
⟨*proof*⟩

lemma *mbinomial-eq-gchoose* [*simp*]: $k \le a \Longrightarrow$ *mbinomial a k* $= a$ *gchoose* k
⟨*proof*⟩

4.3 Preliminaries: Fact D1

from appendix D, page 55

lemma *Fact-D1-73-aux*:
fixes σ::*real* **and** *m b*::*nat*
assumes σ: $0<\sigma$ **and** *bm*: *real b* < *real m*
shows $((\sigma*m)$ *gchoose* $b) *$ *inverse* (m *gchoose* b) $= \sigma$ ^$b * (\prod i<b.\ 1 - ((1-\sigma)*i) / (\sigma * ($*real m* − *real i*)))
⟨*proof*⟩

This is fact 4.2 (page 11) as well as equation (73), page 55.

lemma *Fact-D1-73*:
fixes σ::*real* **and** *m b*::*nat*
assumes σ: $0<\sigma$ $\sigma\leq 1$ **and** *b*: *real* $b \leq \sigma * m / 2$
shows $(\sigma * m)$ *gchoose* $b \in \{\sigma$ ^$b * ($*real m gchoose b*$) * exp (- ($*real b* ^ $2) / (\sigma * m))$ .. σ ^$b * (m$ *gchoose* $b)\}$
⟨*proof*⟩

Exact at zero, so cannot be done using the approximation method

lemma *exp-inequality-17*:
fixes *x*::*real*
assumes $0 \leq x$ $x \leq 1/7$
shows $1 - 4*x/3 \geq exp\ (-3*x/2)$
⟨*proof*⟩

additional part

lemma *Fact-D1-75*:
fixes σ::*real* **and** *m b*::*nat*
assumes σ: $0<\sigma$ $\sigma<1$ **and** *b*: *real* $b \leq \sigma * m / 2$ **and** b': $b \leq m/7$ **and** σ': $\sigma \geq 7/15$
shows $(\sigma * m)$ *gchoose* $b \geq exp\ (- (3 *$ *real* b ^ $2) / (4*m)) * \sigma$ ^$b * (m$ *gchoose* $b)$
⟨*proof*⟩

lemma *power2-12*: $m \geq 12 \Longrightarrow 25 * m^2 \leq 2$^$m$
⟨*proof*⟩

How b and m are obtained from l

definition *b-of* **where** *b-of* $\equiv \lambda l$::*nat*. *nat*$\lceil l$ *powr* $(1/4)\rceil$
definition *m-of* **where** *m-of* $\equiv \lambda l$::*nat*. *nat*$\lceil l$ *powr* $(2/3)\rceil$

definition *Big-Blue-4-1* $\equiv$
$\lambda\mu\ l.$ *m-of* $l \geq 12 \ \wedge\ l \geq (6/\mu)$ *powr* $(12/5) \ \wedge\ l \geq 15$
$\wedge\ 1 \leq 5/4 * exp\ (-$ *real*$(($*b-of* $l)^2) / ((\mu - 2/l) *$ *m-of* $l)) \ \wedge\ \mu > 2/l$
$\wedge\ 2/l \leq (\mu - 2/l) * ((5/4)$ *powr* $(1/$*b-of* $l) - 1)$

Establishing the size requirements for 4.1. NOTE: it doesn't become clear until SECTION 9 that all bounds involving the parameter μ must hold for a RANGE of values

lemma *Big-Blue-4-1*:
assumes $0<\mu 0$
shows $\forall^\infty l.\ \forall \mu.\ \mu \in \{\mu 0 .. \mu 1\} \longrightarrow$ *Big-Blue-4-1* $\mu\ l$
⟨*proof*⟩

context *Book*
begin

proposition *Blue-4-1*:

assumes $X \subseteq V$ **and** *manyb*: *many-bluish X*
and *big*: *Big-Blue-4-1* μ *l*
shows $\exists S\ T.$ *good-blue-book X* $(S,T) \wedge$ *card S* $\geq$ *l powr* $(1/4)$
⟨*proof*⟩

Lemma 4.3

proposition *bblue-step-limit*:
assumes *big*: *Big-Blue-4-1* μ *l*
shows *card* (*Step-class* {*bblue-step*}) $\leq$ *l powr* $(3/4)$
⟨*proof*⟩

lemma *red-steps-eq-A*:
defines *REDS* $\equiv \lambda r.$ $\{i.\ i < r \wedge$ *stepper-kind i* = *red-step*$\}$
shows *card*(*REDS n*) = *card* (*Aseq n*)
⟨*proof*⟩

proposition *red-step-eq-Aseq*: *card* (*Step-class* {*red-step*}) = *card* (*Aseq halted-point*)
⟨*proof*⟩

proposition *red-step-limit*: *card* (*Step-class* {*red-step*}) $<$ *k*
⟨*proof*⟩

proposition *bblue-dboost-step-limit*:
assumes *big*: *Big-Blue-4-1* μ *l*
shows *card* (*Step-class* {*bblue-step*}) + *card* (*Step-class* {*dboost-step*}) $<$ *l*
⟨*proof*⟩

end

end

5 Red Steps: theorems

theory *Red-Steps* **imports** *Big-Blue-Steps*

begin

Bhavik Mehta: choose-free Ramsey lower bound that's okay for very small p

lemma *Ramsey-number-lower-simple*:
assumes *n*: *of-real* $n\hat{\ }k * p$ *powr* (*real* $k\hat{\ }2$ / 4) + *of-real* $n\hat{\ }l * exp$ $(-p *$ *real* $l\hat{\ }2$ / $4) < 1$
assumes *p01*: $0<p$ $p<1$ **and** $k>1$ $l>1$
shows $\neg$ *is-Ramsey-number k l n*
⟨*proof*⟩

context *Book*

begin

5.1 Density-boost steps

5.1.1 Observation 5.5

lemma *sum-Weight-ge0*:
  assumes $X \subseteq V$ $Y \subseteq V$ *disjnt X Y*
  shows $(\sum x \in X.\ \sum x' \in X.\ Weight\ X\ Y\ x\ x') \geq 0$
⟨*proof*⟩

end

5.1.2 Lemma 5.6

definition *Big-Red-5-6-Ramsey* ≡
  $\lambda c\ l.\ nat\ \lceil real\ l\ powr\ (3/4) \rceil \geq 3$
    $\wedge\ (l\ powr\ (3/4) * (c - 1/32) \leq -1)$
    $\wedge\ (\forall k \geq l.\ k * (c * l\ powr\ (3/4) * ln\ k - k\ powr\ (7/8)\ /\ 4) \leq -1)$

establishing the size requirements for 5.6

lemma *Big-Red-5-6-Ramsey*:
  assumes $0<c$ $c<1/32$
  shows $\forall^\infty l.\ Big\text{-}Red\text{-}5\text{-}6\text{-}Ramsey\ c\ l$
⟨*proof*⟩

lemma *Red-5-6-Ramsey*:
  assumes $0<c$ $c<1/32$ **and** $l \leq k$ **and** *big*: *Big-Red-5-6-Ramsey c l*
  shows $exp\ (c * l\ powr\ (3/4) * ln\ k) \leq RN\ k\ (nat\lceil l\ powr\ (3/4) \rceil)$
⟨*proof*⟩

definition *ineq-Red-5-6* ≡ $\lambda c\ l.\ \forall k.\ l \leq k \longrightarrow exp\ (c * real\ l\ powr\ (3/4) * ln\ k)$
$\leq RN\ k\ (nat\lceil l\ powr\ (3/4) \rceil)$

definition *Big-Red-5-6* ≡
  $\lambda l.\ 6 + m\text{-}of\ l \leq (1/128) * l\ powr\ (3/4) \wedge ineq\text{-}Red\text{-}5\text{-}6\ (1/128)\ l$

establishing the size requirements for 5.6

lemma *Big-Red-5-6*: $\forall^\infty l.\ Big\text{-}Red\text{-}5\text{-}6\ l$
⟨*proof*⟩

lemma (**in** *Book*) *Red-5-6*:
  assumes *big*: *Big-Red-5-6 l*
  shows $RN\ k\ (nat\lceil l\ powr\ (3/4) \rceil) \geq k\hat{}6 * RN\ k\ (m\text{-}of\ l)$
⟨*proof*⟩

5.2 Lemma 5.4

definition *Big-Red-5-4* ≡ $\lambda l.\ Big\text{-}Red\text{-}5\text{-}6\ l \wedge (\forall k \geq l.\ real\ k + 2 * real\ k\hat{}6 \leq real$
$k\hat{}7)$

establishing the size requirements for 5.4

lemma *Big-Red-5-4*: $\forall^\infty l.$ *Big-Red-5-4 l*
⟨*proof*⟩

context *Book*
begin

lemma *Red-5-4*:
assumes *i*: $i \in$ *Step-class* {*red-step*,*dboost-step*}
and *big*: *Big-Red-5-4 l*
defines $X \equiv$ *Xseq i* **and** $Y \equiv$ *Yseq i*
shows *weight X Y* (*cvx i*) $\geq -$ *card X* / (*real k*)^5
⟨*proof*⟩

lemma *Red-5-7a*: *eps k* / *k* $\leq$ *alpha* (*hgt p*)
⟨*proof*⟩

lemma *Red-5-7b*:
assumes $p \geq$ *qfun 0* **shows** *alpha* (*hgt p*) $\leq$ *eps k* $*$ ($p -$ *qfun 0* + *1*/*k*)
⟨*proof*⟩

lemma *Red-5-7c*:
assumes $p \leq$ *qfun 1* **shows** *alpha* (*hgt p*) = *eps k* / *k*
⟨*proof*⟩

lemma *Red-5-8*:
assumes *i*: $i \in$ *Step-class* {*dreg-step*} **and** *x*: $x \in$ *Xseq* (*Suc i*)
shows *card* (*Neighbours Red x* $\cap$ *Yseq* (*Suc i*))
$\geq$ (*1* $-$ (*eps k*) *powr* (*1*/*2*)) $*$ *pee i* $*$ (*card* (*Yseq* (*Suc i*)))
⟨*proof*⟩

corollary *Y-Neighbours-nonempty-Suc*:
assumes *i*: $i \in$ *Step-class* {*dreg-step*} **and** *x*: $x \in$ *Xseq* (*Suc i*) **and** $k \geq 2$
shows *Neighbours Red x* $\cap$ *Yseq* (*Suc i*) $\neq$ {}
⟨*proof*⟩

corollary *Y-Neighbours-nonempty*:
assumes *i*: $i \in$ *Step-class* {*red-step*,*dboost-step*} **and** *x*: $x \in$ *Xseq i* **and** $k \geq 2$
shows *card* (*Neighbours Red x* $\cap$ *Yseq i*) > *0*
⟨*proof*⟩

end

5.3 Lemma 5.1

definition *Big-Red-5-1* $\equiv \lambda\mu\ l.$ (*1*$-\mu$) $*$ *real l* > *1* $\wedge$ *l powr* (*5*/*2*) $\geq$ *3* / (*1*$-\mu$) $\wedge$ *l powr* (*1*/*4*) $\geq$ *4*
$\wedge$ *Big-Red-5-4 l* $\wedge$ *Big-Red-5-6 l*

establishing the size requirements for 5.1

lemma *Big-Red-5-1*:
  assumes $\mu 1 < 1$
  shows $\forall^\infty l.\ \forall \mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-Red-5-1* μ *l*
⟨*proof*⟩

context *Book*
begin

lemma *card-cvx-Neighbours*:
  assumes *i*: $i \in$ *Step-class* {*red-step*,*dboost-step*}
  defines $x \equiv$ *cvx i*
  defines $X \equiv$ *Xseq i*
  defines *NBX* $\equiv$ *Neighbours Blue* $x \cap X$
  defines *NRX* $\equiv$ *Neighbours Red* $x \cap X$
  shows *card NBX* $\leq \mu *$ *card X card NRX* $\geq (1-\mu) *$ *card X* $-$ *1*
⟨*proof*⟩

proposition *Red-5-1*:
  assumes *i*: $i \in$ *Step-class* {*red-step*,*dboost-step*} **and** *Big*: *Big-Red-5-1* μ *l*
  defines $p \equiv$ *pee i*
  defines $x \equiv$ *cvx i*
  defines $X \equiv$ *Xseq i* **and** $Y \equiv$ *Yseq i*
  defines *NBX* $\equiv$ *Neighbours Blue* $x \cap X$
  defines *NRX* $\equiv$ *Neighbours Red* $x \cap X$
  defines *NRY* $\equiv$ *Neighbours Red* $x \cap Y$
  defines $\beta \equiv$ *card NBX* / *card X*
  shows *red-density NRX NRY* $\geq p -$ *alpha* (*hgt p*)
    $\vee$ *red-density NBX NRY* $\geq p + (1 -$ *eps k*$) * ((1-\beta) / \beta) *$ *alpha* (*hgt p*)
$\wedge\ \beta > 0$
⟨*proof*⟩

This and the previous result are proved under the assumption of a sufficiently large *l*

corollary *Red-5-2*:
  assumes *i*: $i \in$ *Step-class* {*dboost-step*}
    and *Big*: *Big-Red-5-1* μ *l*
  shows *pee* (*Suc i*) $-$ *pee i* $\geq (1 -$ *eps k*$) * ((1 -$ *beta i*$) /$ *beta i*$) *$ *alpha* (*hgt* (*pee i*)) $\wedge$
      beta i > 0
⟨*proof*⟩

end

5.4 Lemma 5.3

This is a weaker consequence of the previous results

definition
  Big-Red-5-3 $\equiv$
    $\lambda\mu$ *l*. *Big-Red-5-1* μ *l*

$\wedge$ ($\forall k \geq l.$ $k>1 \wedge 1$ / (real $k)^2 \leq \mu \wedge 1$ / (real $k)^2 \leq 1$ / (k / eps k / (1 − eps k) + 1))

establishing the size requirements for 5.3. The one involving μ, namely 1 / (real $k)^2 \leq \mu$, will be useful later with "big beta".

lemma *Big-Red-5-3*:
assumes $0<\mu 0$ $\mu 1<1$
shows $\forall^{\infty} l.$ $\forall \mu.$ $\mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-Red-5-3* μ l
⟨*proof*⟩

context *Book*
begin

corollary *Red-5-3*:
assumes *i*: $i \in$ *Step-class* {*dboost-step*}
and *big*: *Big-Red-5-3* μ l
shows *pee* (*Suc i*) $\geq$ *pee i* $\wedge$ *beta i* ≥ 1 / (*real k*)^2
⟨*proof*⟩

corollary *beta-gt0*:
assumes $i \in$ *Step-class* {*dboost-step*}
and *Big-Red-5-3* μ l
shows *beta i* > 0
⟨*proof*⟩

end

end

6 Bounding the Size of Y

theory *Bounding-Y* **imports** *Red-Steps*

begin

yet another telescope variant, with weaker promises but a different conclusion; as written it holds even if $n = (0::'a)$

lemma *prod-lessThan-telescope-mult*:
fixes *f*::*nat* $\Rightarrow$ *'a*::*field*
assumes $\bigwedge i.$ $i<n \Longrightarrow f\ i \neq 0$
shows $(\prod i<n.\ f\ (Suc\ i)\ /\ f\ i) * f\ 0 = f\ n$
⟨*proof*⟩

6.1 The following results together are Lemma 6.4

Compared with the paper, all the indices are greater by one!!

context *Book*
begin

lemma *Y-6-4-Red*:
  assumes $i \in$ *Step-class* {*red-step*}
  shows *pee* (*Suc i*) $\geq$ *pee i* $-$ *alpha* (*hgt* (*pee i*))
  ⟨*proof*⟩

lemma *Y-6-4-DegreeReg*:
  assumes $i \in$ *Step-class* {*dreg-step*}
  shows *pee* (*Suc i*) $\geq$ *pee i*
  ⟨*proof*⟩

lemma *Y-6-4-Bblue*:
  assumes *i*: $i \in$ *Step-class* {*bblue-step*}
  shows *pee* (*Suc i*) $\geq$ *pee* ($i-1$) $-$ (*eps k powr* ($-1/2$)) $*$ *alpha* (*hgt* (*pee* ($i-1$)))
⟨*proof*⟩

The basic form is actually *Red-5-3*. This variant covers a gap of two, thanks to degree regularisation

corollary *Y-6-4-dbooSt*:
  assumes *i*: $i \in$ *Step-class* {*dboost-step*} **and** *big*: *Big-Red-5-3* μ *l*
  shows *pee* (*Suc i*) $\geq$ *pee* ($i-1$)
⟨*proof*⟩

6.2 Towards Lemmas 6.3

definition *Z-class* $\equiv$ {$i \in$ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}.
                            pee (*Suc i*) $<$ *pee* ($i-1$) $\wedge$ *pee* ($i-1$) $\leq$ *p0*}

lemma *finite-Z-class*: *finite* (*Z-class*)
  ⟨*proof*⟩

lemma *Y-6-3*:
  assumes *big53*: *Big-Red-5-3* μ *l* **and** *big41*: *Big-Blue-4-1* μ *l*
  shows ($\sum i \in$ *Z-class*. *pee* ($i-1$) $-$ *pee* (*Suc i*)) $\leq 2 *$ *eps k*
⟨*proof*⟩

6.3 Lemma 6.5

lemma *Y-6-5-Red*:
  assumes *i*: $i \in$ *Step-class* {*red-step*} **and** $k \geq 16$
  defines $h \equiv \lambda i.$ *hgt* (*pee i*)
  shows *h* (*Suc i*) $\geq$ *h i* $-$ *2*
⟨*proof*⟩

lemma *Y-6-5-DegreeReg*:
  assumes $i \in$ *Step-class* {*dreg-step*}
  shows *hgt* (*pee* (*Suc i*)) $\geq$ *hgt* (*pee i*)
  ⟨*proof*⟩

corollary *Y-6-5-dbooSt*:

assumes $i \in$ *Step-class* {*dboost-step*} **and** *Big-Red-5-3* μ *l*
shows *hgt* (*pee* (*Suc i*)) $\geq$ *hgt* (*pee i*)
⟨*proof*⟩

this remark near the top of page 19 only holds in the limit

lemma $\forall^\infty k$. (*1* + *eps k*) *powr* (− *real* (*nat* ⌊*2* ∗ *eps k powr* (−*1*/*2*)⌋)) ≤ *1* − *eps k powr* (*1*/*2*)
⟨*proof*⟩

end

definition *Big-Y-6-5-Bblue* ≡ λ*l*. ∀ *k*≥*l*. (*1* + *eps k*) *powr* (− *real* (*nat* ⌊*2*∗(*eps k powr* (−*1*/*2*))⌋)) ≤ *1* − *eps k powr* (*1*/*2*)

establishing the size requirements for Y 6.5

lemma *Big-Y-6-5-Bblue*:
shows $\forall^\infty l$. *Big-Y-6-5-Bblue l*
⟨*proof*⟩

lemma (**in** *Book*) *Y-6-5-Bblue*:
fixes κ::*real*
defines κ ≡ *eps k powr* (−*1*/*2*)
assumes *i*: *i* ∈ *Step-class* {*bblue-step*} **and** *big*: *Big-Y-6-5-Bblue l*
defines *h* ≡ *hgt* (*pee* (*i*−*1*))
shows *hgt* (*pee* (*Suc i*)) ≥ *h* − *2*∗κ
⟨*proof*⟩

6.4 Lemma 6.2

definition *Big-Y-6-2* ≡ λμ *l*. *Big-Y-6-5-Bblue l* ∧ *Big-Red-5-3* μ *l* ∧ *Big-Blue-4-1* μ *l*
∧ (∀ *k*≥*l*. ((*1* + *eps k*)^*2*) ∗ *eps k powr* (*1*/*2*) ≤ *1*
∧ (*1* + *eps k*) *powr* (*2* ∗ *eps k powr* (−*1*/*2*)) ≤ *2* ∧ *k* ≥ *16*)

establishing the size requirements for 6.2

lemma *Big-Y-6-2*:
assumes *0*<μ*0* μ*1*<*1*
shows $\forall^\infty l$. ∀ μ. μ ∈ {μ*0*..μ*1*} ⟶ *Big-Y-6-2* μ *l*
⟨*proof*⟩

context *Book*
begin

Following Bhavik in excluding the even steps (degree regularisation). Assuming it hasn't halted, the conclusion also holds for the even cases anyway.

proposition *Y-6-2*:
defines *RBS* ≡ *Step-class* {*red-step*,*bblue-step*,*dboost-step*}
assumes *j*: *j* ∈ *RBS* **and** *big*: *Big-Y-6-2* μ *l*
shows *pee* (*Suc j*) ≥ *p0* − *3* ∗ *eps k*

⟨proof⟩

corollary *Y-6-2-halted*:
assumes *big*: *Big-Y-6-2 μ l*
shows *pee halted-point* ≥ *p0* − *3* ∗ *eps k*
⟨proof⟩

end

6.5 Lemma 6.1

context *P0-min*
begin

definition *ok-fun-61* ≡ λ*k*. (*2* ∗ *real k* / *ln 2*) ∗ *ln* (*1* − *2* ∗ *eps k powr* (*1/2*) / *p0-min*)

Not actually used, but justifies the definition above

lemma *ok-fun-61-works*:
assumes *k*>*0* *p0-min* > *2* ∗ *eps k powr* (*1/2*)
shows *2 powr* (*ok-fun-61 k*) = (*1* − *2* ∗ (*eps k*) *powr*(*1/2*) / *p0-min*) ^ (*2*∗*k*)
⟨proof⟩

lemma *ok-fun-61*: *ok-fun-61* ∈ *o*(*real*)
⟨proof⟩

definition
Big-Y-6-1 ≡
λμ *l*. *Big-Y-6-2 μ l* ∧ (∀ *k*≥*l*. *eps k powr* (*1/2*) ≤ *1/3* ∧ *p0-min* > *2* ∗ *eps k powr* (*1/2*))

establishing the size requirements for 6.1

lemma *Big-Y-6-1*:
assumes *0*<μ*0* μ*1*<*1*
shows ∀$^\infty$*l*. ∀ μ. μ ∈ {μ*0*..μ*1*} ⟶ *Big-Y-6-1 μ l*
⟨proof⟩

end

lemma (**in** *Book*) *Y-6-1*:
assumes *big*: *Big-Y-6-1 μ l*
defines *st* ≡ *Step-class* {*red-step*,*dboost-step*}
shows *card* (*Yseq halted-point*) / *card Y0* ≥ *2 powr* (*ok-fun-61 k*) ∗ *p0* ^ *card st*
⟨proof⟩

end

7 Bounding the Size of X

theory *Bounding-X* **imports** *Bounding-Y*

begin

7.1 Preliminaries

lemma *sum-odds-even*:
fixes f :: *nat* ⇒ *'a* :: *ab-group-add*
assumes *even m*
shows $(\sum i \in \{i.\ i<m \wedge odd\ i\}.\ f\ (Suc\ i) - f\ (i - Suc\ 0)) = f\ m - f\ 0$
⟨*proof*⟩

lemma *sum-odds-odd*:
fixes f :: *nat* ⇒ *'a* :: *ab-group-add*
assumes *odd m*
shows $(\sum i \in \{i.\ i<m \wedge odd\ i\}.\ f\ (Suc\ i) - f\ (i - Suc\ 0)) = f\ (m-1) - f\ 0$
⟨*proof*⟩

context *Book*
begin

the set of moderate density-boost steps (page 20)

definition *dboost-star* **where**
dboost-star $\equiv \{i \in Step\text{-}class\ \{dboost\text{-}step\}.\ real\ (hgt\ (pee\ (Suc\ i))) - hgt\ (pee\ i) \le eps\ k\ powr\ (-1/4)\}$

definition *bigbeta* **where**
bigbeta $\equiv$ *let* $S = dboost\text{-}star$ *in if* $S = \{\}$ *then* μ *else* $(card\ S) * inverse\ (\sum i{\in}S.\ inverse\ (beta\ i))$

lemma *dboost-star-subset*: *dboost-star* $\subseteq$ *Step-class* {*dboost-step*}
⟨*proof*⟩

lemma *finite-dboost-star*: *finite* (*dboost-star*)
⟨*proof*⟩

lemma *bigbeta-ge0*: *bigbeta* ≥ 0
⟨*proof*⟩

lemma *bigbeta-ge-square*:
assumes *big*: *Big-Red-5-3* μ *l*
shows *bigbeta* $\ge 1\ /\ (real\ k)\hat{\ }2$
⟨*proof*⟩

lemma *bigbeta-gt0*:
assumes *big*: *Big-Red-5-3* μ *l*

shows *bigbeta* > 0
⟨*proof*⟩

lemma *bigbeta-less1*:
assumes *big*: *Big-Red-5-3* μ *l*
shows *bigbeta* < 1
⟨*proof*⟩

lemma *bigbeta-le*:
assumes *big*: *Big-Red-5-3* μ *l*
shows *bigbeta* $\leq \mu$
⟨*proof*⟩

end

7.2 Lemma 7.2

definition *Big-X-7-2* $\equiv \lambda\mu\ l.\ nat\ \lceil real\ l\ powr\ (3/4)\rceil \geq 3 \wedge l > 1\ /\ (1-\mu)$

establishing the size requirements for 7.11

lemma *Big-X-7-2*:
assumes $0<\mu 0\ \ \mu 1<1$
shows $\forall^{\infty} l.\ \forall \mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-X-7-2* μ *l*
⟨*proof*⟩

definition *ok-fun-72* $\equiv \lambda\mu\ k.\ (real\ k\ /\ ln\ 2) * ln\ (1\ -\ 1\ /\ (k * (1-\mu)))$

lemma *ok-fun-72*:
assumes $\mu<1$
shows *ok-fun-72* $\mu \in o(real)$
⟨*proof*⟩

lemma *ok-fun-72-uniform*:
assumes $0<\mu 0\ \ \mu 1<1$
assumes $e>0$
shows $\forall^{\infty} k.\ \forall \mu.\ \mu 0 \leq \mu \wedge \mu \leq \mu 1 \longrightarrow |$*ok-fun-72* $\mu\ k|\ /\ k \leq e$
⟨*proof*⟩

lemma (**in** *Book*) *X-7-2*:
defines $\mathcal{R} \equiv$ *Step-class* {*red-step*}
assumes *big*: *Big-X-7-2* μ *l*
shows $(\prod i{\in}\mathcal{R}.\ card\ (Xseq(Suc\ i))\ /\ card\ (Xseq\ i)) \geq 2\ powr\ (ok\text{-}fun\text{-}72\ \mu\ k) *$
$(1-\mu)$ ^ $card\ \mathcal{R}$
⟨*proof*⟩

7.3 Lemma 7.3

context *Book*
begin

definition *Bdelta* ≡ λ μ *i*. *Bseq* (*Suc i*) \ *Bseq i*

lemma *card-Bdelta*: *card* (*Bdelta* μ *i*) = *card* (*Bseq* (*Suc i*)) − *card* (*Bseq i*)
⟨*proof*⟩

lemma *card-Bseq-mono*: *card* (*Bseq* (*Suc i*)) ≥ *card* (*Bseq i*)
⟨*proof*⟩

lemma *card-Bseq-sum*: *card* (*Bseq i*) = (∑ *j*<*i*. *card* (*Bdelta* μ *j*))
⟨*proof*⟩

definition *get-blue-book* ≡ λ*i*. *let* (*X*,*Y*,*A*,*B*) = *stepper i in choose-blue-book* (*X*,*Y*,*A*,*B*)

Tracking changes to X and B. The sets are necessarily finite

lemma *Bdelta-bblue-step*:
assumes *i* ∈ *Step-class* {*bblue-step*}
shows ∃ *S* ⊆ *Xseq i*. *Bdelta* μ *i* = *S*
∧ *card* (*Xseq* (*Suc i*)) ≥ (μ ^ *card S*) ∗ *card* (*Xseq i*) / *2*
⟨*proof*⟩

lemma *Bdelta-dboost-step*:
assumes *i* ∈ *Step-class* {*dboost-step*}
shows ∃ *x* ∈ *Xseq i*. *Bdelta* μ *i* = {*x*}
⟨*proof*⟩

lemma *card-Bdelta-dboost-step*:
assumes *i* ∈ *Step-class* {*dboost-step*}
shows *card* (*Bdelta* μ *i*) = *1*
⟨*proof*⟩

lemma *Bdelta-trivial-step*:
assumes *i*: *i* ∈ *Step-class* {*red-step*,*dreg-step*,*halted*}
shows *Bdelta* μ *i* = {}
⟨*proof*⟩

end

definition *ok-fun-73* ≡ λ*k*. − (*real k powr* (*3*/*4*))

lemma *ok-fun-73*: *ok-fun-73* ∈ *o*(*real*)
⟨*proof*⟩

lemma (**in** *Book*) *X-7-3*:
assumes *big*: *Big-Blue-4-1* μ *l*
defines ℬ ≡ *Step-class* {*bblue-step*}
defines 𝒮 ≡ *Step-class* {*dboost-step*}
shows (∏ *i* ∈ ℬ. *card* (*Xseq*(*Suc i*)) / *card* (*Xseq i*)) ≥ *2 powr* (*ok-fun-73 k*) ∗ μ ^ (*l* − *card* 𝒮)

⟨*proof*⟩

7.4 Lemma 7.5

Small $o(k)$ bounds on summations for this section

This is the explicit upper bound for heights given just below (5) on page 9

definition *ok-fun-26* ≡ *λk. 2 ∗ ln k / eps k*

definition *ok-fun-28* ≡ *λk. −2 ∗ real k powr (7/8)*

lemma *ok-fun-26*: *ok-fun-26* ∈ *o*(*real*) **and** *ok-fun-28*: *ok-fun-28* ∈ *o*(*real*)
⟨*proof*⟩

definition
Big-X-7-5 ≡
λμ l. Big-Blue-4-1 μ l ∧ *Big-Red-5-3 μ l* ∧ *Big-Y-6-5-Bblue l*
∧ (∀ *k*≥*l. Big-height-upper-bound k* ∧ *k*≥*16* ∧ (*ok-fun-26 k* − *ok-fun-28 k* ≤ *k*))

establishing the size requirements for 7.5

lemma *Big-X-7-5*:
assumes *0<μ0 μ1<1*
shows ∀$^\infty$*l.* ∀*μ. μ* ∈ {*μ0..μ1*} ⟶ *Big-X-7-5 μ l*
⟨*proof*⟩

context *Book*
begin

lemma *X-26-and-28*:
assumes *big*: *Big-X-7-5 μ l*
defines $\mathcal{D}$ ≡ *Step-class* {*dreg-step*}
defines $\mathcal{B}$ ≡ *Step-class* {*bblue-step*}
defines $\mathcal{H}$ ≡ *Step-class* {*halted*}
defines *h* ≡ *λi. real* (*hgt* (*pee i*))
obtains (∑ *i*∈{..<*halted-point*} \ $\mathcal{D}$. *h* (*Suc i*) − *h* (*i*−*1*)) ≤ *ok-fun-26 k*
ok-fun-28 k ≤ (∑ *i* ∈ $\mathcal{B}$. *h*(*Suc i*) − *h*(*i*−*1*))
⟨*proof*⟩

proposition *X-7-5*:
assumes *μ*: *0<μ μ<1*
defines $\mathcal{S}$ ≡ *Step-class* {*dboost-step*} **and** $\mathcal{SS}$ ≡ *dboost-star*
assumes *big*: *Big-X-7-5 μ l*
shows *card* ($\mathcal{S}$\\$\mathcal{SS}$) ≤ *3 ∗ eps k powr* (*1/4*) *∗ k*
⟨*proof*⟩

end

7.5 Lemma 7.4

definition
Big-X-7-4 $\equiv \lambda\mu\ l.$ *Big-X-7-5* $\mu\ l \wedge$ *Big-Red-5-3* $\mu\ l$

establishing the size requirements for 7.4

lemma *Big-X-7-4*:
assumes $0<\mu 0\ \mu 1<1$
shows $\forall^\infty l.\ \forall \mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-X-7-4* $\mu\ l$
⟨*proof*⟩

definition *ok-fun-74* $\equiv \lambda k.\ -6 * eps\ k\ powr\ (1/4) * k * ln\ k\ /\ ln\ 2$

lemma *ok-fun-74*: *ok-fun-74* $\in o(real)$
⟨*proof*⟩

context *Book*
begin

lemma *X-7-4*:
assumes *big*: *Big-X-7-4* $\mu\ l$
defines $\mathcal{S} \equiv$ *Step-class* {*dboost-step*}
shows $(\prod i{\in}\mathcal{S}.\ card\ (Xseq\ (Suc\ i))\ /\ card\ (Xseq\ i)) \geq 2\ powr\ ok\text{-}fun\text{-}74\ k *$ *bigbeta* ^ *card* $\mathcal{S}$
⟨*proof*⟩

7.6 Observation 7.7

lemma *X-7-7*:
assumes *i*: $i \in$ *Step-class* {*dreg-step*}
defines $q \equiv eps\ k\ powr\ (-1/2) * alpha\ (hgt\ (pee\ i))$
shows $pee\ (Suc\ i) - pee\ i \geq card\ (Xseq\ i \setminus Xseq\ (Suc\ i))\ /\ card\ (Xseq\ (Suc\ i)) * q \wedge card\ (Xseq\ (Suc\ i)) > 0$
⟨*proof*⟩

end

7.7 Lemma 7.8

definition *Big-X-7-8* $\equiv \lambda k.\ k{\geq}2 \wedge eps\ k\ powr\ (1/2)\ /\ k \geq 2\ /\ k$^2

lemma *Big-X-7-8*: $\forall^\infty k.$ *Big-X-7-8* k
⟨*proof*⟩

lemma (**in** *Book*) *X-7-8*:
assumes *big*: *Big-X-7-8* k
and *i*: $i \in$ *Step-class* {*dreg-step*}
shows $card\ (Xseq\ (Suc\ i)) \geq card\ (Xseq\ i)\ /\ k$^2
⟨*proof*⟩

7.8 Lemma 7.9

definition *Big-X-7-9* ≡ $\lambda k.$ ((*1* + *eps k*) *powr* (*eps k powr* (−*1/4*) + *1*) − *1*) / *eps k* ≤ *2* ∗ *eps k powr* (−*1/4*)
∧ *k*≥*2* ∧ *eps k powr* (*1/2*) / *k* ≥ *2* / *k*ˆ*2*

lemma *Big-X-7-9*: ∀∞*k*. *Big-X-7-9 k*
⟨*proof*⟩

lemma *one-plus-powr-le*:
fixes *p*::*real*
assumes *0*≤*p p*≤*1 x*≥*0*
shows (*1*+*x*) *powr p* − *1* ≤ *x*∗*p*
⟨*proof*⟩

lemma (**in** *Book*) *X-7-9*:
assumes *i*: *i* ∈ *Step-class* {*dreg-step*} **and** *big*: *Big-X-7-9 k*
defines *hp* ≡ λ*i*. *hgt* (*pee i*)
assumes *pee i* ≥ *p0* **and** *hgt*: *hp* (*Suc i*) ≤ *hp i* + *eps k powr* (−*1/4*)
shows *card* (*Xseq* (*Suc i*)) ≥ (*1* − *2* ∗ *eps k powr* (*1/4*)) ∗ *card* (*Xseq i*)
⟨*proof*⟩

7.9 Lemma 7.10

definition *Big-X-7-10* ≡ λμ *l*. *Big-X-7-5* μ *l* ∧ *Big-Red-5-3* μ *l*

establishing the size requirements for 7.10

lemma *Big-X-7-10*:
assumes *0*<μ*0* μ*1*<*1*
shows ∀∞*l*. ∀μ. μ ∈ {μ*0*..μ*1*} ⟶ *Big-X-7-10* μ *l*
⟨*proof*⟩

lemma (**in** *Book*) *X-7-10*:
defines $\mathcal{R}$ ≡ *Step-class* {*red-step*}
defines $\mathcal{S}$ ≡ *Step-class* {*dboost-step*}
defines *h* ≡ λ*i*. *real* (*hgt* (*pee i*))
defines *C* ≡ {*i*. *h i* ≥ *h* (*i*−*1*) + *eps k powr* (−*1/4*)}
assumes *big*: *Big-X-7-10* μ *l*
shows *card* (($\mathcal{R}$∪$\mathcal{S}$) ∩ *C*) ≤ *3* ∗ *eps k powr* (*1/4*) ∗ *k*
⟨*proof*⟩

7.10 Lemma 7.11

definition *Big-X-7-11-inequalities* ≡ λ*k*.
eps k ∗ *eps k powr* (−*1/4*) ≤ (*1* + *eps k*) ˆ (*2* ∗ *nat* ⌊*eps k powr* (−*1/4*)⌋) − *1*
∧ *k* ≥ *2* ∗ *eps k powr* (−*1/2*) ∗ *k powr* (*3/4*)
∧ ((*1* + *eps k*) ∗ (*1* + *eps k*) *powr* (*2* ∗ *eps k powr* (−*1/4*))) ≤ *2*

$\wedge$ (*1* + *eps k*) ^ (*nat* $\lfloor$*2* $*$ *eps k powr* (−*1*/*4*)$\rfloor$ + *nat* $\lfloor$*2* $*$ *eps k powr* (−*1*/*2*)$\rfloor$ − *1*) $\le$ *2*

definition *Big-X-7-11* $\equiv$
$\lambda\mu$ *l*. *Big-X-7-5* μ *l* $\wedge$ *Big-Red-5-3* μ *l* $\wedge$ *Big-Y-6-5-Bblue l*
$\wedge$ ($\forall$ *k*. *l*$\le$*k* $\longrightarrow$ *Big-X-7-11-inequalities k*)

establishing the size requirements for 7.11

lemma *Big-X-7-11*:
assumes *0*<μ*0* μ*1*<*1*
shows $\forall^\infty$*l*. $\forall$ μ. μ $\in$ {μ*0*..μ*1*} $\longrightarrow$ *Big-X-7-11* μ *l*
⟨*proof*⟩

lemma (**in** *Book*) *X-7-11*:
defines $\mathcal{R}$ $\equiv$ *Step-class* {*red-step*}
defines $\mathcal{S}$ $\equiv$ *Step-class* {*dboost-step*}
defines *C* $\equiv$ {*i*. *pee i* $\ge$ *pee* (*i*−*1*) + *eps k powr* (−*1*/*4*) $*$ *alpha 1* $\wedge$ *pee* (*i*−*1*) $\le$ *p0*}
assumes *big*: *Big-X-7-11* μ *l*
shows *card* (($\mathcal{R}$$\cup$$\mathcal{S}$) $\cap$ *C*) $\le$ *4* $*$ *eps k powr* (*1*/*4*) $*$ *k*
⟨*proof*⟩

7.11 Lemma 7.12

definition *Big-X-7-12* $\equiv$
$\lambda\mu$ *l*. *Big-X-7-11* μ *l* $\wedge$ *Big-X-7-10* μ *l* $\wedge$ ($\forall$ *k*. *l*$\le$*k* $\longrightarrow$ *Big-X-7-9 k*)

establishing the size requirements for 7.12

lemma *Big-X-7-12*:
assumes *0*<μ*0* μ*1*<*1*
shows $\forall^\infty$*l*. $\forall$ μ. μ $\in$ {μ*0*..μ*1*} $\longrightarrow$ *Big-X-7-12* μ *l*
⟨*proof*⟩

lemma (**in** *Book*) *X-7-12*:
defines $\mathcal{R}$ $\equiv$ *Step-class* {*red-step*}
defines $\mathcal{S}$ $\equiv$ *Step-class* {*dboost-step*}
defines *C* $\equiv$ {*i*. *card* (*Xseq i*) < (*1* − *2* $*$ *eps k powr* (*1*/*4*)) $*$ *card* (*Xseq* (*i*−*1*))}
assumes *big*: *Big-X-7-12* μ *l*
shows *card* (($\mathcal{R}$$\cup$$\mathcal{S}$) $\cap$ *C*) $\le$ *7* $*$ *eps k powr* (*1*/*4*) $*$ *k*
⟨*proof*⟩

7.12 Lemma 7.6

definition *Big-X-7-6* $\equiv$
$\lambda\mu$ *l*. *Big-Blue-4-1* μ *l* $\wedge$ *Big-X-7-12* μ *l* $\wedge$ ($\forall$ *k*. *k*$\ge$*l* $\longrightarrow$ *Big-X-7-8 k* $\wedge$ *1* − *2* $*$ *eps k powr* (*1*/*4*) > *0*)

lemma *Big-X-7-6*:
assumes *0*<μ*0* μ*1*<*1*

shows $\forall^\infty l.\ \forall\mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-X-7-6* μ l
⟨*proof*⟩

definition *ok-fun-76* ≡
$\lambda k.$ ((*1* + *2* ∗ *real k*) ∗ *ln* (*1* − *2* ∗ *eps k powr* (*1/4*))
− (*k powr* (*3/4*) + *7* ∗ *eps k powr* (*1/4*) ∗ *k* + *1*) ∗ (*2* ∗ *ln k*)) / *ln 2*

lemma *ok-fun-76*: *ok-fun-76* ∈ *o*(*real*)
⟨*proof*⟩

lemma (**in** *Book*) *X-7-6*:
assumes *big*: *Big-X-7-6* μ l
defines $\mathcal{D}$ ≡ *Step-class* {*dreg-step*}
shows ($\prod$ *i*∈$\mathcal{D}$. *card*(*Xseq*(*Suc i*)) / *card* (*Xseq i*)) ≥ *2 powr ok-fun-76 k*
⟨*proof*⟩

7.13 Lemma 7.1

definition *Big-X-7-1* ≡
$\lambda\mu$ *l*. *Big-Blue-4-1* μ *l* ∧ *Big-X-7-2* μ *l* ∧ *Big-X-7-4* μ *l* ∧ *Big-X-7-6* μ *l*

establishing the size requirements for 7.11

lemma *Big-X-7-1*:
assumes *0*<μ*0* μ*1*<*1*
shows $\forall^\infty l.\ \forall\mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-X-7-1* μ l
⟨*proof*⟩

definition *ok-fun-71* ≡ $\lambda\mu$ *k*. *ok-fun-72* μ *k* + *ok-fun-73 k* + *ok-fun-74 k* + *ok-fun-76 k*

lemma *ok-fun-71*:
assumes *0*<μ μ<*1*
shows *ok-fun-71* μ ∈ *o*(*real*)
⟨*proof*⟩

lemma (**in** *Book*) *X-7-1*:
assumes *big*: *Big-X-7-1* μ l
defines $\mathcal{D}$ ≡ *Step-class* {*dreg-step*}
defines $\mathcal{R}$ ≡ *Step-class* {*red-step*} **and** $\mathcal{S}$ ≡ *Step-class* {*dboost-step*}
shows *card* (*Xseq halted-point*) ≥ *2 powr ok-fun-71* μ *k* ∗ μ^*l* ∗ (*1*−μ) ^ *card* $\mathcal{R}$ ∗ (*bigbeta* / μ) ^ *card* $\mathcal{S}$ ∗ *card X0*
⟨*proof*⟩

end

8 The Zigzag Lemma

theory *Zigzag* **imports** *Bounding-X*

begin

8.1 Lemma 8.1 (the actual Zigzag Lemma)

definition *Big-ZZ-8-2* ≡ $\lambda k.$ (*1* + *eps k powr* (*1/2*)) ≥ (*1* + *eps k*) *powr* (*eps k powr* (−*1/4*))

An inequality that pops up in the proof of (39)

definition *Big39* ≡ $\lambda k.$ *1/2* ≤ (*1* + *eps k*) *powr* (−*2* ∗ *eps k powr* (−*1/2*))

Two inequalities that pops up in the proof of (42)

definition *Big42a* ≡ $\lambda k.$ (*1* + *eps k*)^2 / (*1* − *eps k powr* (*1/2*)) ≤ *1* + *2* ∗ *k powr* (−*1/16*)

definition *Big42b* ≡ $\lambda k.$ *2* ∗ *k powr* (−*1/16*) ∗ *k*
+ (*1* + *2* ∗ *ln k* / *eps k* + *2* ∗ *k powr* (*7/8*)) / (*1* − *eps k powr* (*1/2*))
≤ *real k powr* (*19/20*)

definition *Big-ZZ-8-1* ≡
$\lambda\mu$ *l*. *Big-Blue-4-1* μ *l* ∧ *Big-Red-5-1* μ *l* ∧ *Big-Red-5-3* μ *l* ∧ *Big-Y-6-5-Bblue l*
∧ (∀ *k*. *k*≥*l* ⟶ *Big-height-upper-bound k* ∧ *Big-ZZ-8-2 k* ∧ *k*≥*16* ∧ *Big39 k*
∧ *Big42a k* ∧ *Big42b k*)

(*16*::′*a*) ≤ *k* is for *Y-6-5-Red*

lemma *Big-ZZ-8-1*:
assumes *0*<μ*0* μ*1*<*1*
shows ∀$^\infty$*l*. ∀μ. μ ∈ {μ*0*..μ*1*} ⟶ *Big-ZZ-8-1* μ *l*
⟨*proof*⟩

lemma (**in** *Book*) *ZZ-8-1*:
assumes *big*: *Big-ZZ-8-1* μ *l*
defines $\mathcal{R}$ ≡ *Step-class* {*red-step*}
defines *sum-SS* ≡ (∑ *i*∈*dboost-star*. (*1* − *beta i*) / *beta i*)
shows *sum-SS* ≤ *real* (*card* $\mathcal{R}$) + *k powr* (*19/20*)
⟨*proof*⟩

8.2 Lemma 8.5

An inequality that pops up in the proof of (39)

definition *inequality85* ≡ $\lambda k.$ *3* ∗ *eps k powr* (*1/4*) ∗ *k* ≤ *k powr* (*19/20*)

definition *Big-ZZ-8-5* ≡
$\lambda\mu$ *l*. *Big-X-7-5* μ *l* ∧ *Big-ZZ-8-1* μ *l* ∧ *Big-Red-5-3* μ *l*
∧ (∀ *k*≥*l*. *inequality85 k*)

lemma *Big-ZZ-8-5*:

assumes $0<\mu 0$ $\mu 1<1$
shows $\forall^\infty l.\ \forall \mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-ZZ-8-5* μ l
⟨*proof*⟩

lemma (**in** *Book*) *ZZ-8-5*:
assumes *big*: *Big-ZZ-8-5* μ l
defines $\mathcal{R} \equiv$ *Step-class* {*red-step*} **and** $\mathcal{S} \equiv$ *Step-class* {*dboost-step*}
shows *card* $\mathcal{S} \le$ (*bigbeta* / (1 − *bigbeta*)) ∗ *card* $\mathcal{R}$
+ (2 / (1−μ)) ∗ *k powr* (19/20)
⟨*proof*⟩

8.3 Lemma 8.6

For some reason this was harder than it should have been. It does require a further small limit argument.

definition *Big-ZZ-8-6* ≡
$\lambda\mu$ l. *Big-ZZ-8-5* μ l ∧ (∀ $k\ge l$. 2 / (1−μ) ∗ *k powr* (19/20) < *k powr* (39/40))

lemma *Big-ZZ-8-6*:
assumes $0<\mu 0$ $\mu 1<1$
shows $\forall^\infty l.\ \forall \mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow$ *Big-ZZ-8-6* μ l
⟨*proof*⟩

lemma (**in** *Book*) *ZZ-8-6*:
assumes *big*: *Big-ZZ-8-6* μ l
defines $\mathcal{R} \equiv$ *Step-class* {*red-step*} **and** $\mathcal{S} \equiv$ *Step-class* {*dboost-step*}
and $a \equiv 2$ / (1−μ)
assumes *s-ge*: *card* $\mathcal{S} \ge$ *k powr* (39/40)
shows *bigbeta* ≥ (1 − *a* ∗ *k powr* (−1/40)) ∗ (*card* $\mathcal{S}$ / (*card* $\mathcal{S}$ + *card* $\mathcal{R}$))
⟨*proof*⟩

end

9 An exponential improvement far from the diagonal

theory *Far-From-Diagonal*
imports *Zigzag Stirling-Formula.Stirling-Formula*

begin

9.1 An asymptotic form for binomial coefficients via Stirling's formula

From Appendix D.3, page 56

lemma *const-smallo-real*: ($\lambda n.\ x$) ∈ *o*(*real*)
⟨*proof*⟩

lemma *o-real-shift*:
  assumes $f \in o(real)$
  shows $(\lambda i.\ f(i+j)) \in o(real)$
  ⟨*proof*⟩

lemma *tendsto-zero-imp-o1*:
  fixes $a :: nat \Rightarrow real$
  assumes $a \longrightarrow 0$
  shows $a \in o(1)$
⟨*proof*⟩

9.2 Fact D.3 from the Appendix

And hence, Fact 9.4

definition $stir \equiv \lambda n.\ fact\ n\ /\ (sqrt\ (2*pi*n) * (n\ /\ exp\ 1)\ \hat{}\ n) - 1$

Generalised to the reals to allow derivatives

definition $stirG \equiv \lambda n.\ Gamma\ (n+1)\ /\ (sqrt\ (2*pi*n) * (n\ /\ exp\ 1)\ powr\ n) - 1$

lemma *stir-eq-stirG*: $n>0 \Longrightarrow stir\ n = stirG\ (real\ n)$
  ⟨*proof*⟩

lemma *stir-ge0*: $n>0 \Longrightarrow stir\ n \geq 0$
  ⟨*proof*⟩

lemma *stir-to-0*: $stir \longrightarrow 0$
  ⟨*proof*⟩

lemma *stir-o1*: $stir \in o(1)$
  ⟨*proof*⟩

lemma *fact-eq-stir-times*: $n \neq 0 \Longrightarrow fact\ n = (1 + stir\ n) * (sqrt\ (2*pi*n) * (n\ /\ exp\ 1)\ \hat{}\ n)$
  ⟨*proof*⟩

definition $logstir \equiv \lambda n.\ if\ n=0\ then\ 0\ else\ log\ 2\ ((1 + stir\ n) * sqrt\ (2*pi*n))$

lemma *logstir-o-real*: $logstir \in o(real)$
⟨*proof*⟩

lemma *logfact-eq-stir-times*:
  $fact\ n = 2\ powr\ (logstir\ n) * (n\ /\ exp\ 1)\ \hat{}\ n$
⟨*proof*⟩

lemma *mono-G*:
  defines $G \equiv (\lambda x::real.\ Gamma\ (x + 1)\ /\ (x\ /\ exp\ 1)\ powr\ x)$
  shows $mono\text{-}on\ \{0<..\}\ G$

⟨*proof*⟩

lemma *mono-logstir*: *mono logstir*
⟨*proof*⟩

definition *ok-fun-94* $\equiv \lambda k.\ -\ logstir\ k$

lemma *ok-fun-94*: *ok-fun-94* $\in o(real)$
⟨*proof*⟩

lemma *fact-9-4*:
assumes *l*: $0 < l\ l \leq k$
defines $\gamma \equiv l\ /\ (real\ k + real\ l)$
shows $k{+}l\ choose\ l \geq 2\ powr\ ok\text{-}fun\text{-}94\ k * \gamma\ powr\ (-l) * (1{-}\gamma)\ powr\ (-k)$
⟨*proof*⟩

9.3 Fact D.2

For Fact 9.6

lemma *D2*:
fixes *k l*
assumes *t*: $0{<}t\ t \leq k$
defines $\gamma \equiv l\ /\ (real\ k + real\ l)$
shows $(k{+}l{-}t\ choose\ l) \leq exp\ (-\ \gamma * (t{-}1)\hat{}2\ /\ (2{*}k)) * (k\ /\ (k{+}l))\hat{}t * (k{+}l\ choose\ l)$
⟨*proof*⟩

Statement borrowed from Bhavik; no o(k) function

corollary *Far-9-6*:
fixes *k l*
assumes *t*: $0{<}t\ t \leq k$
defines $\gamma \equiv l\ /\ (k + real\ l)$
shows $exp\ (-1) * (1{-}\gamma)\ powr\ (-\ real\ t) * exp\ (\gamma * (real\ t)^2\ /\ real(2{*}k)) * (k{-}t{+}l\ choose\ l) \leq (k{+}l\ choose\ l)$
⟨*proof*⟩

9.4 Lemma 9.3

definition *ok-fun-93g* $\equiv \lambda\gamma\ k.\ (nat\ \lceil k\ powr\ (3/4)\rceil) * log\ 2\ k - (ok\text{-}fun\text{-}71\ \gamma\ k + ok\text{-}fun\text{-}94\ k) + 1$

lemma *ok-fun-93g*:
assumes $0 < \gamma\ \gamma < 1$
shows *ok-fun-93g* $\gamma \in o(real)$
⟨*proof*⟩

definition *ok-fun-93h* $\equiv \lambda\gamma\ k.\ (2\ /\ (1{-}\gamma)) * k\ powr\ (19/20) * (ln\ \gamma + 2 * ln\ k) + ok\text{-}fun\text{-}93g\ \gamma\ k * ln\ 2$

lemma *ok-fun-93h*:
assumes $0 < \gamma$ $\gamma < 1$
shows *ok-fun-93h* $\gamma \in o(real)$
⟨*proof*⟩

lemma *ok-fun-93h-uniform*:
assumes $\mu 01$: $0<\mu 0$ $\mu 1<1$
assumes $e>0$
shows $\forall^\infty k.\ \forall \mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow |ok\text{-}fun\text{-}93h\ \mu\ k|\ /\ k \leq e$
⟨*proof*⟩

context *P0-min*
begin

definition *Big-Far-9-3* ≡
$\lambda\mu\ l.\ Big\text{-}ZZ\text{-}8\text{-}5\ \mu\ l \wedge Big\text{-}X\text{-}7\text{-}1\ \mu\ l \wedge Big\text{-}Y\text{-}6\text{-}2\ \mu\ l \wedge Big\text{-}Red\text{-}5\text{-}3\ \mu\ l$
$\wedge (\forall k \geq l.\ p0\text{-}min - 3 * eps\ k > 1/k \wedge k \geq 2$
$\wedge |ok\text{-}fun\text{-}93h\ \mu\ k\ /\ (\mu * (1 + 1\ /\ (exp\ 1 * (1-\mu))))|\ /\ k \leq 0.667 -$
$2/3)$

lemma *Big-Far-9-3*:
assumes $0<\mu 0$ $\mu 0 \leq \mu 1$ $\mu 1<1$
shows $\forall^\infty l.\ \forall \mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow Big\text{-}Far\text{-}9\text{-}3\ \mu\ l$
⟨*proof*⟩

end

lemma $(\lambda k.\ (nat\ \lceil real\ k\ powr\ (3/4)\rceil) * log\ 2\ k) \in o(real)$
⟨*proof*⟩

lemma *RN34-le-2powr-ok*:
fixes $l\ k::nat$
assumes $l \leq k$ $0<k$
defines $l34 \equiv nat\ \lceil real\ l\ powr\ (3/4)\rceil$
shows $RN\ k\ l34 \leq 2\ powr\ (\lceil k\ powr\ (3/4)\rceil * log\ 2\ k)$
⟨*proof*⟩

Here n really refers to the cardinality of V, so actually nV

lemma (**in** *Book'*) *Far-9-3*:
defines $\delta \equiv min\ (1/200)\ (\gamma/20)$
defines $\mathcal{R} \equiv Step\text{-}class\ \{red\text{-}step\}$
defines $t \equiv card\ \mathcal{R}$
assumes $\gamma 15$: $\gamma \leq 1/5$ **and** $p0$: $p0 \geq 1/4$
and *nge*: $n \geq exp\ (-\delta * real\ k) * (k+l\ choose\ l)$
and *X0ge*: $card\ X0 \geq n/2$
— Because $n\ /\ 2 \leq real\ (card\ X0)$ makes the proof harder
assumes *big*: $Big\text{-}Far\text{-}9\text{-}3\ \gamma\ l$
shows $t \geq 2*k\ /\ 3$
⟨*proof*⟩

9.5 Lemma 9.5

context *P0-min*
begin

Again stolen from Bhavik: cannot allow a dependence on γ

definition *ok-fun-95a* $\equiv \lambda k.$ *ok-fun-61 k* $-$ (*2* + *4* $*$ *k powr* (*19/20*))

definition *ok-fun-95b* $\equiv \lambda k.$ *ln 2* $*$ *ok-fun-95a k* $-$ *1*

lemma *ok-fun-95a*: *ok-fun-95a* $\in$ *o*(*real*)
⟨*proof*⟩

lemma *ok-fun-95b*: *ok-fun-95b* $\in$ *o*(*real*)
⟨*proof*⟩

definition *Big-Far-9-5* $\equiv \lambda\mu$ *l*. *Big-Red-5-3* μ *l* $\wedge$ *Big-Y-6-1* μ *l* $\wedge$ *Big-ZZ-8-5* μ *l*

lemma *Big-Far-9-5*:
assumes *0*<μ*0* μ*1*<*1*
shows $\forall^\infty$*l*. $\forall\mu$. μ*0* $\leq \mu \wedge \mu \leq$ μ*1* $\longrightarrow$ *Big-Far-9-5* μ *l*
⟨*proof*⟩

end

Y0 is an additional assumption found in Bhavik's version. (He had a couple of others). The first $o(k)$ function adjusts for the error in $n/2$

lemma (**in** *Book'*) *Far-9-5*:
fixes δ η::*real*
defines $\mathcal{R} \equiv$ *Step-class* {*red-step*}
defines *t* $\equiv$ *card* $\mathcal{R}$
assumes *nV*: *real nV* $\geq$ *exp* ($-\delta$ $*$ *k*) $*$ (*k*+*l choose l*) **and** *Y0*: *card Y0* $\geq$ *nV div 2*
assumes *p0*: *1/2* $\leq$ *1*$-\gamma-\eta$ *1*$-\gamma-\eta$ $\leq$ *p0* **and** *0*$\leq\eta$
assumes *big*: *Big-Far-9-5* γ *l*
shows *card* (*Yseq halted-point*) $\geq$
exp ($-\delta$ $*$ *k* + *ok-fun-95b k*) $*$ (*1*$-\gamma-\eta$) *powr* ($\gamma$$*$*t* / (*1*$-\gamma$)) $*$ ((*1*$-\gamma-\eta$)/(*1*$-\gamma$)) ^*t*

$*$ *exp* (γ $*$ (*real t*)^2 / (*2*$*$*k*)) $*$ (*k*$-$*t*+*l choose l*)   (**is** - $\geq$ *?rhs*)
⟨*proof*⟩

9.6 Lemma 9.2 actual proof

context *P0-min*
begin

lemma *error-9-2*:
assumes μ>*0* *d* > *0*
shows $\forall^\infty$*k*. *ok-fun-95b k* + μ $*$ *real k* / *d* $\geq$ *0*

⟨*proof*⟩

definition *Big-Far-9-2* $\equiv$ $\lambda\mu$ *l*. *Big-Far-9-3* μ *l* $\wedge$ *Big-Far-9-5* μ *l* $\wedge$ ($\forall k \geq l$. *ok-fun-95b k* + $\mu * k/60 \geq 0$)

lemma *Big-Far-9-2*:
  assumes $0<\mu 0$ $\mu 0 \leq \mu 1$ $\mu 1<1$
  shows $\forall^\infty l$. $\forall \mu$. $\mu 0 \leq \mu \wedge \mu \leq \mu 1 \longrightarrow$ *Big-Far-9-2* μ *l*
⟨*proof*⟩

end

lemma (**in** *Book'*) *Far-9-2-conclusion*:
  defines $\mathcal{R} \equiv$ *Step-class* {*red-step*}
  defines $t \equiv$ *card* $\mathcal{R}$
  assumes *Y*: ($k-t+l$ *choose l*) $\leq$ *card* (*Yseq halted-point*)
  shows *False*
⟨*proof*⟩

A little tricky to express since the Book locale assumes that there are no cliques in the original graph (page 9). So it's a contrapositive

lemma (**in** *Book'*) *Far-9-2-aux*:
  fixes δ η::*real*
  defines $\delta \equiv \gamma/20$
  assumes *0*: *real* (*card X0*) $\geq nV/2$ *card Y0* $\geq nV$ *div 2 p0* $\geq 1-\gamma-\eta$
    — These are the assumptions about the red density of the graph
  assumes γ: $\gamma \leq 1/10$ **and** η: $0 \leq \eta$ $\eta \leq \gamma/15$
  assumes *nV*: *real nV* $\geq$ *exp* $(-\delta * k) *$ ($k+l$ *choose l*)
  assumes *big*: *Big-Far-9-2* γ *l*
  shows *False*
⟨*proof*⟩

Mediation of 9.2 (and 10.2) from locale *Book-Basis* to the book locales with the starting sets of equal size

lemma (**in** *No-Cliques*) *Basis-imp-Book*:
  assumes *gd*: *p0-min* $\leq$ *graph-density Red*
  assumes $\mu 01$: $0 < \mu$ $\mu < 1$
  obtains *X0 Y0* **where** $l \geq 2$ *card X0* $\geq$ *real nV* / *2 card Y0* = *gorder div 2*
    and *X0* = *V* \ *Y0 Y0* $\subseteq$ *V*
    and *graph-density Red* $\leq$ *gen-density Red X0 Y0*
    and *Book V E p0-min Red Blue l k* μ *X0 Y0*
⟨*proof*⟩

Material that needs to be proved **outside** the book locales

As above, for *Book'*

lemma (**in** *No-Cliques*) *Basis-imp-Book'*:
  assumes *gd*: *p0-min* $\leq$ *graph-density Red*
  assumes *l*: $0<l$ $l \leq k$

obtains *X0 Y0* **where** $l \geq 2$ *card X0* $\geq$ *real nV* / *2 card Y0* = *gorder div 2* **and** *X0* = *V* \ *Y0 Y0*$\subseteq$*V*
and *graph-density Red* $\leq$ *gen-density Red X0 Y0*
and *Book' V E p0-min Red Blue l k* (*real l* / (*real k* + *real l*)) *X0 Y0*
⟨*proof*⟩

lemma (**in** *No-Cliques*) *Far-9-2*:
fixes δ γ η::*real*
defines $\gamma \equiv l$ / (*real k* + *real l*)
defines $\delta \equiv \gamma/20$
assumes *nV*: *real nV* $\geq$ *exp* ($-\delta * k$) $*$ (*k*+*l choose l*)
assumes *gd*: *graph-density Red* $\geq 1-\gamma-\eta$ **and** *p0-min-OK*: *p0-min* $\leq 1-\gamma-\eta$
assumes *big*: *Big-Far-9-2* γ *l*
assumes $\gamma \leq 1/10$ **and** η: $0 \leq \eta$ $\eta \leq \gamma/15$
shows *False*
⟨*proof*⟩

9.7 Theorem 9.1

An arithmetical lemma proved outside of the locales

lemma *kl-choose*:
fixes *l k*::*nat*
assumes *m*<*l k*>*0*
defines *PM* $\equiv \prod$ *i*<*m*. (*l* − *real i*) / (*k*+*l*−*real i*)
shows (*k*+*l choose l*) = (*k*+*l*−*m choose* (*l*−*m*)) / *PM*
⟨*proof*⟩

context *P0-min*
begin

The proof considers a smaller graph, so l needs to be so big that the smaller l' will be big enough.

definition *Big-Far-9-1* :: *real* $\Rightarrow$ *nat* $\Rightarrow$ *bool* **where**
Big-Far-9-1 $\equiv \lambda\mu$ *l*. $l \geq 3 \wedge (\forall l'\ \gamma.$ *real* $l' \geq (10/11) * \mu *$ *real* $l \longrightarrow \mu^2 \leq \gamma \wedge \gamma \leq 1/10 \longrightarrow$ *Big-Far-9-2* $\gamma\ l'$)

The proof of theorem 10.1 requires a range of values

lemma *Big-Far-9-1*:
assumes $0<\mu 0$ $\mu 0 \leq 1/10$
shows $\forall^\infty l.\ \forall \mu.\ \mu 0 \leq \mu \wedge \mu \leq 1/10 \longrightarrow$ *Big-Far-9-1* μ *l*
⟨*proof*⟩

The text claims the result for all k and l, not just those sufficiently large, but the $o(k)$ function allowed in the exponent provides a fudge factor

theorem *Far-9-1*:
fixes *l k*::*nat*
fixes δ γ::*real*
defines $\gamma \equiv$ *real l* / (*real k* + *real l*)

defines $\delta \equiv \gamma/20$
assumes γ: $\gamma \le 1/10$
assumes *big*: *Big-Far-9-1* γ *l*
assumes *p0-min-91*: *p0-min* $\le 1 - (1/10) * (1 + 1/15)$
shows *RN k l* $\le$ *exp* $(-\delta * k + 1) * (k+l \text{ choose } l)$
⟨*proof*⟩

end

end

10 An exponential improvement closer to the diagonal

theory *Closer-To-Diagonal*
imports *Far-From-Diagonal*

begin

10.1 Lemma 10.2

context *P0-min*
begin

lemma *error-10-2*:
assumes μ / *real d* $> 1/200$
shows $\forall^\infty k$. *ok-fun-95b k* + μ ∗ *real k* / *real d* $\ge k/200$
⟨*proof*⟩

The "sufficiently large" assumptions are problematical. The proof's calculation for $(3::'a) / (20::'a) < \gamma$ is sharp. We need a finite gap for the limit to exist. We can get away with 1/300.

definition *x320::real* **where** *x320* $\equiv 3/20 + 1/300$

lemma *error-10-2-True*: $\forall^\infty k$. *ok-fun-95b k* + *x320* ∗ *real k* / *real 30* $\ge k/200$
⟨*proof*⟩

lemma *error-10-2-False*: $\forall^\infty k$. *ok-fun-95b k* + $(1/10)$ ∗ *real k* / *real 15* $\ge k/200$
⟨*proof*⟩

definition *Big-Closer-10-2* $\equiv \lambda\mu$ *l*. *Big-Far-9-3* μ *l* $\wedge$ *Big-Far-9-5* μ *l*
$\wedge$ ($\forall k \ge l$. *ok-fun-95b k* + (*if* $\mu >$ *x320 then* $\mu * k/30$ *else* $\mu * k/15$) $\ge$ $k/200$)

lemma *Big-Closer-10-2*:
assumes $1/10 \le \mu 1$ $\mu 1 < 1$
shows $\forall^\infty l$. $\forall \mu$. $1/10 \le \mu \wedge \mu \le \mu 1 \longrightarrow$ *Big-Closer-10-2* μ *l*
⟨*proof*⟩

end

A little tricky to express since the Book locale assumes that there are no cliques in the original graph (page 10). So it's a contrapositive

lemma (**in** *Book'*) *Closer-10-2-aux*:
  assumes *0*: *real* (*card X0*) $\geq$ *nV*/*2 card Y0* $\geq$ *nV div 2 p0* $\geq$ *1*$-\gamma$
    — These are the assumptions about the red density of the graph
  assumes γ: *1/10* $\leq \gamma$ $\gamma \leq$ *1/5*
  assumes *nV*: *real nV* $\geq$ *exp* ($-k$/*200*) $*$ (*k*+*l choose l*)
  assumes *big*: *Big-Closer-10-2* γ *l*
  shows *False*
⟨*proof*⟩

Material that needs to be proved **outside** the book locales

lemma (**in** *No-Cliques*) *Closer-10-2*:
  fixes γ::*real*
  defines $\gamma \equiv$ *l* / (*real k* + *real l*)
  assumes *nV*: *real nV* $\geq$ *exp* ($-$ *real k*/*200*) $*$ (*k*+*l choose l*)
  assumes *gd*: *graph-density Red* $\geq$ *1*$-\gamma$ **and** *p0-min-OK*: *p0-min* $\leq$ *1*$-\gamma$
  assumes *big*: *Big-Closer-10-2* γ *l* **and** *l*$\leq$*k*
  assumes γ: *1/10* $\leq \gamma$ $\gamma \leq$ *1/5*
  shows *False*
⟨*proof*⟩

10.2 Theorem 10.1

context *P0-min*
begin

definition *Big101a* $\equiv \lambda k.$ *2* + *real k* / *2* $\leq$ *exp* (*of-int*$\lfloor k/10 \rfloor$ $*$ *2* $-$ *k*/*200*)

definition *Big101b* $\equiv \lambda k.$ (*real k*)^2 $-$ *10* $*$ *real k* $>$ (*k*/*10*) $*$ *real*(*10* + *9*$*$*k*)

The proof considers a smaller graph, so l needs to be so big that the smaller l' will be big enough.

definition *Big101c* $\equiv \lambda\gamma 0\ l.\ \forall l'\ \gamma.\ l' \geq$ *nat* $\lfloor 2/5 * l \rfloor \longrightarrow \gamma 0 \leq \gamma \longrightarrow \gamma \leq 1/10$ $\longrightarrow$ *Big-Far-9-1* γ l'

definition *Big101d* $\equiv \lambda l.$ ($\forall l'\ \gamma.\ l' \geq$ *nat* $\lfloor 2/5 * l \rfloor \longrightarrow 1/10 \leq \gamma \longrightarrow \gamma \leq 1/5$ $\longrightarrow$ *Big-Closer-10-2* γ l')

definition *Big-Closer-10-1* $\equiv \lambda\gamma 0\ l.\ l \geq 9 \wedge (\forall k \geq l.$ *Big101c* $\gamma 0$ *k* $\wedge$ *Big101d k* $\wedge$ *Big101a k* $\wedge$ *Big101b k*)

lemma *Big-Closer-10-1-upward*: ⟦*Big-Closer-10-1* $\gamma 0$ *l*; $l \leq k$; $\gamma 0 \leq \gamma$⟧ $\Longrightarrow$ *Big-Closer-10-1* γ *k*
  ⟨*proof*⟩

The need for $\gamma 0$ is unfortunate, but it seems simpler to hide the precise value of this term in the main proof.

lemma *Big-Closer-10-1*:
fixes $\gamma 0$::*real*
assumes $\gamma 0>0$
shows $\forall^\infty l$. *Big-Closer-10-1* $\gamma 0$ *l*
⟨*proof*⟩

The strange constant $\gamma 0$ is needed for the case where we consider a subgraph; see near the end of this proof

theorem *Closer-10-1*:
fixes *l k*::*nat*
fixes δ γ::*real*
defines $\gamma \equiv$ *real l* / (*real k* + *real l*)
defines $\delta \equiv \gamma/40$
defines $\gamma 0 \equiv$ *min* γ (0.07) — Since $36 \leq k$, the lower bound (1::′a) / (10::′a) − (1::′a) / (36::′a) works
assumes *big*: *Big-Closer-10-1* $\gamma 0$ *l*
assumes γ: $\gamma \leq 1/5$
assumes *p0-min-101*: *p0-min* $\leq 1 - 1/5$
shows *RN k l* $\leq$ *exp* $(-\delta * k + 3) * (k+l$ *choose l*)
⟨*proof*⟩

definition *ok-fun-10-1* $\equiv \lambda\gamma\ k.$ *if Big-Closer-10-1* (*min* γ 0.07) (*nat*$\lceil((\gamma / (1-\gamma)) * k)\rceil$) *then* 3 *else* $(\gamma/40 * k)$

lemma *ok-fun-10-1*:
assumes $0 < \gamma$ $\gamma < 1$
shows *ok-fun-10-1* $\gamma \in o(real)$
⟨*proof*⟩

theorem *Closer-10-1-unconditional*:
fixes *l k*::*nat*
fixes δ γ::*real*
defines $\gamma \equiv$ *real l* / (*real k* + *real l*)
defines $\delta \equiv \gamma/40$
assumes γ: $0 < \gamma$ $\gamma \leq 1/5$
assumes *p0-min-101*: *p0-min* $\leq 1 - 1/5$
shows *RN k l* $\leq$ *exp* $(-\delta * k +$ *ok-fun-10-1* γ $k) * (k+l$ *choose l*)
⟨*proof*⟩

end

end

11 From diagonal to off-diagonal

theory *From-Diagonal*

imports *Closer-To-Diagonal*

begin

11.1 Lemma 11.2

definition *ok-fun-11-2a* $\equiv \lambda k.\ \lceil real\ k\ powr\ (3/4) \rceil * log\ 2\ k$

definition *ok-fun-11-2b* $\equiv \lambda\mu\ k.\ k\ powr\ (39/40) * (log\ 2\ \mu + 3 * log\ 2\ k)$

definition *ok-fun-11-2c* $\equiv \lambda\mu\ k.\ -\ k * log\ 2\ (1 - (2\ /\ (1-\mu)) * k\ powr\ (-1/40))$

definition *ok-fun-11-2* $\equiv \lambda\mu\ k.\ 2 - ok\text{-}fun\text{-}71\ \mu\ k + ok\text{-}fun\text{-}11\text{-}2a\ k$
$+ max\ (ok\text{-}fun\text{-}11\text{-}2b\ \mu\ k)\ (ok\text{-}fun\text{-}11\text{-}2c\ \mu\ k)$

lemma *ok-fun-11-2a*: *ok-fun-11-2a* $\in o(real)$
⟨*proof*⟩

possibly, the functions that depend upon μ need a more refined analysis to cover a closed interval of possible values. But possibly not, as the text implies $\mu = (2::'a)\ /\ (5::'a)$.

lemma *ok-fun-11-2b*: *ok-fun-11-2b* $\mu \in o(real)$
⟨*proof*⟩

lemma *ok-fun-11-2c*: *ok-fun-11-2c* $\mu \in o(real)$
⟨*proof*⟩

lemma *ok-fun-11-2*:
assumes $0<\mu\ \ \mu<1$
shows *ok-fun-11-2* $\mu \in o(real)$
⟨*proof*⟩

definition *Big-From-11-2* $\equiv$
$\lambda\mu\ k.\ Big\text{-}ZZ\text{-}8\text{-}6\ \mu\ k \wedge Big\text{-}X\text{-}7\text{-}1\ \mu\ k \wedge Big\text{-}Y\text{-}6\text{-}2\ \mu\ k \wedge Big\text{-}Red\text{-}5\text{-}3\ \mu\ k \wedge Big\text{-}Blue\text{-}4\text{-}1\ \mu\ k$
$\wedge\ 1 \le \mu\hat{}2 * real\ k \wedge 2\ /\ (1-\mu) * real\ k\ powr\ (-1/40) < 1 \wedge 1/k < 1/2 - 3 * eps\ k$

lemma *Big-From-11-2*:
assumes $0<\mu 0\ \ \mu 0 \le \mu 1\ \ \mu 1<1$
shows $\forall^\infty k.\ \forall\mu.\ \mu \in \{\mu 0..\mu 1\} \longrightarrow Big\text{-}From\text{-}11\text{-}2\ \mu\ k$
⟨*proof*⟩

Simply to prevent issues about the positioning of the function *real*

abbreviation *ratio* $\equiv \lambda\mu\ s\ t.\ \mu * (real\ s + real\ t)\ /\ real\ s$

the text refers to the actual Ramsey number but I don't see how that could work. Theorem 11.1 will define n to be one less than the Ramsey number, hence we add that one back here.

lemma (**in** *Book*) *From-11-2*:
assumes $l=k$
assumes *big*: *Big-From-11-2* μ *k*
defines $\mathcal{R} \equiv$ *Step-class* {*red-step*} **and** $\mathcal{S} \equiv$ *Step-class* {*dboost-step*}
defines $t \equiv$ *card* $\mathcal{R}$ **and** $s \equiv$ *card* $\mathcal{S}$
defines $nV' \equiv$ *Suc nV*
assumes *0*: *card X0* $\geq$ *nV div 2* **and** *p0* $\geq 1/2$
shows *log 2 nV'* $\leq k * $ *log 2* $(1/\mu) + t *$ *log 2* $(1 / (1-\mu)) + s *$ *log 2* (*ratio* μ *s t*) + *ok-fun-11-2* μ *k*
⟨*proof*⟩

11.2 Lemma 11.3

same remark as in Lemma 11.2 about the use of the Ramsey number in the conclusion

lemma (**in** *Book*) *From-11-3*:
assumes $l=k$
assumes *big*: *Big-Y-6-1* μ *k*
defines $\mathcal{R} \equiv$ *Step-class* {*red-step*} **and** $\mathcal{S} \equiv$ *Step-class* {*dboost-step*}
defines $t \equiv$ *card* $\mathcal{R}$ **and** $s \equiv$ *card* $\mathcal{S}$
defines $nV' \equiv$ *Suc nV*
assumes *0*: *card Y0* $\geq$ *nV div 2* **and** *p0* $\geq 1/2$
shows *log 2 nV'* $\leq$ *log 2* (*RN k* $(k-t)$) $+ s + t + 2 -$ *ok-fun-61 k*
⟨*proof*⟩

11.3 Theorem 11.1

definition *FF* :: *nat* ⇒ *real* ⇒ *real* ⇒ *real* **where**
FF ≡ λ*k x y*. *log 2* (*RN k* (*nat*⌊*real k* − *x* ∗ *real k*⌋)) / *real k* + *x* + *y*

definition *GG* :: *real* ⇒ *real* ⇒ *real* ⇒ *real* **where**
GG ≡ λμ *x y*. *log 2* $(1/\mu) + x *$ *log 2* $(1/(1-\mu)) + y *$ *log 2* $(\mu * (x+y) / y)$

definition *FF-bound* :: *nat* ⇒ *real* ⇒ *real* **where**
FF-bound ≡ λ*k u*. *FF k 0 u* + *1*

lemma *log2-RN-ge0*: $0 \leq$ *log 2* (*RN k k*) / *k*
⟨*proof*⟩

lemma *le-FF-bound*:
assumes *x*: $x \in \{0..1\}$ **and** $y \in \{0..u\}$
shows *FF k x y* $\leq$ *FF-bound k u*
⟨*proof*⟩

lemma *FF2*: $y' \leq y \Longrightarrow$ *FF k x y'* $\leq$ *FF k x y*
⟨*proof*⟩

lemma *FF-GG-bound*:

assumes μ: $0 < \mu$ $\mu < 1$ **and** x: $x \in \{0..1\}$ **and** y: $y \in \{0..\mu * x / (1-\mu) + \eta\}$
shows $min\ (FF\ k\ x\ y)\ (GG\ \mu\ x\ y) + \eta \leq FF\text{-}bound\ k\ (\mu / (1-\mu) + \eta) + \eta$
⟨*proof*⟩

context *P0-min*
begin

definition $ok\text{-}fun\text{-}11\text{-}1 \equiv \lambda\mu\ k.\ max\ (ok\text{-}fun\text{-}11\text{-}2\ \mu\ k)\ (2 - ok\text{-}fun\text{-}61\ k)$

lemma *ok-fun-11-1*:
assumes $0<\mu$ $\mu<1$
shows $ok\text{-}fun\text{-}11\text{-}1\ \mu \in o(real)$
⟨*proof*⟩

lemma *eventually-ok111-le-*η:
assumes $\eta > 0$ **and** μ: $0<\mu$ $\mu<1$
shows $\forall^\infty k.\ ok\text{-}fun\text{-}11\text{-}1\ \mu\ k / k \leq \eta$
⟨*proof*⟩

lemma *eventually-powr-le-*η:
assumes $\eta > 0$
shows $\forall^\infty k.\ (2 / (1-\mu)) * k\ powr\ (-1/20) \leq \eta$
⟨*proof*⟩

definition *Big-From-11-1* $\equiv$
$\lambda\eta\ \mu\ k.\ Big\text{-}From\text{-}11\text{-}2\ \mu\ k \wedge Big\text{-}ZZ\text{-}8\text{-}5\ \mu\ k \wedge Big\text{-}Y\text{-}6\text{-}1\ \mu\ k \wedge ok\text{-}fun\text{-}11\text{-}1\ \mu\ k / k \leq \eta/2$
$\wedge\ (2 / (1-\mu)) * k\ powr\ (-1/20) \leq \eta/2$
$\wedge\ Big\text{-}Closer\text{-}10\text{-}1\ (1/101)\ (nat\lceil k/100\rceil) \wedge 3 / (k * ln\ 2) \leq \eta/2 \wedge k\geq 3$

In sections 9 and 10 (and by implication all proceeding sections), we needed to consider a closed interval of possible values of μ. Let's hope, maybe not here. The fact below can only be proved with the strict inequality $(0::'a) < \eta$, which is why it is also strict in the theorems depending on this property.

lemma *Big-From-11-1*:
assumes $\eta > 0$ $0<\mu$ $\mu<1$
shows $\forall^\infty k.\ Big\text{-}From\text{-}11\text{-}1\ \eta\ \mu\ k$
⟨*proof*⟩

The actual proof of theorem 11.1 is now combined with the development of section 12, since the concepts seem to be inescapably mixed up.

end

end

12 The Proof of Theorem 1.1

theory *The-Proof*
imports *From-Diagonal*

begin

12.1 The bounding functions

definition $H \equiv \lambda p.\ -p * log\ 2\ p - (1-p) * log\ 2\ (1-p)$

definition *dH* **where** $dH \equiv \lambda x{::}real.\ -ln(x)/ln(2) + ln(1 - x)/ln(2)$

lemma *dH* [*derivative-intros*]:
assumes $0<x\ x<1$
shows (*H has-real-derivative dH x*) (*at x*)
⟨*proof*⟩

lemma *H0* [*simp*]: $H\ 0 = 0$ **and** *H1* [*simp*]: $H\ 1 = 0$
⟨*proof*⟩

lemma *H-reflect*: $H\ (1-p) = H\ p$
⟨*proof*⟩

lemma *H-ge0*:
assumes $0 \le p\ p \le 1$
shows $0 \le H\ p$
⟨*proof*⟩

Going up, from 0 to 1/2

lemma *H-half-mono*:
assumes $0 \le p'\ p' \le p\ p \le 1/2$
shows $H\ p' \le H\ p$
⟨*proof*⟩

Going down, from 1/2 to 1

lemma *H-half-mono′*:
assumes $1/2 \le p'\ p' \le p\ p \le 1$
shows $H\ p' \ge H\ p$
⟨*proof*⟩

lemma *H-half*: $H(1/2) = 1$
⟨*proof*⟩

lemma *H-le1*:
assumes $0 \le p\ p \le 1$
shows $H\ p \le 1$
⟨*proof*⟩

Many thanks to Fedor Petrov on mathoverflow

lemma *H-12-1*:
  fixes *a b::nat*
  assumes $a \geq b$
  shows $log\ 2\ (a\ choose\ b) \leq a * H(b/a)$
⟨*proof*⟩

definition $gg \equiv GG\ (2/5)$

lemma *gg-eq*: $gg\ x\ y = log\ 2\ (5/2) + x * log\ 2\ (5/3) + y * log\ 2\ ((2 * (x+y)) / (5*y))$
  ⟨*proof*⟩

definition $f1 \equiv \lambda x\ y.\ x + y + (2-x) * H(1/(2-x))$

definition $f2 \equiv \lambda x\ y.\ f1\ x\ y - (1\ /\ (40 * ln\ 2)) * ((1-x)\ /\ (2-x))$

definition $ff \equiv \lambda x\ y.\ if\ x < 3/4\ then\ f1\ x\ y\ else\ f2\ x\ y$

Incorporating Bhavik's idea, which gives us a lower bound for γ of 1/101

definition *ffGG* :: *real* ⇒ *real* ⇒ *real* ⇒ *real* **where**
  $ffGG \equiv \lambda\mu\ x\ y.\ max\ 1.9\ (min\ (ff\ x\ y)\ (GG\ \mu\ x\ y))$

The proofs involving *Sup* are needlessly difficult because ultimately the sets involved are finite, eliminating the need to demonstrate boundedness. Simpler might be to use the extended reals.

lemma *f1-le*:
  assumes $x \leq 1$
  shows $f1\ x\ y \leq y+2$
  ⟨*proof*⟩

lemma *ff-le4*:
  assumes $x \leq 1\ y \leq 1$
  shows $ff\ x\ y \leq 4$
⟨*proof*⟩

lemma *ff-GG-bound*:
  assumes $x \leq 1\ y \leq 1$
  shows $ffGG\ \mu\ x\ y \leq 4$
  ⟨*proof*⟩

lemma *bdd-above-ff-GG*:
  assumes $x \leq 1\ u \leq 1$
  shows *bdd-above* $((\lambda y.\ ffGG\ \mu\ x\ y + \eta)$ ` $\{0..u\})$
  ⟨*proof*⟩

lemma *bdd-above-SUP-ff-GG*:
  assumes $0 \leq u\ u \leq 1$
  shows *bdd-above* $((\lambda x.\ \bigsqcup y \in \{0..u\}.\ ffGG\ \mu\ x\ y + \eta)$ ` $\{0..1\})$
  ⟨*proof*⟩

Claim (62). A singularity if $x = 1$. Okay if we put $\ln(0) = 0$

lemma *FF-le-f1*:
fixes *k*::*nat* **and** *x y*::*real*
assumes *x*: *0* ≤ *x x* ≤ *1* **and** *y*: *0* ≤ *y y* ≤ *1*
shows *FF k x y* ≤ *f1 x y*
⟨*proof*⟩

Bhavik's *eleven-one-large-end*

lemma *f1-le-19*:
fixes *k*::*nat* **and** *x y*::*real*
assumes *x*: *0.99* ≤ *x x* ≤ *1* **and** *y*: *0* ≤ *y y* ≤ *3/4*
shows *f1 x y* ≤ *1.9*
⟨*proof*⟩

Claim (63) in weakened form; we get rid of the extra bit later

lemma (**in** *P0-min*) *FF-le-f2*:
fixes *k*::*nat* **and** *x y*::*real*
assumes *x*: *3/4* ≤ *x x* ≤ *1* **and** *y*: *0* ≤ *y y* ≤ *1*
and *l*: *real l* = *k* − *x*∗*k*
assumes *p0-min-101*: *p0-min* ≤ *1* − *1/5*
defines *γ* ≡ *real l* / (*real k* + *real l*)
defines *γ0* ≡ *min γ* (*0.07*)
assumes *γ* > *0*
shows *FF k x y* ≤ *f2 x y* + *ok-fun-10-1 γ k* / (*k* ∗ *ln 2*)
⟨*proof*⟩

The body of the proof has been extracted to allow the symmetry argument. And 1/12 is 3/4-2/3, the latter number corresponding to *μ* = (*2*::′*a*) / (*5*::′*a*)

lemma (**in** *Book-Basis*) *From-11-1-Body*:
fixes *V* :: ′*a set*
assumes *μ*: *0* < *μ μ* ≤ *2/5* **and** *η*: *0* < *η η* ≤ *1/12*
and *ge-RN*: *Suc nV* ≥ *RN k k*
and *Red*: *graph-density Red* ≥ *1/2*
and *p0-min12*: *p0-min* ≤ *1/2*
and *Red-E*: *Red* ⊆ *E* **and** *Blue-def*: *Blue* = *E*∖*Red*
and *no-Red-K*: ¬ (∃ *K*. *size-clique k K Red*)
and *no-Blue-K*: ¬ (∃ *K*. *size-clique k K Blue*)
and *big*: *Big-From-11-1 η μ k*
shows *log 2* (*RN k k*) / *k* ≤ (*SUP x* ∈ {*0..1*}. *SUP y* ∈ {*0..3/4*}. *ffGG μ x y* + *η*)
⟨*proof*⟩

theorem (**in** *P0-min*) *From-11-1*:
assumes *μ*: *0* < *μ μ* ≤ *2/5* **and** *η* > *0* **and** *le*: *η* ≤ *1/12*
and *p0-min12*: *p0-min* ≤ *1/2* **and** *big*: *Big-From-11-1 η μ k*
shows *log 2* (*RN k k*) / *k* ≤ (*SUP x* ∈ {*0..1*}. *SUP y* ∈ {*0..3/4*}. *ffGG μ x y* + *η*)
⟨*proof*⟩

12.2 The monster calculation from appendix A

12.2.1 Observation A.1

lemma *gg-increasing*:
  assumes $x \leq x'$ $0 \leq x$ $0 \leq y$
  shows $gg\ x\ y \leq gg\ x'\ y$
⟨*proof*⟩

Thanks to Manuel Eberl

lemma *continuous-on-x-ln*: *continuous-on* $\{0..\}$ $(\lambda x{::}real.\ x * ln\ x)$
⟨*proof*⟩

lemma *continuous-on-f1*: *continuous-on* $\{..1\}$ $(\lambda x.\ f1\ x\ y)$
⟨*proof*⟩

definition *df1* **where** $df1 \equiv \lambda x.\ log\ 2\ (2 * ((1-x)\ /\ (2-x)))$

lemma *Df1* [*derivative-intros*]:
  assumes $x<1$
  shows $((\lambda x.\ f1\ x\ y)$ *has-real-derivative* $df1\ x)\ (at\ x)$
⟨*proof*⟩

definition *delta* **where** $delta \equiv \lambda u{::}real.\ 1\ /\ (ln\ 2 * 40 * (2 - u)^2)$

lemma *Df2*:
  assumes $1/2 \leq x$ $x<1$
  shows $((\lambda x.\ f2\ x\ y)$ *has-real-derivative* $df1\ x + delta\ x)\ (at\ x)$
  ⟨*proof*⟩

lemma *antimono-on-ff*:
  assumes $0 \leq y$ $y < 1$
  shows *antimono-on* $\{1/2..1\}$ $(\lambda x.\ ff\ x\ y)$
⟨*proof*⟩

12.2.2 Claims A.2–A.4

Called simply x in the paper, but are you kidding me?

definition $x\text{-}of \equiv \lambda y{::}real.\ 3*y/5 + 0.5454$

lemma *x-of*: $x\text{-}of \in \{0..3/4\} \rightarrow \{1/2..1\}$
  ⟨*proof*⟩

definition $y\text{-}of \equiv \lambda x{::}real.\ 5 * x/3 - 0.909$

lemma *y-of-x-of* [*simp*]: $y\text{-}of\ (x\text{-}of\ y) = y$
  ⟨*proof*⟩

lemma *x-of-y-of* [*simp*]: $x\text{-}of\ (y\text{-}of\ x) = x$
  ⟨*proof*⟩

lemma *Df1-y* [*derivative-intros*]:
  assumes *x*<*1*
  shows ((λ*x*. *f1 x* (*y-of x*)) *has-real-derivative 5*/*3* + *df1 x*) (*at x*)
⟨*proof*⟩

lemma *Df2-y* [*derivative-intros*]:
  assumes *1*/*2*≤*x x*<*1*
  shows ((λ*x*. *f2 x* (*y-of x*)) *has-real-derivative 5*/*3* + *df1 x* + *delta x*) (*at x*)
  ⟨*proof*⟩

definition *Dg-x* ≡ λ*y*. *3* ∗ *log 2* (*5*/*3*) / *5* + *log 2* ((*2727* + *y* ∗ *8000*) / (*y* ∗ *12500*))
          − *2727* / (*ln 2* ∗ (*2727* + *y* ∗ *8000*))

lemma *Dg-x* [*derivative-intros*]:
  assumes *y* ∈ {*0*<..<*3*/*4*}
  shows ((λ*y*. *gg* (*x-of y*) *y*) *has-real-derivative Dg-x y*) (*at y*)
  ⟨*proof*⟩

Claim A2 is difficult because it comes *real close*: max value = 1.999281, when y = 0.4339. There is no simple closed form for the maximum point (where the derivative goes to 0).

Due to the singularity at zero, we need to cover the zero case analytically, but at least interval arithmetic covers the maximum point

lemma *A2*:
  assumes *y* ∈ {*0*..*3*/*4*}
  shows *gg* (*x-of y*) *y* ≤ *2* − *1*/*2*^*11*
⟨*proof*⟩

lemma *A3*:
  assumes *y* ∈ {*0*..*0*.*341*}
  shows *f1* (*x-of y*) *y* ≤ *2* − *1*/*2*^*11*
⟨*proof*⟩

This one also comes close: max value = 1.999271, when y = 0.4526. The specified upper bound is 1.99951

lemma *A4*:
  assumes *y* ∈ {*0*.*341*..*3*/*4*}
  shows *f2* (*x-of y*) *y* ≤ *2* − *1*/*2*^*11*
  ⟨*proof*⟩

context *P0-min*
begin

The truly horrible Lemma 12.3

lemma *123*:

fixes δ::*real*
assumes $0 < \delta$ $\delta \leq 1\ /\ 2\hat{}11$
shows $(SUP\ x \in \{0..1\}.\ SUP\ y \in \{0..3/4\}.\ ffGG\ (2/5)\ x\ y) \leq 2{-}\delta$
⟨*proof*⟩

end

12.3 Concluding the proof

we subtract a tiny bit, as we seem to need this gap

definition *delta'*::*real* **where** $delta' \equiv 1\ /\ 2\hat{}11\ -\ 1\ /\ 2\hat{}18$

lemma *Aux-1-1*:
assumes *p0-min12*: $p0\text{-}min \leq 1/2$
shows $\forall^{\infty}k.\ log\ 2\ (RN\ k\ k)\ /\ k \leq 2\ -\ delta'$
⟨*proof*⟩

Main theorem 1.1: the exponent is approximately 3.9987

theorem *Main-1-1*:
obtains ε::*real* **where** $\varepsilon{>}0$ $\forall^{\infty}k.\ RN\ k\ k \leq (4{-}\varepsilon)\hat{}k$
⟨*proof*⟩

end

References

[1] M. Campos, S. Griffiths, R. Morris, and J. Sahasrabudhe. An exponential improvement for diagonal Ramsey, 2023. arXiv, 2303.09521.