Real-Valued Special Functions: Upper and Lower Bounds

Lawrence C. Paulson

May 26, 2024

Abstract

This development proves upper and lower bounds for several familiar real-valued functions. For sin, cos, exp and the square root function, it defines and verifies infinite families of upper and lower bounds, mostly based on Taylor series expansions. For $\tan^{-1}$, ln and exp, it verifies a finite collection of upper and lower bounds, originally obtained from the functions' continued fraction expansions using the computer algebra system Maple. A common theme in these proofs is to take the difference between a function and its approximation, which should be zero at one point, and then consider the sign of the derivative.

The immediate purpose of this development is to verify axioms used by MetiTarski [1], an automatic theorem prover for real-valued special functions. Crucial to MetiTarski's operation is the provision of upper and lower bounds for each function of interest.

Contents

Chapter 1

General Lemmas for Proving Function Inequalities

theory *Bounds-Lemmas*
imports *Complex-Main*

begin

These are for functions that are differentiable over a closed interval.

lemma *gen-lower-bound-increasing*:
  fixes a :: *real*
  assumes $a \leq x$
    and $\bigwedge y.\ a \leq y \Longrightarrow y \leq x \Longrightarrow ((\lambda x.\ fl\ x - f\ x)$ *has-real-derivative* $g\ y)\ (at\ y)$
    and $\bigwedge y.\ a \leq y \Longrightarrow y \leq x \Longrightarrow g\ y \leq 0$
    and $fl\ a = f\ a$
  shows $fl\ x \leq f\ x$
proof $-$
  have $fl\ x - f\ x \leq fl\ a - f\ a$
    apply (*rule DERIV-nonpos-imp-nonincreasing* [**where** $f = \lambda x.\ fl\ x - f\ x$])
    apply (*rule assms*)
    apply (*intro allI impI exI conjI*)
    apply (*rule assms* | *simp*)+
    done
  also have ... = *0*
    by (*simp add*: *assms*)
  finally show *?thesis*
    by *simp*
qed

lemma *gen-lower-bound-decreasing*:
  fixes a :: *real*
  assumes $x \leq a$
    and $\bigwedge y.\ x \leq y \Longrightarrow y \leq a \Longrightarrow ((\lambda x.\ fl\ x - f\ x)$ *has-real-derivative* $g\ y)\ (at\ y)$
    and $\bigwedge y.\ x \leq y \Longrightarrow y \leq a \Longrightarrow g\ y \geq 0$
    and $fl\ a = f\ a$

shows $fl\ x \leq f\ x$
proof $-$
have $fl\ (-\ (-x)) \leq f\ (-\ (-x))$
apply (*rule gen-lower-bound-increasing* [*of* $-a$ $-x$ - - $\lambda u.\ -\ g\ (-u)$])
apply (*auto simp*: *assms*)
apply (*subst DERIV-mirror* [*symmetric*])
apply (*simp add*: *assms*)
done
then show *?thesis*
by *simp*
qed

lemma *gen-upper-bound-increasing*:
fixes a :: *real*
assumes $a \leq x$
and $\bigwedge y.\ a \leq y \Longrightarrow y \leq x \Longrightarrow ((\lambda x.\ fu\ x - f\ x)$ *has-real-derivative* $g\ y)$ (*at* y)
and $\bigwedge y.\ a \leq y \Longrightarrow y \leq x \Longrightarrow g\ y \geq 0$
and $fu\ a = f\ a$
shows $f\ x \leq fu\ x$
apply (*rule gen-lower-bound-increasing* [*of* a x f fu $\lambda u.\ -\ g\ u$])
using *assms DERIV-minus* [**where** $f = \lambda x.\ fu\ x - f\ x$]
apply *auto*
done

lemma *gen-upper-bound-decreasing*:
fixes a :: *real*
assumes $x \leq a$
and $\bigwedge y.\ x \leq y \Longrightarrow y \leq a \Longrightarrow ((\lambda x.\ fu\ x - f\ x)$ *has-real-derivative* $g\ y)$ (*at* y)
and $\bigwedge y.\ x \leq y \Longrightarrow y \leq a \Longrightarrow g\ y \leq 0$
and $fu\ a = f\ a$
shows $f\ x \leq fu\ x$
apply (*rule gen-lower-bound-decreasing* [*of* x a - - $\lambda u.\ -\ g\ u$])
using *assms DERIV-minus* [**where** $f = \lambda x.\ fu\ x - f\ x$]
apply *auto*
done

end

Chapter 2

Arctan Upper and Lower Bounds

theory *Atan-CF-Bounds*
imports *Bounds-Lemmas*
HOL−Library.Sum-of-Squares

begin

Covers all bounds used in arctan-upper.ax, arctan-lower.ax and arctan-extended.ax, excepting only arctan-extended2.ax, which is used in two atan-error-analysis problems.

2.1 Upper Bound 1

definition *arctan-upper-11* :: *real* ⇒ *real*
where *arctan-upper-11* ≡ $\lambda x.\ -(pi/2) - 1/x$

definition *diff-delta-arctan-upper-11* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-11* ≡ $\lambda x.\ 1\ /\ (x\hat{\ }2 * (1 + x\hat{\ }2))$

lemma *d-delta-arctan-upper-11*: $x \neq 0 \Longrightarrow$
((λx. *arctan-upper-11 x* − *arctan x*) *has-field-derivative diff-delta-arctan-upper-11 x*) (*at x*)
unfolding *arctan-upper-11-def diff-delta-arctan-upper-11-def*
apply (*intro derivative-eq-intros* | *simp*)+
apply (*simp add*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
done

lemma *d-delta-arctan-upper-11-pos*: $x \neq 0 \Longrightarrow$ *diff-delta-arctan-upper-11 x* > 0
unfolding *diff-delta-arctan-upper-11-def*
by (*simp add*: *divide-simps zero-less-mult-iff add-pos-pos*)

Different proof needed here: they coincide not at zero, but at (-) infinity!

lemma *arctan-upper-11*:

assumes *x* < *0*
shows *arctan*(*x*) < *arctan-upper-11 x*
proof −
have ((λ*x*. *arctan-upper-11 x* − *arctan x*) ⟶ − (*pi* / *2*) − *0* − (− (*pi* / *2*))) *at-bot*
unfolding *arctan-upper-11-def*
apply (*intro tendsto-intros tendsto-arctan-at-bot*, *auto simp*: *ext* [*OF divide-inverse*])
apply (*metis tendsto-inverse-0 at-bot-le-at-infinity tendsto-mono*)
done
then have ∗: ((λ*x*. *arctan-upper-11 x* − *arctan x*) ⟶ *0*) *at-bot*
by *simp*
have *0* < *arctan-upper-11 x* − *arctan x*
apply (*rule DERIV-pos-imp-increasing-at-bot* [*OF* - ∗])
apply (*metis assms d-delta-arctan-upper-11 d-delta-arctan-upper-11-pos not-le*)
done
then show *?thesis*
by *auto*
qed

definition *arctan-upper-12* :: *real* ⇒ *real*
where *arctan-upper-12* ≡ λ*x*. *3*∗*x* / (*x*^*2* + *3*)

definition *diff-delta-arctan-upper-12* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-12* ≡ λ*x*. −*4*∗*x*^*4* / ((*x*^*2*+*3*)^*2* ∗ (*1*+*x*^*2*))

lemma *d-delta-arctan-upper-12*:
((λ*x*. *arctan-upper-12 x* − *arctan x*) *has-field-derivative diff-delta-arctan-upper-12 x*) (*at x*)
unfolding *arctan-upper-12-def diff-delta-arctan-upper-12-def*
apply (*intro derivative-eq-intros*, *simp-all*)
apply (*auto simp*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
done

Strict inequalities also possible

lemma *arctan-upper-12*:
assumes *x* ≤ *0* **shows** *arctan*(*x*) ≤ *arctan-upper-12 x*
apply (*rule gen-upper-bound-decreasing* [*OF assms d-delta-arctan-upper-12*])
apply (*auto simp*: *diff-delta-arctan-upper-12-def arctan-upper-12-def*)
done

definition *arctan-upper-13* :: *real* ⇒ *real*
where *arctan-upper-13* ≡ λ*x*. *x*

definition *diff-delta-arctan-upper-13* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-13* ≡ λ*x*. *x*^*2* / (*1* + *x*^*2*)

lemma *d-delta-arctan-upper-13*:
((λ*x*. *arctan-upper-13 x* − *arctan x*) *has-field-derivative diff-delta-arctan-upper-13 x*) (*at x*)

unfolding *arctan-upper-13-def diff-delta-arctan-upper-13-def*
apply (*intro derivative-eq-intros*, *simp-all*)
apply (*simp add*: *divide-simps add-nonneg-eq-0-iff*)
done

lemma *arctan-upper-13*:
  assumes $x \geq 0$ **shows** $arctan(x) \leq arctan\text{-}upper\text{-}13\ x$
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-arctan-upper-13*])
apply (*auto simp*: *diff-delta-arctan-upper-13-def arctan-upper-13-def*)
done

definition *arctan-upper-14* :: *real* ⇒ *real*
  where *arctan-upper-14* ≡ λ*x*. *pi*/*2* − *3*∗*x* / (*1* + *3*∗*x*^*2*)

definition *diff-delta-arctan-upper-14* :: *real* ⇒ *real*
  where *diff-delta-arctan-upper-14* ≡ λ*x*. −*4* / ((*1* + *3*∗*x*^*2*)^*2* ∗ (*1*+*x*^*2*))

lemma *d-delta-arctan-upper-14*:
  ((λ*x*. *arctan-upper-14 x* − *arctan x*) *has-field-derivative diff-delta-arctan-upper-14 x*) (*at x*)
unfolding *arctan-upper-14-def diff-delta-arctan-upper-14-def*
apply (*intro derivative-eq-intros* | *simp add*: *add-nonneg-eq-0-iff*)+
apply (*simp add*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
done

lemma *d-delta-arctan-upper-14-neg*: *diff-delta-arctan-upper-14 x* < *0*
unfolding *diff-delta-arctan-upper-14-def*
apply (*auto simp*: *divide-simps add-nonneg-eq-0-iff zero-less-mult-iff*)
using *power2-less-0* [*of x*]
apply *arith*
done

lemma *lim14*: ((λ*x*::*real*. $3 * x / (1 + 3 * x^2)$) ⟶ *0*) *at-infinity*
  apply (*rule tendsto-0-le* [**where** *f* = *inverse* **and** *K*=*1*])
  apply (*metis tendsto-inverse-0*)
  apply (*simp add*: *eventually-at-infinity*)
  apply (*rule-tac x*=*1* **in** *exI*)
  apply (*simp add*: *power-eq-if abs-if divide-simps add-sign-intros*)
  done

Different proof needed here: they coincide not at zero, but at (+) infinity!

lemma *arctan-upper-14*:
  assumes $x > 0$
    shows $arctan(x) < arctan\text{-}upper\text{-}14\ x$
proof −
  have ((λ*x*. *arctan-upper-14 x* − *arctan x*) ⟶ *pi* / *2* − *0* − *pi* / *2*) *at-top*
    unfolding *arctan-upper-14-def*
    apply (*intro tendsto-intros tendsto-arctan-at-top*)
    apply (*auto simp*: *tendsto-mono* [*OF at-top-le-at-infinity lim14*])

done
then have ∗: $((\lambda x.\ \mathit{arctan\text{-}upper\text{-}14}\ x - \mathit{arctan}\ x) \longrightarrow 0)$ *at-top*
by *simp*
have $0 < \mathit{arctan\text{-}upper\text{-}14}\ x - \mathit{arctan}\ x$
apply (*rule DERIV-neg-imp-decreasing-at-top* [*OF* - ∗])
apply (*metis d-delta-arctan-upper-14 d-delta-arctan-upper-14-neg*)
done
then show *?thesis*
by *auto*
qed

2.2 Lower Bound 1

definition *arctan-lower-11* :: *real* ⇒ *real*
where *arctan-lower-11* ≡ $\lambda x.\ -(pi/2) - 3*x\ /\ (1 + 3*x\hat{}2)$

lemma *arctan-lower-11*:
assumes $x < 0$
shows $\mathit{arctan}(x) > \mathit{arctan\text{-}lower\text{-}11}\ x$
using *arctan-upper-14* [*of* $-x$] *assms*
by (*auto simp*: *arctan-upper-14-def arctan-lower-11-def arctan-minus*)

abbreviation *arctan-lower-12* ≡ *arctan-upper-13*

lemma *arctan-lower-12*:
assumes $x \le 0$
shows $\mathit{arctan}(x) \ge \mathit{arctan\text{-}lower\text{-}12}\ x$
using *arctan-upper-13* [*of* $-x$] *assms*
by (*auto simp*: *arctan-upper-13-def arctan-minus*)

abbreviation *arctan-lower-13* ≡ *arctan-upper-12*

lemma *arctan-lower-13*:
assumes $x \ge 0$
shows $\mathit{arctan}(x) \ge \mathit{arctan\text{-}lower\text{-}13}\ x$
using *arctan-upper-12* [*of* $-x$] *assms*
by (*auto simp*: *arctan-upper-12-def arctan-minus*)

definition *arctan-lower-14* :: *real* ⇒ *real*
where *arctan-lower-14* ≡ $\lambda x.\ pi/2 - 1/x$

lemma *arctan-lower-14*:
assumes $x > 0$
shows $\mathit{arctan}(x) > \mathit{arctan\text{-}lower\text{-}14}\ x$
using *arctan-upper-11* [*of* $-x$] *assms*
by (*auto simp*: *arctan-upper-11-def arctan-lower-14-def arctan-minus*)

2.3 Upper Bound 3

definition *arctan-upper-31* :: *real* ⇒ *real*
 where *arctan-upper-31* ≡ λ*x*. −(*pi*/*2*) − (*64* + *735*∗*x*^*2* + *945*∗*x*^*4*) / (*15*∗*x*∗(*15* + *70*∗*x*^*2* + *63*∗*x*^*4*))

definition *diff-delta-arctan-upper-31* :: *real* ⇒ *real*
 where *diff-delta-arctan-upper-31* ≡ λ*x*. *64* / (*x*^*2* ∗ (*15* + *70*∗*x*^*2* + *63*∗*x*^*4*)^*2* ∗ (*1* + *x*^*2*))

lemma *d-delta-arctan-upper-31*:
 assumes *x* ≠ *0*
 shows ((λ*x*. *arctan-upper-31 x* − *arctan x*) *has-field-derivative diff-delta-arctan-upper-31 x*) (*at x*)
 unfolding *arctan-upper-31-def diff-delta-arctan-upper-31-def*
 using *assms*
 apply (*intro derivative-eq-intros*)
 apply (*rule refl* | *simp add*: *add-nonneg-eq-0-iff*)+
 apply (*simp add*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
 done

lemma *d-delta-arctan-upper-31-pos*: *x* ≠ *0* ⟹ *diff-delta-arctan-upper-31 x* > *0*
unfolding *diff-delta-arctan-upper-31-def*
by (*auto simp*: *divide-simps zero-less-mult-iff add-pos-pos add-nonneg-eq-0-iff*)

lemma *arctan-upper-31*:
 assumes *x* < *0*
 shows *arctan*(*x*) < *arctan-upper-31 x*
proof −
 have ∗: ⋀*x*::*real*. (*15* + *70* ∗ x^2 + *63* ∗ *x* ^ *4*) > *0*
 by (*sos* ((*R*<*1* + ((*R*<*1* ∗ ((*R*<*7*/*8* ∗ [*19*/*7*∗*x*^*2* + *1*]^*2*) + ((*R*<*4* ∗ [*x*]^*2*) + (*R*<*10*/*7* ∗ [*x*^*2*]^*2*)))) + ((*A*<=*0* ∗ *R*<*1*) ∗ (*R*<*1*/*8* ∗ [*1*]^*2*))))))
 then have ∗∗: ⋀*x*::*real*. ¬ (*15* + *70* ∗ x^2 + *63* ∗ *x* ^ *4*) < *0*
 by (*simp add*: *not-less*)
 have ((λ*x*::*real*. (*64* + *735* ∗ x^2 + *945* ∗ *x* ^ *4*) / (*15* ∗ *x* ∗ (*15* + *70* ∗ x^2 + *63* ∗ *x* ^ *4*))) ⟶ *0*) *at-bot*
 apply (*rule tendsto-0-le* [**where** *f* = *inverse* **and** *K*=*2*])
 apply (*metis at-bot-le-at-infinity tendsto-inverse-0 tendsto-mono*)
 apply (*simp add*: *eventually-at-bot-linorder*)
 apply (*rule-tac x*=−*1* **in** *exI*)
 apply (*auto simp*: *divide-simps abs-if zero-less-mult-iff* ∗∗)
 done
 then have ((λ*x*. *arctan-upper-31 x* − *arctan x*) ⟶ − (*pi* / *2*) − *0* − (− (*pi* / *2*))) *at-bot*
 unfolding *arctan-upper-31-def*
 apply (*intro tendsto-intros tendsto-arctan-at-bot*, *auto*)
 done
 then have ∗: ((λ*x*. *arctan-upper-31 x* − *arctan x*) ⟶ *0*) *at-bot*
 by *simp*

have *0* < *arctan-upper-31 x* − *arctan x*
apply (*rule DERIV-pos-imp-increasing-at-bot* [*OF* - ∗])
apply (*metis assms d-delta-arctan-upper-31 d-delta-arctan-upper-31-pos not-le*)
done
then show *?thesis*
by *auto*
qed

definition *arctan-upper-32* :: *real* ⇒ *real*
where *arctan-upper-32* ≡ λ*x*. *7*∗(*33*∗*x*^*4* + *170*∗*x*^*2* + *165*)∗*x* / (*5*∗(*5*∗*x*^*6* + *105*∗*x*^*4* + *315*∗*x*^*2* + *231*))

definition *diff-delta-arctan-upper-32* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-32* ≡ λ*x*. −*256*∗*x*^*12* / ((*5*∗*x*^*6*+*105*∗*x*^*4*+*315*∗*x*^*2*+*231*)^*2*∗(*1*+*x*^*2*))

lemma *d-delta-arctan-upper-32*:
((λ*x*. *arctan-upper-32 x* − *arctan x*) *has-field-derivative diff-delta-arctan-upper-32 x*) (*at x*)
unfolding *arctan-upper-32-def diff-delta-arctan-upper-32-def*
apply (*intro derivative-eq-intros* | *simp*)+
apply *simp-all*
apply (*auto simp*: *add-nonneg-eq-0-iff divide-simps*, *algebra*)
done

lemma *arctan-upper-32*:
assumes *x* ≤ *0* **shows** *arctan*(*x*) ≤ *arctan-upper-32 x*
apply (*rule gen-upper-bound-decreasing* [*OF assms d-delta-arctan-upper-32*])
apply (*auto simp*: *diff-delta-arctan-upper-32-def arctan-upper-32-def*)
done

definition *arctan-upper-33* :: *real* ⇒ *real*
where *arctan-upper-33* ≡ λ*x*. (*64*∗*x*^*4*+*735*∗*x*^*2*+*945*)∗*x* / (*15*∗(*15*∗*x*^*4*+*70*∗*x*^*2*+*63*))

definition *diff-delta-arctan-upper-33* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-33* ≡ λ*x*. *64*∗*x*^*10* / ((*15*∗*x*^*4*+*70*∗*x*^*2*+*63*)^*2*∗(*1*+*x*^*2*))

lemma *d-delta-arctan-upper-33*:
((λ*x*. *arctan-upper-33 x* − *arctan x*) *has-field-derivative diff-delta-arctan-upper-33 x*) (*at x*)
unfolding *arctan-upper-33-def diff-delta-arctan-upper-33-def*
apply (*intro derivative-eq-intros*, *simp-all*)
apply (*auto simp*: *add-nonneg-eq-0-iff divide-simps*, *algebra*)
done

lemma *arctan-upper-33*:
assumes *x* ≥ *0* **shows** *arctan*(*x*) ≤ *arctan-upper-33 x*
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-arctan-upper-33*])
apply (*auto simp*: *diff-delta-arctan-upper-33-def arctan-upper-33-def*)
done

```
definition arctan-upper-34 :: real ⇒ real
  where arctan-upper-34 ≡
      λx. pi/2 − (33 + 170∗x^2 + 165∗x^4)∗7∗x / (5∗(5 + 105∗x^2 + 315∗x^4
+ 231∗x^6))

definition diff-delta-arctan-upper-34 :: real ⇒ real
 where diff-delta-arctan-upper-34 ≡ λx. −256 / ((5+105∗x^2+315∗x^4+231∗x^6)^2∗(1+x^2))

lemma d-delta-arctan-upper-34:
 ((λx. arctan-upper-34 x − arctan x) has-field-derivative diff-delta-arctan-upper-34
x) (at x)
unfolding arctan-upper-34-def diff-delta-arctan-upper-34-def
apply (intro derivative-eq-intros | simp add: add-nonneg-eq-0-iff)+
apply (simp add: divide-simps add-nonneg-eq-0-iff, algebra)
done

lemma d-delta-arctan-upper-34-pos: diff-delta-arctan-upper-34 x < 0
unfolding diff-delta-arctan-upper-34-def
apply (simp add: divide-simps add-nonneg-eq-0-iff zero-less-mult-iff)
using power2-less-0 [of x]
apply arith
done

lemma arctan-upper-34:
  assumes x > 0
    shows arctan(x) < arctan-upper-34 x
proof −
  have ((λx. arctan-upper-34 x − arctan x) ⟶ pi / 2 − 0 − pi / 2) at-top
    unfolding arctan-upper-34-def
    apply (intro tendsto-intros tendsto-arctan-at-top, auto)
    apply (rule tendsto-0-le [where f = inverse and K=1])
    apply (metis tendsto-inverse-0 at-top-le-at-infinity tendsto-mono)
    apply (simp add: eventually-at-top-linorder)
    apply (rule-tac x=1 in exI)
    apply (auto simp: divide-simps power-eq-if add-pos-pos algebra-simps)
    done
  then have ∗: ((λx. arctan-upper-34 x − arctan x) ⟶ 0) at-top
    by simp
  have 0 < arctan-upper-34 x − arctan x
    apply (rule DERIV-neg-imp-decreasing-at-top [OF - ∗])
    apply (metis d-delta-arctan-upper-34 d-delta-arctan-upper-34-pos)
    done
  then show ?thesis
    by auto
qed
```

2.4 Lower Bound 3

definition *arctan-lower-31* :: *real* ⇒ *real*
where *arctan-lower-31* ≡ λ*x*. −(*pi*/*2*) − (*33* + *170*∗*x*^*2* + *165*∗*x*^*4*)∗*7*∗*x* / (*5*∗(*5* + *105*∗*x*^*2* + *315*∗*x*^*4* + *231*∗*x*^*6*))

lemma *arctan-lower-31*:
assumes $x < 0$
shows *arctan*(*x*) > *arctan-lower-31 x*
using *arctan-upper-34* [*of* −*x*] *assms*
by (*auto simp*: *arctan-upper-34-def arctan-lower-31-def arctan-minus*)

abbreviation *arctan-lower-32* ≡ *arctan-upper-33*

lemma *arctan-lower-32*:
assumes $x \leq 0$
shows *arctan*(*x*) ≥ *arctan-lower-32 x*
using *arctan-upper-33* [*of* −*x*] *assms*
by (*auto simp*: *arctan-upper-33-def arctan-minus*)

abbreviation *arctan-lower-33* ≡ *arctan-upper-32*

lemma *arctan-lower-33*:
assumes $x \geq 0$
shows *arctan*(*x*) ≥ *arctan-lower-33 x*
using *arctan-upper-32* [*of* −*x*] *assms*
by (*auto simp*: *arctan-upper-32-def arctan-minus*)

definition *arctan-lower-34* :: *real* ⇒ *real*
where *arctan-lower-34* ≡ λ*x*. *pi*/*2* − (*64* + *735*∗*x*^*2* + *945*∗*x*^*4*) / (*15*∗*x*∗(*15* + *70*∗*x*^*2* + *63*∗*x*^*4*))

lemma *arctan-lower-34*:
assumes $x > 0$
shows *arctan*(*x*) > *arctan-lower-34 x*
using *arctan-upper-31* [*of* −*x*] *assms*
by (*auto simp*: *arctan-upper-31-def arctan-lower-34-def arctan-minus*)

2.5 Upper Bound 4

definition *arctan-upper-41* :: *real* ⇒ *real*
where *arctan-upper-41* ≡
λ*x*. −(*pi*/*2*) − (*256* + *5943*∗*x*^*2* + *19250*∗*x*^*4* + *15015*∗*x*^*6*) /
(*35*∗*x*∗(*35* + *315*∗*x*^*2* + *693*∗*x*^*4* + *429*∗*x*^*6*))

definition *diff-delta-arctan-upper-41* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-41* ≡ λ*x*. *256* / (*x*^*2*∗(*35*+*315*∗*x*^*2*+*693*∗*x*^*4*+*429*∗*x*^*6*)^*2*∗(*1*+*x*^*2*))

lemma *d-delta-arctan-upper-41*:

```
  assumes x ≠ 0
   shows ((λx. arctan-upper-41 x − arctan x) has-field-derivative diff-delta-arctan-upper-41
x) (at x)
  unfolding arctan-upper-41-def diff-delta-arctan-upper-41-def
  using assms
  apply (intro derivative-eq-intros)
  apply (rule refl | simp add: add-nonneg-eq-0-iff)+
  apply (simp add: divide-simps add-nonneg-eq-0-iff, algebra)
  done

lemma d-delta-arctan-upper-41-pos: x ≠ 0 ⟹ diff-delta-arctan-upper-41 x > 0
unfolding diff-delta-arctan-upper-41-def
by (auto simp: zero-less-mult-iff add-pos-pos add-nonneg-eq-0-iff)

lemma arctan-upper-41:
  assumes x < 0
    shows arctan(x) < arctan-upper-41 x
proof −
  have ∗: ⋀x::real. (35 + 315 ∗ x^2 + 693 ∗ x ^ 4 + 429 ∗ x ^ 6) > 0
     by (sos ((R<1 + ((R<1 ∗ ((R<13/8589934592 ∗ [95/26∗x^2 + 1]^2) +
((R<38654705675/4294967296 ∗ [170080704731/154618822700∗x^3 + x]^2) +
((R<14271/446676598784 ∗ [x^2]^2) + (R<3631584276674589067439/2656331147370089676800
∗ [x^3]^2))))) + ((A<=0 ∗ R<1) ∗ (R<245426703/8589934592 ∗ [1]^2))))))
  then have ∗∗: ⋀x::real. x < 0 ⟹ ¬ (35 + 315 ∗ x^2 + 693 ∗ x ^ 4 + 429 ∗ x
^ 6) < 0
    by (simp add: not-less)
  have ((λx::real. (256 + 5943 ∗ x^2 + 19250 ∗ x ^ 4 + 15015 ∗ x ^ 6) /
         (35 ∗ x ∗ (35 + 315 ∗ x^2 + 693 ∗ x ^ 4 + 429 ∗ x ^ 6))) ⟶ 0) at-bot
    apply (rule tendsto-0-le [where f = inverse and K=2])
    apply (metis at-bot-le-at-infinity tendsto-inverse-0 tendsto-mono)
    apply (simp add: eventually-at-bot-linorder)
    apply (rule-tac x=−1 in exI)
    apply (auto simp: ∗∗ abs-if divide-simps zero-less-mult-iff)
    done
  then have ((λx. arctan-upper-41 x − arctan x) ⟶ − (pi / 2) − 0 − (− (pi
/ 2))) at-bot
    unfolding arctan-upper-41-def
    apply (intro tendsto-intros tendsto-arctan-at-bot, auto)
    done
  then have ∗: ((λx. arctan-upper-41 x − arctan x) ⟶ 0) at-bot
    by simp
  have 0 < arctan-upper-41 x − arctan x
    apply (rule DERIV-pos-imp-increasing-at-bot [OF - ∗])
   apply (metis assms d-delta-arctan-upper-41 d-delta-arctan-upper-41-pos not-le)
    done
  then show ?thesis
    by auto
qed
```

definition *arctan-upper-42* :: *real* ⇒ *real*
where *arctan-upper-42* ≡
λx. (15159∗x^6+147455∗x^4+345345∗x^2+225225)∗x / (35∗(35∗x^8+1260∗x^6+6930∗x^4+12012∗

definition *diff-delta-arctan-upper-42* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-42* ≡
λx. −16384∗x^16 / ((35∗x^8+1260∗x^6+6930∗x^4+12012∗x^2+6435)^2∗(1+x^2))

lemma *d-delta-arctan-upper-42*:
((λx. arctan-upper-42 x − arctan x) has-field-derivative diff-delta-arctan-upper-42 x) (at x)
unfolding *arctan-upper-42-def diff-delta-arctan-upper-42-def*
apply (*intro derivative-eq-intros*, *simp-all*)
apply (*auto simp*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
done

lemma *arctan-upper-42*:
assumes *x* ≤ *0* **shows** *arctan*(*x*) ≤ *arctan-upper-42 x*
apply (*rule gen-upper-bound-decreasing* [*OF assms d-delta-arctan-upper-42*])
apply (*auto simp*: *diff-delta-arctan-upper-42-def arctan-upper-42-def*)
done

definition *arctan-upper-43* :: *real* ⇒ *real*
where *arctan-upper-43* ≡
λx. (256∗x^6+5943∗x^4+19250∗x^2+15015)∗x /
(35 ∗ (35∗x^6+315∗x^4+693∗x^2+429))

definition *diff-delta-arctan-upper-43* :: *real* ⇒ *real*
where *diff-delta-arctan-upper-43* ≡ λx. 256∗x^14 / ((35∗x^6+315∗x^4+693∗x^2+429)^2∗(1+x^2))

lemma *d-delta-arctan-upper-43*:
((λx. arctan-upper-43 x − arctan x) has-field-derivative diff-delta-arctan-upper-43 x) (at x)
unfolding *arctan-upper-43-def diff-delta-arctan-upper-43-def*
apply (*intro derivative-eq-intros*, *simp-all*)
apply (*auto simp*: *add-nonneg-eq-0-iff divide-simps*, *algebra*)
done

lemma *arctan-upper-43*:
assumes *x* ≥ *0* **shows** *arctan*(*x*) ≤ *arctan-upper-43 x*
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-arctan-upper-43*])
apply (*auto simp*: *diff-delta-arctan-upper-43-def arctan-upper-43-def*)
done

definition *arctan-upper-44* :: *real* ⇒ *real*
where *arctan-upper-44* ≡
λx. pi/2 − (15159+147455∗x^2+345345∗x^4+225225∗x^6)∗x /
(35∗(35+1260∗x^2+6930∗x^4+12012∗x^6+6435∗x^8))

definition *diff-delta-arctan-upper-44* :: *real* ⇒ *real*
  where *diff-delta-arctan-upper-44* ≡
   λx. −16384 / ((35+1260∗x^2+6930∗x^4+12012∗x^6+6435∗x^8)^2∗(1+x^2))

lemma *d-delta-arctan-upper-44*:
 ((*λx. arctan-upper-44 x − arctan x*) *has-field-derivative diff-delta-arctan-upper-44 x*) (*at x*)
unfolding *arctan-upper-44-def diff-delta-arctan-upper-44-def*
apply (*intro derivative-eq-intros* | *simp add*: *add-nonneg-eq-0-iff*)+
apply (*simp add*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
done

lemma *d-delta-arctan-upper-44-pos*: *diff-delta-arctan-upper-44 x < 0*
unfolding *diff-delta-arctan-upper-44-def*
apply (*auto simp*: *divide-simps add-nonneg-eq-0-iff zero-less-mult-iff*)
using *power2-less-0* [*of x*]
apply *arith*
done

lemma *arctan-upper-44*:
  assumes *x > 0*
    shows *arctan*(*x*) < *arctan-upper-44 x*
proof −
  have ((*λx. arctan-upper-44 x − arctan x*) ⟶ *pi / 2 − 0 − pi / 2*) *at-top*
    unfolding *arctan-upper-44-def*
    apply (*intro tendsto-intros tendsto-arctan-at-top*, *auto*)
    apply (*rule tendsto-0-le* [**where** *f = inverse* **and** *K=1*])
    apply (*metis tendsto-inverse-0 at-top-le-at-infinity tendsto-mono*)
    apply (*simp add*: *eventually-at-top-linorder*)
    apply (*rule-tac x=1* **in** *exI*)
     apply (*auto simp*: *zero-le-mult-iff divide-simps not-le*[*symmetric*] *power-eq-if algebra-simps*)
    done
  then have ∗: ((*λx. arctan-upper-44 x − arctan x*) ⟶ *0*) *at-top*
    by *simp*
  have *0 < arctan-upper-44 x − arctan x*
    apply (*rule DERIV-neg-imp-decreasing-at-top* [*OF* - ∗])
    apply (*metis d-delta-arctan-upper-44 d-delta-arctan-upper-44-pos*)
    done
  then show *?thesis*
    by *auto*
qed

2.6 Lower Bound 4

definition *arctan-lower-41* :: *real* ⇒ *real*
  where *arctan-lower-41* ≡
    λx. −(pi/2) − (15159+147455∗x^2+345345∗x^4+225225∗x^6)∗x /
        (35∗(35+1260∗x^2+6930∗x^4+12012∗x^6+6435∗x^8))

lemma *arctan-lower-41*:
  assumes $x < 0$
    shows $arctan(x) > arctan\text{-}lower\text{-}41\ x$
    using *arctan-upper-44* [*of* $-x$] *assms*
    by (*auto simp*: *arctan-upper-44-def arctan-lower-41-def arctan-minus*)

abbreviation *arctan-lower-42* ≡ *arctan-upper-43*

lemma *arctan-lower-42*:
  assumes $x \leq 0$
    shows $arctan(x) \geq arctan\text{-}lower\text{-}42\ x$
    using *arctan-upper-43* [*of* $-x$] *assms*
    by (*auto simp*: *arctan-upper-43-def arctan-minus*)

abbreviation *arctan-lower-43* ≡ *arctan-upper-42*

lemma *arctan-lower-43*:
  assumes $x \geq 0$
    shows $arctan(x) \geq arctan\text{-}lower\text{-}43\ x$
    using *arctan-upper-42* [*of* $-x$] *assms*
    by (*auto simp*: *arctan-upper-42-def arctan-minus*)

definition *arctan-lower-44* :: *real* ⇒ *real*
  where *arctan-lower-44* ≡
    $\lambda x.\ pi/2 - (256+5943*x\hat{}2+19250*x\hat{}4+15015*x\hat{}6) /$
        $(35*x*(35+315*x\hat{}2+693*x\hat{}4+429*x\hat{}6))$

lemma *arctan-lower-44*:
  assumes $x > 0$
    shows $arctan(x) > arctan\text{-}lower\text{-}44\ x$
    using *arctan-upper-41* [*of* $-x$] *assms*
    by (*auto simp*: *arctan-upper-41-def arctan-lower-44-def arctan-minus*)

end

Chapter 3

Exp Upper and Lower Bounds

theory *Exp-Bounds*
imports *Bounds-Lemmas*
HOL−Library.Sum-of-Squares
Sturm-Sequences.Sturm

begin

Covers all bounds used in exp-upper.ax, exp-lower.ax and exp-extended.ax.

3.1 Taylor Series Bounds

exp-positive is the theorem $0 \leq exp\ ?x$

exp-lower-taylor-1 is the theorem $1 + ?x \leq exp\ ?x$

All even approximants are lower bounds.

lemma *exp-lower-taylor-even*:
fixes *x::real*
shows $even\ n \Longrightarrow (\sum m{<}n.\ (x \ \hat{}\ m)\ /\ (fact\ m)) \leq exp\ x$
using *Maclaurin-exp-le* [*of x n*]
by (*auto simp add*: *zero-le-even-power*)

lemma *exp-upper-taylor-even*:
fixes *x::real*
assumes *n*: *even n*
and *pos*: $(\sum m{<}n.\ ((-x)\ \hat{}\ m)\ /\ (fact\ m)) > 0$ (**is** $?sum > 0$)
shows $exp\ x \leq inverse\ ?sum$
using *exp-lower-taylor-even* [*OF n*, *of* $-x$]
by (*metis exp-minus inverse-inverse-eq le-imp-inverse-le pos*)

3 if the previous lemma is expressed in terms of $(2::'a) * m$.

lemma *exp-lower-taylor-3*:

```
  fixes x::real
  shows 1 + x + (1/2)*x^2 + (1/6)*x^3 + (1/24)*x^4 + (1/120)*x^5 ≤ exp
x
  by (rule order-trans [OF - exp-lower-taylor-even [of 6]])
     (auto simp: lessThan-nat-numeral fact-numeral)

lemma exp-lower-taylor-3-cubed:
  fixes x::real
   shows (1 + x/3 + (1/2)*(x/3)^2 + (1/6)*(x/3)^3 + (1/24)*(x/3)^4 +
(1/120)*(x/3)^5)^3 ≤ exp x
proof −
   have (1 + x/3 + (1/2)*(x/3)^2 + (1/6)*(x/3)^3 + (1/24)*(x/3)^4 +
(1/120)*(x/3)^5) ^ 3
        ≤ exp (x/3) ^ 3
    by (metis power-mono-odd odd-numeral exp-lower-taylor-3)
 also have ... = exp x
   by (simp add: exp-of-nat-mult [symmetric])
 finally show ?thesis .
qed

lemma exp-lower-taylor-2:
  fixes x::real
  shows 1 + x + (1/2)*x^2 + (1/6)*x^3 ≤ exp x
proof −
  have even (4::nat) by simp
  then have (∑m<4. x ^ m / (fact m)) ≤ exp x
    by (rule exp-lower-taylor-even)
  then show ?thesis by (auto simp add: numeral-eq-Suc)
qed

lemma exp-upper-bound-case-3:
  fixes x::real
  assumes x ≤ 3.19
  shows exp x ≤ 2304 / (−(x^3) + 6*x^2 − 24*x + 48)^2
proof −
 have (1/48)*(−(x^3) + 6*x^2 − 24*x + 48) = (1 + (−x/2) + (1/2)*(−x/2)^2
+ (1/6)*(−x/2)^3)
    by (simp add: field-simps power2-eq-square power3-eq-cube)
  also have ... ≤ exp (−x/2)
    by (rule exp-lower-taylor-2)
  finally have 1: (1/48)*(−(x^3) + 6*x^2 − 24*x + 48) ≤ exp (−x/2) .
  have (−(x^3) + 6*x^2 − 24*x + 48)^2 / 2304 = ((1/48)*(−(x^3) + 6*x^2
− 24*x + 48))^2
    by (simp add: field-simps power2-eq-square power3-eq-cube)
  also have ... ≤ (exp (−x/2))^2
    apply (rule power-mono [OF 1])
    apply (simp add: algebra-simps)
    using assms
   apply (sos ((R<1 + ((R<1 * ((R<1323/13 * [~15/49*x + 1]^2) + (R<1/637
```

```
* [x]^2))) + (((A<0 * R<1) * (R<50/13 * [1]^2)) + ((A<=0 * R<1) * ((R<56/13
* [~5/56*x + 1]^2) + (R<199/728 * [x]^2))))))))
    done
  also have ... = inverse (exp x)
    by (metis exp-minus mult-exp-exp power2-eq-square field-sum-of-halves)
  finally have 2: (−(x^3) + 6*x^2 − 24*x + 48)^2 / 2304 ≤ inverse (exp x) .
  have 6 * x² − x ^ 3 − 24 * x + 48 ≠ 0 using assms
    by (sos ((R<1 + (([~400/13] * A=0) + ((R<1 * ((R<1323/13 * [~15/49*x
+ 1]^2) + (R<1/637 * [x]^2))) + ((A<=0 * R<1) * ((R<56/13 * [~5/56*x
+ 1]^2) + (R<199/728 * [x]^2))))))))
  then show ?thesis
    using Fields.linordered-field-class.le-imp-inverse-le [OF 2]
    by simp
qed

lemma exp-upper-bound-case-5:
  fixes x::real
  assumes x ≤ 6.36
  shows exp x ≤ 21743271936 / (−(x^3) + 12*x^2 − 96*x + 384)^4
proof −
 have (1/384)*(−(x^3) + 12*x^2 − 96*x + 384) = (1 + (−x/4) + (1/2)*(−x/4)^2
+ (1/6)*(−x/4)^3)
    by (simp add: field-simps power2-eq-square power3-eq-cube)
  also have ... ≤ exp (−x/4)
    by (rule exp-lower-taylor-2)
  finally have 1: (1/384)*(−(x^3) + 12*x^2 − 96*x + 384) ≤ exp (−x/4) .
 have (−(x^3) + 12*x^2 − 96*x + 384)^4 / 21743271936 = ((1/384)*(−(x^3)
+ 12*x^2 − 96*x + 384))^4
    by (simp add: divide-simps)
  also have ... ≤ (exp (−x/4))^4
    apply (rule power-mono [OF 1])
    apply (simp add: algebra-simps)
    using assms
     apply (sos ((R<1 + ((R<1 * ((R<1777/32 * [~539/3554*x + 1]^2) +
(R<907/227456 * [x]^2))) + (((A<0 * R<1) * (R<25/1024 * [1]^2)) + ((A<=0
* R<1) * ((R<49/32 * [~2/49*x + 1]^2) + (R<45/1568 * [x]^2))))))))
    done
  also have ... = inverse (exp x)
    by (simp add: exp-of-nat-mult [symmetric] exp-minus [symmetric])
  finally have 2: (−(x^3) + 12*x^2 − 96*x + 384)^4 / 21743271936 ≤ inverse
(exp x) .
  have 12 * x² − x ^ 3 − 96 * x + 384 ≠ 0 using assms
   by (sos ((R<1 + (([~25/32] * A=0) + ((R<1 * ((R<1777/32 * [~539/3554*x
+ 1]^2) + (R<907/227456 * [x]^2))) + ((A<=0 * R<1) * ((R<49/32 * [~2/49*x
+ 1]^2) + (R<45/1568 * [x]^2))))))))
  then show ?thesis
    using Fields.linordered-field-class.le-imp-inverse-le [OF 2]
    by simp
qed
```

3.2 Continued Fraction Bound 2

```
definition exp-cf2 :: real ⇒ real
  where exp-cf2 ≡ λx. (x^2 + 6*x + 12) / (x^2 − 6*x + 12)

lemma denom-cf2-pos: fixes x::real shows x^2 − 6 * x + 12 > 0
  by (sos ((R<1 + ((R<1 * ((R<5 * [~3/10*x + 1]^2) + (R<1/20 * [x]^2)))
+ ((A<=0 * R<1) * (R<1/2 * [1]^2))))))

lemma numer-cf2-pos: fixes x::real shows x^2 + 6 * x + 12 > 0
  by (sos ((R<1 + ((R<1 * ((R<5 * [3/10*x + 1]^2) + (R<1/20 * [x]^2))) +
((A<=0 * R<1) * (R<1/2 * [1]^2))))))

lemma exp-cf2-pos: exp-cf2 x > 0
  unfolding exp-cf2-def
  by (auto simp add: divide-simps denom-cf2-pos numer-cf2-pos)

definition diff-delta-lnexp-cf2 :: real ⇒ real
  where diff-delta-lnexp-cf2 ≡ λx. − (x^4) / ((x^2 − 6*x + 12) * (x^2 + 6*x
+ 12))

lemma d-delta-lnexp-cf2-nonpos: diff-delta-lnexp-cf2 x ≤ 0
unfolding diff-delta-lnexp-cf2-def
by (sos (((R<1 + ((R<1 * ((R<5/4 * [~3/40*x^2 + 1]^2) + (R<11/1280 *
[x^2]^2))) +
      ((A<1 * R<1) * (R<1/64 * [1]^2))))) & ((R<1 + ((R<1 * ((R<5/4 *
[~3/40*x^2 + 1]^2) + (R<11/1280 * [x^2]^2))) + ((A<1 * R<1) * (R<1/64
* [1]^2)))))))

lemma d-delta-lnexp-cf2:
    ((λx. ln (exp-cf2 x) − x) has-field-derivative diff-delta-lnexp-cf2 x) (at x)
  unfolding exp-cf2-def diff-delta-lnexp-cf2-def
  apply (intro derivative-eq-intros | simp)+
  apply (metis exp-cf2-def exp-cf2-pos)
  apply (simp-all add: )
  using denom-cf2-pos [of x] numer-cf2-pos [of x]
  apply (auto simp: divide-simps)
  apply algebra
  done
```

Upper bound for non-positive x

```
lemma ln-exp-cf2-upper-bound-neg:
  assumes x ≤ 0
    shows x ≤ ln (exp-cf2 x)
 by (rule gen-upper-bound-decreasing [OF assms d-delta-lnexp-cf2 d-delta-lnexp-cf2-nonpos])
    (simp add: exp-cf2-def)

theorem exp-cf2-upper-bound-neg: x≤0 ⟹ exp(x) ≤ exp-cf2 x
  by (metis ln-exp-cf2-upper-bound-neg exp-cf2-pos exp-le-cancel-iff exp-ln-iff)
```

Lower bound for non-negative x

lemma *ln-exp-cf2-lower-bound-pos*:
assumes *0≤x*
shows *ln* (*exp-cf2 x*) ≤ *x*
by (*rule gen-lower-bound-increasing* [*OF assms d-delta-lnexp-cf2 d-delta-lnexp-cf2-nonpos*])
(*simp add*: *exp-cf2-def*)

theorem *exp-cf2-lower-bound-pos*: *0≤x* ⟹ *exp-cf2 x* ≤ *exp x*
by (*metis exp-cf2-pos exp-le-cancel-iff exp-ln ln-exp-cf2-lower-bound-pos*)

3.3 Continued Fraction Bound 3

This bound crosses the X-axis twice, causing complications.

definition *numer-cf3* :: *real* ⇒ *real*
where *numer-cf3* ≡ λ*x*. *x^3* + *12∗x^2* + *60∗x* + *120*

definition *exp-cf3* :: *real* ⇒ *real*
where *exp-cf3* ≡ λ*x*. *numer-cf3 x* / *numer-cf3* (−*x*)

lemma *numer-cf3-pos*: −*4.64* ≤ *x* ⟹ *numer-cf3 x* > *0*
unfolding *numer-cf3-def*
by *sturm*

lemma *exp-cf3-pos*: *numer-cf3 x* > *0* ⟹ *numer-cf3* (−*x*) > *0* ⟹ *exp-cf3 x* > *0*
by (*simp add*: *exp-cf3-def*)

definition *diff-delta-lnexp-cf3* :: *real* ⇒ *real*
where *diff-delta-lnexp-cf3* ≡ λ*x*. (*x^6*) / (*numer-cf3* (−*x*) ∗ *numer-cf3 x*)

lemma *d-delta-lnexp-cf3-nonneg*: *numer-cf3 x* > *0* ⟹ *numer-cf3* (−*x*) > *0* ⟹ *diff-delta-lnexp-cf3 x* ≥ *0*
unfolding *diff-delta-lnexp-cf3-def*
by (*auto simp*: *mult-less-0-iff intro*: *divide-nonneg-neg*)

lemma *d-delta-lnexp-cf3*:
assumes *numer-cf3 x* > *0 numer-cf3* (−*x*) > *0*
shows ((λ*x*. *ln* (*exp-cf3 x*) − *x*) *has-field-derivative diff-delta-lnexp-cf3 x*) (*at x*)
unfolding *exp-cf3-def numer-cf3-def diff-delta-lnexp-cf3-def*
apply (*intro derivative-eq-intros* | *simp*)+
using *assms numer-cf3-pos* [*of x*] *numer-cf3-pos* [*of* −*x*]
apply (*auto simp*: *numer-cf3-def*)
apply (*auto simp add*: *divide-simps add-nonneg-eq-0-iff*)
apply *algebra*
done

lemma *numer-cf3-mono*: *y* ≤ *x* ⟹ *numer-cf3 y* ≤ *numer-cf3 x*
unfolding *numer-cf3-def*

by (*sos* (((*A<0* ∗ *R<1*) + ((*A<=0* ∗ *R<1*) ∗ ((*R<60* ∗ [*1/10*∗*x* + *1/10*∗*y* + *1*]^*2*) +
((*R<2/5* ∗ [*x* + ~*1/4*∗*y*]^*2*) + (*R<3/8* ∗ [*y*]^*2*)))))))

Upper bound for non-negative x

lemma *ln-exp-cf3-upper-bound-nonneg*:
assumes *x0*: *0* ≤ *x* **and** *xless*: *numer-cf3* (−*x*) > *0*
shows *x* ≤ *ln* (*exp-cf3 x*)
proof −
have *ncf3*: ⋀*y*. *0* ≤ *y* ⟹ *y* ≤ *x* ⟹ *numer-cf3* (−*y*) > *0*
by (*metis neg-le-iff-le numer-cf3-mono order.strict-trans2 xless*)
show *?thesis*
apply (*rule gen-upper-bound-increasing* [*OF x0 d-delta-lnexp-cf3 d-delta-lnexp-cf3-nonneg*])
apply (*auto simp add*: *ncf3 assms numer-cf3-pos*)
apply (*simp add*: *exp-cf3-def numer-cf3-def*)
done
qed

theorem *exp-cf3-upper-bound-pos*: *0* ≤ *x* ⟹ *numer-cf3* (−*x*) > *0* ⟹ *exp x* ≤ *exp-cf3 x*
using *ln-exp-cf3-upper-bound-nonneg* [*of x*] *exp-cf3-pos* [*of x*] *numer-cf3-pos* [*of x*]
by *auto* (*metis exp-le-cancel-iff exp-ln-iff*)

corollary *0* ≤ *x* ⟹ *x* ≤ *4.64* ⟹ *exp x* ≤ *exp-cf3 x*
by (*metis numer-cf3-pos neg-le-iff-le exp-cf3-upper-bound-pos*)

Lower bound for negative x, provided *0* < *exp-cf3 x*]

lemma *ln-exp-cf3-lower-bound-neg*:
assumes *x0*: *x* ≤ *0* **and** *xgtr*: *numer-cf3 x* > *0*
shows *ln* (*exp-cf3 x*) ≤ *x*
proof −
have *ncf3*: ⋀*y*. *x* ≤ *y* ⟹ *y* ≤ *0* ⟹ *numer-cf3 y* > *0*
by (*metis dual-order.strict-trans1 numer-cf3-mono xgtr*)
show *?thesis*
apply (*rule gen-lower-bound-decreasing* [*OF x0 d-delta-lnexp-cf3 d-delta-lnexp-cf3-nonneg*])
apply (*auto simp add*: *ncf3 assms numer-cf3-pos*)
apply (*simp add*: *exp-cf3-def numer-cf3-def*)
done
qed

theorem *exp-cf3-lower-bound-pos*:
assumes *x* ≤ *0* **shows** *exp-cf3 x* ≤ *exp x*
proof (*cases numer-cf3 x* > *0*)
case *True*
then have *exp-cf3 x* > *0*
using *assms numer-cf3-pos* [*of* −*x*]
by (*auto simp*: *exp-cf3-pos*)
then show *?thesis*

using *ln-exp-cf3-lower-bound-neg* [*of x*] *assms True*
by *auto* (*metis exp-le-cancel-iff exp-ln-iff*)
next
case *False*
then have *exp-cf3 x* ≤ *0*
using *assms numer-cf3-pos* [*of* −*x*]
unfolding *exp-cf3-def*
by (*simp add*: *divide-nonpos-pos*)
then show *?thesis*
by (*metis exp-ge-zero order.trans*)
qed

3.4 Continued Fraction Bound 4

Here we have the extended exp continued fraction bounds

definition *numer-cf4* :: *real* ⇒ *real*
where *numer-cf4* ≡ λ*x*. *x*^*4* + *20*∗*x*^*3* + *180*∗*x*^*2* + *840*∗*x* + *1680*

definition *exp-cf4* :: *real* ⇒ *real*
where *exp-cf4* ≡ λ*x*. *numer-cf4 x* / *numer-cf4* (−*x*)

lemma *numer-cf4-pos*: **fixes** *x*::*real* **shows** *numer-cf4 x* > *0*
unfolding *numer-cf4-def*
by (*sos* ((*R<1* + ((*R<1* ∗ ((*R<4469/256* ∗ [*1135/71504*∗*x*^*2* + *4725/17876*∗*x* + *1*]^*2*) + ((*R<3728645/18305024* ∗ [*536265/2982916*∗*x*^*2* + *x*]^*2*) + (*R<106265/24436047872* ∗ [*x*^*2*]^*2*)))) + ((*A<=0* ∗ *R<1*) ∗ (*R<45/4096* ∗ [*1*]^*2*))))))

lemma *exp-cf4-pos*: *exp-cf4 x* > *0*
unfolding *exp-cf4-def*
by (*auto simp add*: *divide-simps numer-cf4-pos*)

definition *diff-delta-lnexp-cf4* :: *real* ⇒ *real*
where *diff-delta-lnexp-cf4* ≡ λ*x*. − (*x*^*8*) / (*numer-cf4* (−*x*) ∗ *numer-cf4 x*)

lemma *d-delta-lnexp-cf4-nonpos*: *diff-delta-lnexp-cf4 x* ≤ *0*
unfolding *diff-delta-lnexp-cf4-def*
using *numer-cf4-pos* [*of x*] *numer-cf4-pos* [*of* −*x*]
by (*simp add*: *zero-le-divide-iff zero-le-mult-iff*)

lemma *d-delta-lnexp-cf4*:
((λ*x*. *ln* (*exp-cf4 x*) − *x*) *has-field-derivative diff-delta-lnexp-cf4 x*) (*at x*)
unfolding *exp-cf4-def numer-cf4-def diff-delta-lnexp-cf4-def*
apply (*intro derivative-eq-intros* | *simp*)+
using *exp-cf4-pos*
apply (*simp add*: *exp-cf4-def numer-cf4-def*)
apply (*simp-all add*:)
using *numer-cf4-pos* [*of x*] *numer-cf4-pos* [*of* −*x*]
apply (*auto simp*: *divide-simps numer-cf4-def*)

apply *algebra*
done

Upper bound for non-positive x

lemma *ln-exp-cf4-upper-bound-neg*:
assumes $x \leq 0$
shows $x \leq ln\ (exp\text{-}cf4\ x)$
by (*rule gen-upper-bound-decreasing* [*OF assms d-delta-lnexp-cf4 d-delta-lnexp-cf4-nonpos*])
(*simp add*: *exp-cf4-def numer-cf4-def*)

theorem *exp-cf4-upper-bound-neg*: $x \leq 0 \Longrightarrow exp(x) \leq exp\text{-}cf4\ x$
by (*metis ln-exp-cf4-upper-bound-neg exp-cf4-pos exp-le-cancel-iff exp-ln-iff*)

Lower bound for non-negative x

lemma *ln-exp-cf4-lower-bound-pos*:
assumes $0 \leq x$
shows $ln\ (exp\text{-}cf4\ x) \leq x$
by (*rule gen-lower-bound-increasing* [*OF assms d-delta-lnexp-cf4 d-delta-lnexp-cf4-nonpos*])
(*simp add*: *exp-cf4-def numer-cf4-def*)

theorem *exp-cf4-lower-bound-pos*: $0 \leq x \Longrightarrow exp\text{-}cf4\ x \leq exp\ x$
by (*metis exp-cf4-pos exp-le-cancel-iff exp-ln ln-exp-cf4-lower-bound-pos*)

3.5 Continued Fraction Bound 5

This bound crosses the X-axis twice, causing complications.

definition *numer-cf5* :: *real* ⇒ *real*
where $numer\text{-}cf5 \equiv \lambda x.\ x\hat{}5 + 30*x\hat{}4 + 420*x\hat{}3 + 3360*x\hat{}2 + 15120*x + 30240$

definition *exp-cf5* :: *real* ⇒ *real*
where $exp\text{-}cf5 \equiv \lambda x.\ numer\text{-}cf5\ x\ /\ numer\text{-}cf5\ (-x)$

lemma *numer-cf5-pos*: $-7.293 \leq x \Longrightarrow numer\text{-}cf5\ x > 0$
unfolding *numer-cf5-def*
by *sturm*

lemma *exp-cf5-pos*: $numer\text{-}cf5\ x > 0 \Longrightarrow numer\text{-}cf5\ (-x) > 0 \Longrightarrow exp\text{-}cf5\ x > 0$
unfolding *exp-cf5-def numer-cf5-def*
by (*simp add*: *divide-neg-neg*)

definition *diff-delta-lnexp-cf5* :: *real* ⇒ *real*
where $diff\text{-}delta\text{-}lnexp\text{-}cf5 \equiv \lambda x.\ (x\hat{}10)\ /\ (numer\text{-}cf5\ (-x) * numer\text{-}cf5\ x)$

lemma *d-delta-lnexp-cf5-nonneg*: $numer\text{-}cf5\ x > 0 \Longrightarrow numer\text{-}cf5\ (-x) > 0 \Longrightarrow diff\text{-}delta\text{-}lnexp\text{-}cf5\ x \geq 0$
unfolding *diff-delta-lnexp-cf5-def*
by (*auto simp add*: *mult-less-0-iff intro*: *divide-nonneg-neg*)

lemma *d-delta-lnexp-cf5*:
  assumes *numer-cf5 x > 0 numer-cf5 (−x) > 0*
  shows ((λ*x. ln (exp-cf5 x) − x) has-field-derivative diff-delta-lnexp-cf5 x*) (*at x*)
unfolding *exp-cf5-def numer-cf5-def diff-delta-lnexp-cf5-def*
apply (*intro derivative-eq-intros* | *simp*)+
using *assms numer-cf5-pos* [*of x*] *numer-cf5-pos* [*of −x*]
apply (*auto simp*: *numer-cf5-def*)
apply (*auto simp add*: *divide-simps add-nonneg-eq-0-iff*)
apply *algebra*
done

3.5.1 Proving monotonicity via a non-negative derivative

definition *numer-cf5-deriv* :: *real* ⇒ *real*
 where *numer-cf5-deriv* ≡ λ*x*. 5*x^4 + 120*x^3 + 1260*x^2 + 6720*x + 15120

lemma *numer-cf5-deriv*:
  shows (*numer-cf5 has-field-derivative numer-cf5-deriv x*) (*at x*)
  unfolding *numer-cf5-def numer-cf5-deriv-def*
  by (*intro derivative-eq-intros* | *simp*)+

lemma *numer-cf5-deriv-pos*: *numer-cf5-deriv x* ≥ *0*
  unfolding *numer-cf5-deriv-def*
 by (*sos* ((R<1 + ((R<1 * ((R<185533/8192 * [73459/5937056*x^2 + 43050/185533*x + 1]^2) + ((R<4641265253/24318181376 * [700850925/4641265253*x^2 + x]^2) + (R<38142496079/38933754831437824 * [x^2]^2)))) + ((A<0 * R<1) * (R<205/131072 * [1]^2))))))

lemma *numer-cf5-mono*: *y* ≤ *x* ⟹ *numer-cf5 y* ≤ *numer-cf5 x*
 by (*auto intro*: *DERIV-nonneg-imp-nondecreasing numer-cf5-deriv numer-cf5-deriv-pos*)

3.5.2 Results

Upper bound for non-negative x

lemma *ln-exp-cf5-upper-bound-nonneg*:
  assumes *x0*: *0* ≤ *x* **and** *xless*: *numer-cf5* (−*x*) > *0*
    shows *x* ≤ *ln* (*exp-cf5 x*)
proof −
  have *ncf5*: ⋀*y*. *0* ≤ *y* ⟹ *y* ≤ *x* ⟹ *numer-cf5* (−*y*) > *0*
    by (*metis neg-le-iff-le numer-cf5-mono order.strict-trans2 xless*)
  show *?thesis*
   apply (*rule gen-upper-bound-increasing* [*OF x0 d-delta-lnexp-cf5 d-delta-lnexp-cf5-nonneg*])
    apply (*auto simp add*: *ncf5 assms numer-cf5-pos*)
    apply (*simp add*: *exp-cf5-def numer-cf5-def*)
    done
qed

theorem *exp-cf5-upper-bound-pos*: $0 \leq x \Longrightarrow$ *numer-cf5* $(-x) > 0 \Longrightarrow$ *exp* $x \leq$ *exp-cf5* x
using *ln-exp-cf5-upper-bound-nonneg* [*of* x] *exp-cf5-pos* [*of* x] *numer-cf5-pos* [*of* x]
by *auto* (*metis exp-le-cancel-iff exp-ln-iff*)

corollary $0 \leq x \Longrightarrow x \leq 7.293 \Longrightarrow$ *exp* $x \leq$ *exp-cf5* x
by (*metis neg-le-iff-le numer-cf5-pos exp-cf5-upper-bound-pos*)

Lower bound for negative x, provided $0 <$ *exp-cf5* x]

lemma *ln-exp-cf5-lower-bound-neg*:
assumes *x0*: $x \leq 0$ **and** *xgtr*: *numer-cf5* $x > 0$
shows *ln* (*exp-cf5* x) $\leq x$
proof −
have *ncf5*: $\bigwedge y.\ x \leq y \Longrightarrow y \leq 0 \Longrightarrow$ *numer-cf5* $y > 0$
by (*metis dual-order.strict-trans1 numer-cf5-mono xgtr*)
show *?thesis*
apply (*rule gen-lower-bound-decreasing* [*OF x0 d-delta-lnexp-cf5 d-delta-lnexp-cf5-nonneg*])
apply (*auto simp add*: *ncf5 assms numer-cf5-pos*)
apply (*simp add*: *exp-cf5-def numer-cf5-def*)
done
qed

theorem *exp-cf5-lower-bound-pos*:
assumes $x \leq 0$ **shows** *exp-cf5* $x \leq$ *exp* x
proof (*cases numer-cf5* $x > 0$)
case *True*
then have *exp-cf5* $x > 0$
using *assms numer-cf5-pos* [*of* $-x$]
by (*auto simp*: *exp-cf5-pos*)
then show *?thesis*
using *ln-exp-cf5-lower-bound-neg* [*of* x] *assms True*
by *auto* (*metis exp-le-cancel-iff exp-ln-iff*)
next
case *False*
then have *exp-cf5* $x \leq 0$
using *assms numer-cf5-pos* [*of* $-x$]
unfolding *exp-cf5-def numer-cf5-def*
by (*simp add*: *divide-nonpos-pos*)
then show *?thesis*
by (*metis exp-ge-zero order.trans*)
qed

3.6 Continued Fraction Bound 6

definition *numer-cf6* :: *real* ⇒ *real*
where *numer-cf6* $\equiv \lambda x.\ x\hat{}6 + 42{*}x\hat{}5 + 840{*}x\hat{}4 + 10080{*}x\hat{}3 + 75600{*}x\hat{}2 + 332640{*}x + 665280$

definition *exp-cf6* :: *real* ⇒ *real*
where *exp-cf6* ≡ λ*x*. *numer-cf6 x* / *numer-cf6* (−*x*)

lemma *numer-cf6-pos*: **fixes** *x*::*real* **shows** *numer-cf6 x* > *0*
unfolding *numer-cf6-def*
by *sturm*

lemma *exp-cf6-pos*: *exp-cf6 x* > *0*
unfolding *exp-cf6-def*
by (*auto simp add*: *divide-simps numer-cf6-pos*)

definition *diff-delta-lnexp-cf6* :: *real* ⇒ *real*
where *diff-delta-lnexp-cf6* ≡ λ*x*. − (*x*^*12*) / (*numer-cf6* (−*x*) ∗ *numer-cf6 x*)

lemma *d-delta-lnexp-cf6-nonpos*: *diff-delta-lnexp-cf6 x* ≤ *0*
unfolding *diff-delta-lnexp-cf6-def*
using *numer-cf6-pos* [*of x*] *numer-cf6-pos* [*of* −*x*]
by (*simp add*: *zero-le-divide-iff zero-le-mult-iff*)

lemma *d-delta-lnexp-cf6*:
((λ*x*. *ln* (*exp-cf6 x*) − *x*) *has-field-derivative diff-delta-lnexp-cf6 x*) (*at x*)
unfolding *exp-cf6-def diff-delta-lnexp-cf6-def numer-cf6-def*
apply (*intro derivative-eq-intros* | *simp*)+
using *exp-cf6-pos*
apply (*simp add*: *exp-cf6-def numer-cf6-def*)
apply (*simp-all add*:)
using *numer-cf6-pos* [*of x*] *numer-cf6-pos* [*of* −*x*]
apply (*auto simp*: *divide-simps numer-cf6-def*)
apply *algebra*
done

Upper bound for non-positive x

lemma *ln-exp-cf6-upper-bound-neg*:
assumes *x* ≤ *0*
shows *x* ≤ *ln* (*exp-cf6 x*)
by (*rule gen-upper-bound-decreasing* [*OF assms d-delta-lnexp-cf6 d-delta-lnexp-cf6-nonpos*])
(*simp add*: *exp-cf6-def numer-cf6-def*)

theorem *exp-cf6-upper-bound-neg*: *x*≤*0* ⟹ *exp*(*x*) ≤ *exp-cf6 x*
by (*metis ln-exp-cf6-upper-bound-neg exp-cf6-pos exp-le-cancel-iff exp-ln-iff*)

Lower bound for non-negative x

lemma *ln-exp-cf6-lower-bound-pos*:
assumes *0*≤*x*
shows *ln* (*exp-cf6 x*) ≤ *x*
by (*rule gen-lower-bound-increasing* [*OF assms d-delta-lnexp-cf6 d-delta-lnexp-cf6-nonpos*])
(*simp add*: *exp-cf6-def numer-cf6-def*)

theorem *exp-cf6-lower-bound-pos*: *0*≤*x* ⟹ *exp-cf6 x* ≤ *exp x*
by (*metis exp-cf6-pos exp-le-cancel-iff exp-ln ln-exp-cf6-lower-bound-pos*)

3.7 Continued Fraction Bound 7

This bound crosses the X-axis twice, causing complications.

definition *numer-cf7* :: *real* ⇒ *real*
where *numer-cf7* ≡ λ*x*. *x*^*7* + *56*∗*x*^*6* + *1512*∗*x*^*5* + *25200*∗*x*^*4* + *277200*∗*x*^*3* + *1995840*∗*x*^*2* + *8648640*∗*x* + *17297280*

definition *exp-cf7* :: *real* ⇒ *real*
where *exp-cf7* ≡ λ*x*. *numer-cf7 x* / *numer-cf7* (−*x*)

lemma *numer-cf7-pos*: −*9.943* ≤ *x* ⟹ *numer-cf7 x* > *0*
unfolding *numer-cf7-def*
by *sturm*

lemma *exp-cf7-pos*: *numer-cf7 x* > *0* ⟹ *numer-cf7* (−*x*) > *0* ⟹ *exp-cf7 x* > *0*
by (*simp add*: *exp-cf7-def*)

definition *diff-delta-lnexp-cf7* :: *real* ⇒ *real*
where *diff-delta-lnexp-cf7* ≡ λ*x*. (*x*^*14*) / (*numer-cf7* (−*x*) ∗ *numer-cf7 x*)

lemma *d-delta-lnexp-cf7-nonneg*: *numer-cf7 x* > *0* ⟹ *numer-cf7* (−*x*) > *0* ⟹ *diff-delta-lnexp-cf7 x* ≥ *0*
unfolding *diff-delta-lnexp-cf7-def*
by (*auto simp*: *mult-less-0-iff intro*: *divide-nonneg-neg*)

lemma *d-delta-lnexp-cf7*:
assumes *numer-cf7 x* > *0* *numer-cf7* (−*x*) > *0*
shows ((λ*x*. *ln* (*exp-cf7 x*) − *x*) *has-field-derivative diff-delta-lnexp-cf7 x*) (*at x*)
unfolding *exp-cf7-def numer-cf7-def diff-delta-lnexp-cf7-def*
apply (*intro derivative-eq-intros* | *simp*)+
using *assms numer-cf7-pos* [*of x*] *numer-cf7-pos* [*of* −*x*]
apply (*auto simp*: *numer-cf7-def*)
apply (*auto simp*: *divide-simps add-nonneg-eq-0-iff*)
apply *algebra*
done

3.7.1 Proving monotonicity via a non-negative derivative

definition *numer-cf7-deriv* :: *real* ⇒ *real*
where *numer-cf7-deriv* ≡ λ*x*. *7*∗*x*^*6* + *336*∗*x*^*5* + *7560*∗*x*^*4* + *100800*∗*x*^*3* + *831600*∗*x*^*2* + *3991680*∗*x* + *8648640*

lemma *numer-cf7-deriv*:
shows (*numer-cf7 has-field-derivative numer-cf7-deriv x*) (*at x*)
unfolding *numer-cf7-def numer-cf7-deriv-def*
by (*intro derivative-eq-intros* | *simp*)+

lemma *numer-cf7-deriv-pos*: *numer-cf7-deriv x* ≥ *0*
unfolding *numer-cf7-deriv-def*

apply (*rule order.strict-implies-order*) — FIXME should not be necessary
by *sturm*

lemma *numer-cf7-mono*: $y \le x \Longrightarrow$ *numer-cf7* $y \le$ *numer-cf7* x
by (*auto intro*: *DERIV-nonneg-imp-nondecreasing numer-cf7-deriv numer-cf7-deriv-pos*)

3.7.2 Results

Upper bound for non-negative x

lemma *ln-exp-cf7-upper-bound-nonneg*:
assumes *x0*: $0 \le x$ **and** *xless*: *numer-cf7* $(-x) > 0$
shows $x \le$ *ln* (*exp-cf7* x)
proof −
have *ncf7*: $\bigwedge y.\ 0 \le y \Longrightarrow y \le x \Longrightarrow$ *numer-cf7* $(-y) > 0$
by (*metis neg-le-iff-le numer-cf7-mono order.strict-trans2 xless*)
show *?thesis*
apply (*rule gen-upper-bound-increasing* [*OF x0 d-delta-lnexp-cf7 d-delta-lnexp-cf7-nonneg*])
apply (*auto simp add*: *ncf7 assms numer-cf7-pos*)
apply (*simp add*: *exp-cf7-def numer-cf7-def*)
done
qed

theorem *exp-cf7-upper-bound-pos*: $0 \le x \Longrightarrow$ *numer-cf7* $(-x) > 0 \Longrightarrow$ *exp* $x \le$ *exp-cf7* x
using *ln-exp-cf7-upper-bound-nonneg* [*of x*] *exp-cf7-pos* [*of x*] *numer-cf7-pos* [*of x*]
by *auto* (*metis exp-le-cancel-iff exp-ln-iff*)

corollary $0 \le x \Longrightarrow x \le 9.943 \Longrightarrow$ *exp* $x \le$ *exp-cf7* x
by (*metis neg-le-iff-le numer-cf7-pos exp-cf7-upper-bound-pos*)

Lower bound for negative x, provided $0 <$ *exp-cf7* x]

lemma *ln-exp-cf7-lower-bound-neg*:
assumes *x0*: $x \le 0$ **and** *xgtr*: *numer-cf7* $x > 0$
shows *ln* (*exp-cf7* x) $\le x$
proof −
have *ncf7*: $\bigwedge y.\ x \le y \Longrightarrow y \le 0 \Longrightarrow$ *numer-cf7* $y > 0$
by (*metis dual-order.strict-trans1 numer-cf7-mono xgtr*)
show *?thesis*
apply (*rule gen-lower-bound-decreasing* [*OF x0 d-delta-lnexp-cf7 d-delta-lnexp-cf7-nonneg*])
apply (*auto simp add*: *ncf7 assms numer-cf7-pos*)
apply (*simp add*: *exp-cf7-def numer-cf7-def*)
done
qed

theorem *exp-cf7-lower-bound-pos*:
assumes $x \le 0$ **shows** *exp-cf7* $x \le$ *exp* x
proof (*cases numer-cf7* $x > 0$)
case *True*

```
  then have exp-cf7 x > 0
    using assms numer-cf7-pos [of −x]
    by (auto simp: exp-cf7-pos)
  then show ?thesis
  using ln-exp-cf7-lower-bound-neg [of x] assms True
    by auto (metis exp-le-cancel-iff exp-ln-iff)
next
  case False
  then have exp-cf7 x ≤ 0
    using assms numer-cf7-pos [of −x]
    unfolding exp-cf7-def
    by (simp add: divide-nonpos-pos)
  then show ?thesis
    by (metis exp-ge-zero order.trans)
qed

end
```

Chapter 4

Log Upper and Lower Bounds

theory *Log-CF-Bounds*
imports *Bounds-Lemmas*

begin

theorem *ln-upper-1*: $0<x \Longrightarrow ln(x::real) \leq x - 1$
by (*rule ln-le-minus-one*)

definition *ln-lower-1* :: *real* ⇒ *real*
  where *ln-lower-1* ≡ $\lambda x.\ 1 - (inverse\ x)$

corollary *ln-lower-1*: $0<x \Longrightarrow ln\text{-}lower\text{-}1\ x \leq ln\ x$
  unfolding *ln-lower-1-def*
  by (*metis ln-inverse ln-le-minus-one positive-imp-inverse-positive minus-diff-eq minus-le-iff*)

theorem *ln-lower-1-eq*: $0<x \Longrightarrow ln\text{-}lower\text{-}1\ x = (x - 1)/x$
  by (*auto simp*: *ln-lower-1-def divide-simps*)

4.1 Upper Bound 3

definition *ln-upper-3* :: *real* ⇒ *real*
  where *ln-upper-3* ≡ $\lambda x.\ (x + 5)*(x - 1)\ /\ (2*(2*x + 1))$

definition *diff-delta-ln-upper-3* :: *real* ⇒ *real*
  where *diff-delta-ln-upper-3* ≡ $\lambda x.\ (x - 1)\hat{}3\ /\ ((2*x + 1)\hat{}2 * x)$

lemma *d-delta-ln-upper-3*: $x > 0 \Longrightarrow$
    $((\lambda x.\ ln\text{-}upper\text{-}3\ x - ln\ x)$ *has-field-derivative diff-delta-ln-upper-3 x*) (*at x*)
unfolding *ln-upper-3-def diff-delta-ln-upper-3-def*
apply (*intro derivative-eq-intros* | *simp*)+

apply (*simp add*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
done

Strict inequalities also possible

lemma *ln-upper-3-pos*:
  assumes *1* ≤ *x* **shows** *ln*(*x*) ≤ *ln-upper-3 x*
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-ln-upper-3*])
apply (*auto simp*: *diff-delta-ln-upper-3-def ln-upper-3-def*)
done

lemma *ln-upper-3-neg*:
  assumes *0* < *x* **and** *x1*: *x* ≤ *1* **shows** *ln*(*x*) ≤ *ln-upper-3 x*
apply (*rule gen-upper-bound-decreasing* [*OF x1 d-delta-ln-upper-3*])
using *assms*
apply (*auto simp*: *diff-delta-ln-upper-3-def divide-simps ln-upper-3-def*)
done

theorem *ln-upper-3*: *0*<*x* ⟹ *ln*(*x*) ≤ *ln-upper-3 x*
by (*metis le-less-linear less-eq-real-def ln-upper-3-neg ln-upper-3-pos*)

definition *ln-lower-3* :: *real* ⇒ *real*
  where *ln-lower-3* ≡ λ*x*. − *ln-upper-3* (*inverse x*)

corollary *ln-lower-3*: *0*<*x* ⟹ *ln-lower-3 x* ≤ *ln x*
  unfolding *ln-lower-3-def*
  by (*metis ln-inverse inverse-positive-iff-positive minus-le-iff ln-upper-3*)

theorem *ln-lower-3-eq*: *0*<*x* ⟹ *ln-lower-3 x* = (*1*/*2*)∗(*1* + *5*∗*x*)∗(*x* − *1*) / (*x*∗(*2* + *x*))
  unfolding *ln-lower-3-def ln-upper-3-def*
  by (*simp add*: *divide-simps*) *algebra*

4.2 Upper Bound 5

definition *ln-upper-5* :: *real* ⇒ *real*
  where *ln-upper-5 x* ≡ (*x*ˆ*2* + *19*∗*x* + *10*)∗(*x* − *1*) / (*3*∗(*3*∗*x*ˆ*2* + *6*∗*x* + *1*))

definition *diff-delta-ln-upper-5* :: *real* ⇒ *real*
  where *diff-delta-ln-upper-5* ≡ λ*x*. (*x* − *1*)ˆ*5* / ((*3*∗*x*ˆ*2* + *6*∗*x* + *1*)ˆ*2*∗*x*)

lemma *d-delta-ln-upper-5*: *x* > *0* ⟹
    ((λ*x*. *ln-upper-5 x* − *ln x*) *has-field-derivative diff-delta-ln-upper-5 x*) (*at x*)
  unfolding *ln-upper-5-def diff-delta-ln-upper-5-def*
  apply (*intro derivative-eq-intros* | *simp add*: *add-nonneg-eq-0-iff*)+
  apply (*simp add*: *divide-simps add-nonneg-eq-0-iff*, *algebra*)
  done

lemma *ln-upper-5-pos*:
  assumes *1* ≤ *x* **shows** *ln*(*x*) ≤ *ln-upper-5 x*

apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-ln-upper-5*])
apply (*auto simp*: *diff-delta-ln-upper-5-def ln-upper-5-def*)
done

lemma *ln-upper-5-neg*:
  assumes *0* < *x* **and** *x1*: *x* ≤ *1* **shows** *ln*(*x*) ≤ *ln-upper-5 x*
apply (*rule gen-upper-bound-decreasing* [*OF x1 d-delta-ln-upper-5*])
using *assms*
apply (*auto simp*: *diff-delta-ln-upper-5-def divide-simps ln-upper-5-def mult-less-0-iff*)
done

theorem *ln-upper-5*: *0*<*x* ⟹ *ln*(*x*) ≤ *ln-upper-5 x*
by (*metis le-less-linear less-eq-real-def ln-upper-5-neg ln-upper-5-pos*)

definition *ln-lower-5* :: *real* ⇒ *real*
  where *ln-lower-5* ≡ λ*x*. − *ln-upper-5* (*inverse x*)

corollary *ln-lower-5*: *0*<*x* ⟹ *ln-lower-5 x* ≤ *ln x*
  unfolding *ln-lower-5-def*
  by (*metis ln-inverse inverse-positive-iff-positive minus-le-iff ln-upper-5*)

theorem *ln-lower-5-eq*: *0*<*x* ⟹
    ln-lower-5 x = (*1*/*3*)∗(*10*∗*x*ˆ*2* + *19*∗*x* + *1*)∗(*x* − *1*) / (*x*∗(*x*ˆ*2* + *6*∗*x* + *3*))
  unfolding *ln-lower-5-def ln-upper-5-def*
  by (*simp add*: *zero-less-mult-iff add-pos-pos dual-order.strict-implies-not-eq divide-simps*)
     algebra

4.3 Upper Bound 7

definition *ln-upper-7* :: *real* ⇒ *real*
  where *ln-upper-7 x* ≡ (*3*∗*x*ˆ*3* + *131*∗*x*ˆ*2* + *239*∗*x* + *47*)∗(*x* − *1*) / (*12*∗(*4*∗*x*ˆ*3* + *18*∗*x*ˆ*2* + *12*∗*x* + *1*))

definition *diff-delta-ln-upper-7* :: *real* ⇒ *real*
  where *diff-delta-ln-upper-7* ≡ λ*x*. (*x* − *1*)ˆ*7* / ((*4*∗*x*ˆ*3* + *18*∗*x*ˆ*2* + *12*∗*x* + *1*)ˆ*2* ∗ *x*)

lemma *d-delta-ln-upper-7*: *x* > *0* ⟹
    ((λ*x*. *ln-upper-7 x* − *ln x*) *has-field-derivative diff-delta-ln-upper-7 x*) (*at x*)
unfolding *ln-upper-7-def diff-delta-ln-upper-7-def*
apply (*intro derivative-eq-intros* | *simp*)+
apply *auto*
apply (*auto simp*: *add-pos-pos dual-order.strict-implies-not-eq divide-simps*, *algebra*)
done

lemma *ln-upper-7-pos*:
  assumes *1* ≤ *x* **shows** *ln*(*x*) ≤ *ln-upper-7 x*

apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-ln-upper-7*])
apply (*auto simp*: *diff-delta-ln-upper-7-def ln-upper-7-def*)
done

lemma *ln-upper-7-neg*:
assumes $0 < x$ **and** *x1*: $x \leq 1$ **shows** $ln(x) \leq$ *ln-upper-7 x*
apply (*rule gen-upper-bound-decreasing* [*OF x1 d-delta-ln-upper-7*])
using *assms*
apply (*auto simp*: *diff-delta-ln-upper-7-def divide-simps ln-upper-7-def mult-less-0-iff*)
done

theorem *ln-upper-7*: $0 < x \Longrightarrow ln(x) \leq$ *ln-upper-7 x*
by (*metis le-less-linear less-eq-real-def ln-upper-7-neg ln-upper-7-pos*)

definition *ln-lower-7* :: *real* ⇒ *real*
where *ln-lower-7* ≡ λ*x*. − *ln-upper-7* (*inverse x*)

corollary *ln-lower-7*: $0 < x \Longrightarrow$ *ln-lower-7 x* ≤ *ln x*
unfolding *ln-lower-7-def*
by (*metis ln-inverse inverse-positive-iff-positive minus-le-iff ln-upper-7*)

theorem *ln-lower-7-eq*: $0 < x \Longrightarrow$
ln-lower-7 x = (*1/12*)∗(*47*∗*x*^*3* + *239*∗*x*^*2* + *131*∗*x* + *3*)∗(*x* − *1*) / (*x*∗(*x*^*3* + *12*∗*x*^*2* + *18*∗*x* + *4*))
unfolding *ln-lower-7-def ln-upper-7-def*
by (*simp add*: *zero-less-mult-iff add-pos-pos dual-order.strict-implies-not-eq divide-simps*)
algebra

4.4 Upper Bound 9

definition *ln-upper-9* :: *real* ⇒ *real*
where *ln-upper-9 x* ≡ (*6*∗*x*^*4* + *481*∗*x*^*3* + *1881*∗*x*^*2* + *1281*∗*x* + *131*)∗(*x* − *1*) /
(*30* ∗ (*5*∗*x*^*4* + *40*∗*x*^*3* + *60*∗*x*^*2* + *20*∗*x* + *1*))

definition *diff-delta-ln-upper-9* :: *real* ⇒ *real*
where *diff-delta-ln-upper-9* ≡ λ*x*. (*x* − *1*)^*9* / (((*5*∗*x*^*4* + *40*∗*x*^*3* + *60*∗*x*^*2* + *20*∗*x* + *1*)^*2*) ∗ *x*)

lemma *d-delta-ln-upper-9*: $x > 0 \Longrightarrow$
((λ*x*. *ln-upper-9 x* − *ln x*) *has-field-derivative diff-delta-ln-upper-9 x*) (*at x*)
unfolding *ln-upper-9-def diff-delta-ln-upper-9-def*
apply (*intro derivative-eq-intros* | *simp*)+
apply *auto*
apply (*auto simp*: *add-pos-pos dual-order.strict-implies-not-eq divide-simps*, *algebra*)
done

lemma *ln-upper-9-pos*:
assumes *1* ≤ *x* **shows** *ln*(*x*) ≤ *ln-upper-9 x*
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-ln-upper-9*])
apply (*auto simp*: *diff-delta-ln-upper-9-def ln-upper-9-def*)
done

lemma *ln-upper-9-neg*:
assumes *0* < *x* **and** *x1*: *x* ≤ *1* **shows** *ln*(*x*) ≤ *ln-upper-9 x*
apply (*rule gen-upper-bound-decreasing* [*OF x1 d-delta-ln-upper-9*])
using *assms*
apply (*auto simp*: *diff-delta-ln-upper-9-def divide-simps ln-upper-9-def mult-less-0-iff*)
done

theorem *ln-upper-9*: *0*<*x* ⟹ *ln*(*x*) ≤ *ln-upper-9 x*
by (*metis le-less-linear less-eq-real-def ln-upper-9-neg ln-upper-9-pos*)

definition *ln-lower-9* :: *real* ⇒ *real*
where *ln-lower-9* ≡ λ*x*. − *ln-upper-9* (*inverse x*)

corollary *ln-lower-9*: *0* < *x* ⟹ *ln-lower-9 x* ≤ *ln x*
unfolding *ln-lower-9-def*
by (*metis ln-inverse inverse-positive-iff-positive minus-le-iff ln-upper-9*)

theorem *ln-lower-9-eq*: *0* < *x* ⟹
ln-lower-9 x = (*1*/*30*)∗(*6* + *481*∗*x* + *1881*∗*x*^*2* + *1281*∗*x*^*3* + *131*∗*x*^*4*)∗(*x* − *1*) /
(*x*∗(*5* + *40*∗*x* + *60*∗*x*^*2* + *20*∗*x*^*3* + *x*^*4*))
unfolding *ln-lower-9-def ln-upper-9-def*
by (*simp add*: *zero-less-mult-iff add-pos-pos dual-order.strict-implies-not-eq divide-simps*)
algebra

4.5 Upper Bound 11

Extended bounds start here

definition *ln-upper-11* :: *real* ⇒ *real*
where *ln-upper-11 x* ≡
(*5*∗*x*^*5* + *647*∗*x*^*4* + *4397*∗*x*^*3* + *6397*∗*x*^*2* + *2272*∗*x* + *142*) ∗ (*x* − *1*) /
(*30*∗(*6*∗*x*^*5* + *75*∗*x*^*4* + *200*∗*x*^*3* + *150*∗*x*^*2* + *30*∗*x* + *1*))

definition *diff-delta-ln-upper-11* :: *real* ⇒ *real*
where *diff-delta-ln-upper-11* ≡ λ*x*. (*x* − *1*)^*11* / ((*6*∗*x*^*5* + *75*∗*x*^*4* + *200*∗*x*^*3* + *150*∗*x*^*2* + *30*∗*x* + *1*)^*2* ∗ *x*)

lemma *d-delta-ln-upper-11*: *x* > *0* ⟹
((λ*x*. *ln-upper-11 x* − *ln x*) *has-field-derivative diff-delta-ln-upper-11 x*) (*at x*)

unfolding *ln-upper-11-def diff-delta-ln-upper-11-def*
apply (*intro derivative-eq-intros* | *simp*)+
apply *auto*
apply (*auto simp*: *add-pos-pos dual-order.strict-implies-not-eq divide-simps*, *algebra*)
done

lemma *ln-upper-11-pos*:
  assumes *1* ≤ *x* **shows** *ln*(*x*) ≤ *ln-upper-11 x*
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-ln-upper-11*])
apply (*auto simp*: *diff-delta-ln-upper-11-def ln-upper-11-def*)
done

lemma *ln-upper-11-neg*:
  assumes *0* < *x* **and** *x1*: *x* ≤ *1* **shows** *ln*(*x*) ≤ *ln-upper-11 x*
apply (*rule gen-upper-bound-decreasing* [*OF x1 d-delta-ln-upper-11*])
using *assms*
apply (*auto simp*: *diff-delta-ln-upper-11-def divide-simps ln-upper-11-def mult-less-0-iff*)
done

theorem *ln-upper-11*: *0*<*x* ⟹ *ln*(*x*) ≤ *ln-upper-11 x*
by (*metis le-less-linear less-eq-real-def ln-upper-11-neg ln-upper-11-pos*)

definition *ln-lower-11* :: *real* ⇒ *real*
  where *ln-lower-11* ≡ λ*x*. − *ln-upper-11* (*inverse x*)

corollary *ln-lower-11*: *0*<*x* ⟹ *ln-lower-11 x* ≤ *ln x*
  unfolding *ln-lower-11-def*
  by (*metis ln-inverse inverse-positive-iff-positive minus-le-iff ln-upper-11*)

theorem *ln-lower-11-eq*: *0*<*x* ⟹
    ln-lower-11 x = (*1*/*30*)∗(*142*∗*x*^*5* + *2272*∗*x*^*4* + *6397*∗*x*^*3* + *4397*∗*x*^*2* + *647*∗*x* + *5*)∗(*x* − *1*) /
                (*x*∗(*x*^*5* + *30*∗*x*^*4* + *150*∗*x*^*3* + *200*∗*x*^*2* + *75*∗*x* + *6*))
  unfolding *ln-lower-11-def ln-upper-11-def*
  by (*simp add*: *zero-less-mult-iff add-pos-pos dual-order.strict-implies-not-eq divide-simps*)
    algebra

4.6 Upper Bound 13

definition *ln-upper-13* :: *real* ⇒ *real*
  where *ln-upper-13 x* ≡ (*353* + *8389*∗*x* + *20149*∗*x*^*4* + *50774*∗*x*^*3* + *38524*∗*x*^*2* + *1921*∗*x*^*5* + *10*∗*x*^*6*) ∗ (*x* − *1*)
                / (*70*∗(*1* + *42*∗*x* + *525*∗*x*^*4* + *700*∗*x*^*3* + *315*∗*x*^*2* + *126*∗*x*^*5* + *7*∗*x*^*6*))

definition *diff-delta-ln-upper-13* :: *real* ⇒ *real*
  where *diff-delta-ln-upper-13* ≡ λ*x*. (*x* − *1*)^*13* /

((1 + 42*x + 525*x^4 + 700*x^3 + 315*x^2 + 126*x^5 + 7*x^6)^2*x)

lemma *d-delta-ln-upper-13*: *x > 0 ⟹*
((λx. ln-upper-13 x − ln x) has-field-derivative diff-delta-ln-upper-13 x) (at x)
unfolding *ln-upper-13-def diff-delta-ln-upper-13-def*
apply (*intro derivative-eq-intros* | *simp*)+
apply *auto*
apply (*auto simp*: *add-pos-pos dual-order.strict-implies-not-eq divide-simps*, *algebra*)
done

lemma *ln-upper-13-pos*:
assumes *1 ≤ x* **shows** *ln(x) ≤ ln-upper-13 x*
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-ln-upper-13*])
apply (*auto simp*: *diff-delta-ln-upper-13-def ln-upper-13-def*)
done

lemma *ln-upper-13-neg*:
assumes *0 < x* **and** *x1*: *x ≤ 1* **shows** *ln(x) ≤ ln-upper-13 x*
apply (*rule gen-upper-bound-decreasing* [*OF x1 d-delta-ln-upper-13*])
using *assms*
apply (*auto simp*: *diff-delta-ln-upper-13-def divide-simps ln-upper-13-def mult-less-0-iff*)
done

theorem *ln-upper-13*: *0<x ⟹ ln(x) ≤ ln-upper-13 x*
by (*metis le-less-linear less-eq-real-def ln-upper-13-neg ln-upper-13-pos*)

definition *ln-lower-13* :: *real ⇒ real*
where *ln-lower-13 ≡ λx. − ln-upper-13* (*inverse x*)

corollary *ln-lower-13*: *0<x ⟹ ln-lower-13 x ≤ ln x*
unfolding *ln-lower-13-def*
by (*metis ln-inverse inverse-positive-iff-positive minus-le-iff ln-upper-13*)

theorem *ln-lower-13-eq*: *0<x ⟹*
ln-lower-13 x = (1/70)(10 + 1921*x + 20149*x^2 + 50774*x^3 + 38524*x^4 + 8389*x^5 + 353*x^6)*(x − 1) /*
(x(7 + 126*x + 525*x^2 + 700*x^3 + 315*x^4 + 42*x^5 + x^6))*
unfolding *ln-lower-13-def ln-upper-13-def*
by (*simp add*: *zero-less-mult-iff add-pos-pos dual-order.strict-implies-not-eq divide-simps*)
algebra

4.7 Upper Bound 15

definition *ln-upper-15* :: *real ⇒ real*
where *ln-upper-15 x ≡*

(1487 + 49199*x + 547235*x^4 + 718735*x^3 + 334575*x^2 + 141123*x^5 + 35*x^7 + 9411*x^6)*(x − 1) /
(280*(1 + 56*x + 2450*x^4 + 1960*x^3 + 588*x^2 + 1176*x^5 + 8*x^7 + 196*x^6))

definition *diff-delta-ln-upper-15* :: *real* ⇒ *real*
where *diff-delta-ln-upper-15*
≡ λx. (x − 1)^15 / ((1+56*x+2450*x^4+1960*x^3+588*x^2+8*x^7+196*x^6+1176*x^5)^2 * x)

lemma *d-delta-ln-upper-15*: *x* > *0* ⟹
((λx. *ln-upper-15 x* − *ln x*) *has-field-derivative diff-delta-ln-upper-15 x*) (*at x*)
unfolding *ln-upper-15-def diff-delta-ln-upper-15-def*
apply (*intro derivative-eq-intros* | *simp*)+
apply *auto*
apply (*auto simp*: *add-pos-pos dual-order.strict-implies-not-eq divide-simps*, *algebra*)
done

lemma *ln-upper-15-pos*:
assumes *1* ≤ *x* **shows** *ln*(*x*) ≤ *ln-upper-15 x*
apply (*rule gen-upper-bound-increasing* [*OF assms d-delta-ln-upper-15*])
apply (*auto simp*: *diff-delta-ln-upper-15-def ln-upper-15-def*)
done

lemma *ln-upper-15-neg*:
assumes *0* < *x* **and** *x1*: *x* ≤ *1* **shows** *ln*(*x*) ≤ *ln-upper-15 x*
apply (*rule gen-upper-bound-decreasing* [*OF x1 d-delta-ln-upper-15*])
using *assms*
apply (*auto simp*: *diff-delta-ln-upper-15-def divide-simps ln-upper-15-def mult-less-0-iff*)
done

theorem *ln-upper-15*: *0*<*x* ⟹ *ln*(*x*) ≤ *ln-upper-15 x*
by (*metis le-less-linear less-eq-real-def ln-upper-15-neg ln-upper-15-pos*)

definition *ln-lower-15* :: *real* ⇒ *real*
where *ln-lower-15* ≡ λx. − *ln-upper-15* (*inverse x*)

corollary *ln-lower-15*: *0*<*x* ⟹ *ln-lower-15 x* ≤ *ln x*
unfolding *ln-lower-15-def*
by (*metis ln-inverse inverse-positive-iff-positive minus-le-iff ln-upper-15*)

theorem *ln-lower-15-eq*: *0*<*x* ⟹
ln-lower-15 x = (1/280)*(35 + 9411*x + 141123*x^2 + 547235*x^3 + 718735*x^4 + 334575*x^5 + 49199*x^6 + 1487*x^7)*(x − 1) /
(x*(8 + 196*x + 1176*x^2 + 2450*x^3 + 1960*x^4 + 588*x^5 + 56*x^6 + x^7))
unfolding *ln-lower-15-def ln-upper-15-def*

by (*simp add*: *zero-less-mult-iff add-pos-pos dual-order.strict-implies-not-eq divide-simps*) *algebra*

end

Chapter 5

Sine and Cosine Upper and Lower Bounds

theory *Sin-Cos-Bounds*
imports *Bounds-Lemmas*

begin

5.1 Simple base cases

Upper bound for $(0::'a) \leq x$

```
lemma sin-le-arg:
  fixes x :: real
    shows 0 ≤ x ⟹ sin x ≤ x
  by (fact sin-x-le-x)

lemma cos-ge-1-arg:
  fixes x :: real
  assumes 0 ≤ x
    shows 1 − x ≤ cos x
  apply (rule gen-lower-bound-increasing [OF assms])
  apply (intro derivative-eq-intros, auto)
  done

lemmas sin-Taylor-0-upper-bound-pos = sin-le-arg — MetiTarski bound

lemma cos-Taylor-1-lower-bound:
  fixes x :: real
  assumes 0 ≤ x
    shows (1 − x^2 / 2) ≤ cos x
  apply (rule gen-lower-bound-increasing [OF assms])
  apply (intro derivative-eq-intros)
  apply (rule refl | simp add: sin-le-arg)+
  done
```

lemma *sin-Taylor-1-lower-bound*:
  fixes *x* :: *real*
  assumes *0* ≤ *x*
    shows (*x* − *x* ^ *3* / *6*) ≤ *sin x*
  apply (*rule gen-lower-bound-increasing* [*OF assms*])
  apply (*intro derivative-eq-intros*)
  apply (*rule refl* | *simp add*: *cos-Taylor-1-lower-bound*)+
  done

5.2 Taylor series approximants

definition *sinpoly* :: [*nat*,*real*] ⇒ *real*
  where *sinpoly n* = (λ*x*. ∑ *k*<*n*. *sin-coeff k* ∗ *x* ^ *k*)

definition *cospoly* :: [*nat*,*real*] ⇒ *real*
  where *cospoly n* = (λ*x*. ∑ *k*<*n*. *cos-coeff k* ∗ *x* ^ *k*)

lemma *sinpoly-Suc*: *sinpoly* (*Suc n*) = (λ*x*. *sinpoly n x* + *sin-coeff n* ∗ *x* ^ *n*)
  by (*simp add*: *sinpoly-def*)

lemma *cospoly-Suc*: *cospoly* (*Suc n*) = (λ*x*. *cospoly n x* + *cos-coeff n* ∗ *x* ^ *n*)
  by (*simp add*: *cospoly-def*)

lemma *sinpoly-minus* [*simp*]: *sinpoly n* (−*x*) = − *sinpoly n x*
  by (*induct n*) (*auto simp*: *sinpoly-def sin-coeff-def*)

lemma *cospoly-minus* [*simp*]: *cospoly n* (−*x*) = *cospoly n x*
  by (*induct n*) (*auto simp*: *cospoly-def cos-coeff-def*)

lemma *d-sinpoly-cospoly*:
    (*sinpoly* (*Suc n*) *has-field-derivative cospoly n x*) (*at x*)
proof (*induction n*)
  case *0* **show** *?case*
    by (*simp add*: *sinpoly-def cospoly-def*)
next
  case (*Suc n*) **show** *?case*
  proof (*cases n*=*0*)
    case *True* **then show** *?thesis*
      by (*simp add*: *sinpoly-def sin-coeff-def cospoly-def*)
  next
    case *False* **then**
    have *xn*: *x* ^(*n* − *Suc 0*) ∗ *x* = *x* ^ *n*
      by (*metis Suc-pred mult.commute not-gr0 power-Suc*)
    show *?thesis* **using** *Suc False*
      apply (*simp add*: *sinpoly-Suc* [*of Suc n*] *cospoly-def*)
      apply (*intro derivative-eq-intros* | *simp*)+

apply (*simp add*: *xn mult.assoc sin-coeff-def cos-coeff-def divide-simps del*: *fact-Suc*)
apply (*simp add*: *algebra-simps*)
done
qed
qed

lemma *d-cospoly-sinpoly*:
(*cospoly* (*Suc n*) *has-field-derivative* −*sinpoly n x*) (*at x*)
proof (*induction n*)
case *0* **show** *?case*
by (*simp add*: *sinpoly-def cospoly-def*)
next
case (*Suc n*) **show** *?case*
proof (*cases n=0*)
case *True* **then show** *?thesis*
by (*simp add*: *sinpoly-def cospoly-def cos-coeff-def*)
next
case *False* **then**
have *xn*: $x \,\hat{}\, (n - Suc\ 0) * x = x \,\hat{}\, n$
by (*metis Suc-pred mult.commute not-gr0 power-Suc*)
have *m1*: $odd\ n \Longrightarrow (-1 :: real) \,\hat{}\, ((n - Suc\ 0)\ div\ 2) = - ((-1) \,\hat{}\, (Suc\ n\ div\ 2))$
by (*cases n*) *simp-all*
show *?thesis* **using** *Suc False*
apply (*simp add*: *cospoly-Suc* [*of Suc n*] *sinpoly-def*)
apply (*intro derivative-eq-intros* | *simp*)+
apply (*simp add*: *xn mult.assoc sin-coeff-def cos-coeff-def m1 divide-simps del*: *fact-Suc*)
apply (*simp add*: *algebra-simps*)
done
qed
qed

5.3 Inductive proof of sine inequalities

lemma *sinpoly-lb-imp-cospoly-ub*:
assumes *x0*: $0 \le x$ **and** *k0*: $k>0$ **and** $\bigwedge x.\ 0 \le x \Longrightarrow sinpoly\ (k - 1)\ x \le sin\ x$
shows $cos\ x \le cospoly\ k\ x$
apply (*rule gen-lower-bound-increasing* [*OF x0*])
apply (*intro derivative-eq-intros* | *simp*)+
using *d-cospoly-sinpoly* [*of k* − *1*] *assms*
apply *auto*
apply (*simp add*: *cospoly-def*)
done

lemma *cospoly-ub-imp-sinpoly-ub*:
assumes *x0*: $0 \le x$ **and** *k0*: $k>0$ **and** $\bigwedge x.\ 0 \le x \Longrightarrow cos\ x \le cospoly\ (k - 1)\ x$
shows $sin\ x \le sinpoly\ k\ x$

apply (*rule gen-lower-bound-increasing* [*OF x0*])
apply (*intro derivative-eq-intros* | *simp*)+
using *d-sinpoly-cospoly* [*of k* − *1*] *assms*
apply *auto*
apply (*simp add*: *sinpoly-def*)
done

lemma *sinpoly-ub-imp-cospoly-lb*:
assumes *x0*: *0* ≤ *x* **and** *k0*: *k*>*0* **and** ⋀*x*. *0* ≤ *x* ⟹ *sin x* ≤ *sinpoly* (*k* − *1*) *x*
shows *cospoly k x* ≤ *cos x*
apply (*rule gen-lower-bound-increasing* [*OF x0*])
apply (*intro derivative-eq-intros* | *simp*)+
using *d-cospoly-sinpoly* [*of k* − *1*] *assms*
apply *auto*
apply (*simp add*: *cospoly-def*)
done

lemma *cospoly-lb-imp-sinpoly-lb*:
assumes *x0*: *0* ≤ *x* **and** *k0*: *k*>*0* **and** ⋀*x*. *0* ≤ *x* ⟹ *cospoly* (*k* − *1*) *x* ≤ *cos x*
shows *sinpoly k x* ≤ *sin x*
apply (*rule gen-lower-bound-increasing* [*OF x0*])
apply (*intro derivative-eq-intros* | *simp*)+
using *d-sinpoly-cospoly* [*of k* − *1*] *assms*
apply *auto*
apply (*simp add*: *sinpoly-def*)
done

lemma
assumes *0* ≤ *x*
shows *sinpoly-lower-nonneg*: *sinpoly* (*4* ∗ *Suc n*) *x* ≤ *sin x* (**is** *?th1*)
and *sinpoly-upper-nonneg*: *sin x* ≤ *sinpoly* (*Suc* (*Suc* (*4*∗*n*))) *x* (**is** *?th2*)
proof −
have *sinpoly* (*4* ∗ *Suc n*) *x* ≤ *sin x* ∧ *sin x* ≤ *sinpoly* (*Suc* (*Suc* (*4*∗*n*))) *x*
using *assms*
apply (*induction n arbitrary*: *x*)
apply (*simp add*: *sinpoly-def sin-coeff-def sin-Taylor-1-lower-bound sin-Taylor-0-upper-bound-pos lessThan-nat-numeral fact-numeral*)
apply (*auto simp*: *cospoly-lb-imp-sinpoly-lb sinpoly-ub-imp-cospoly-lb cospoly-ub-imp-sinpoly-ub sinpoly-lb-imp-cospoly-ub*)
done
then show *?th1 ?th2*
using *assms*
by *auto*
qed

5.4 Collecting the results

corollary *sinpoly-upper-nonpos*:
x ≤ *0* ⟹ *sin x* ≤ *sinpoly* (*4* ∗ *Suc n*) *x*

using *sinpoly-lower-nonneg* [*of* $-x$ *n*]
by *simp*

corollary *sinpoly-lower-nonpos*:
$x \leq 0 \Longrightarrow$ *sinpoly* (*Suc* (*Suc* (*4*∗*n*))) $x \leq$ *sin* x
using *sinpoly-upper-nonneg* [*of* $-x$ *n*]
by *simp*

corollary *cospoly-lower-nonneg*:
$0 \leq x \Longrightarrow$ *cospoly* (*Suc* (*Suc* (*Suc* (*4*∗*n*)))) $x \leq$ *cos* x
by (*auto simp*: *sinpoly-upper-nonneg sinpoly-ub-imp-cospoly-lb*)

lemma *cospoly-lower*:
cospoly (*Suc* (*Suc* (*Suc* (*4*∗*n*)))) $x \leq$ *cos* x
proof (*cases rule*: *le-cases* [*of 0 x*])
case *le* **then show** *?thesis*
by (*simp add*: *cospoly-lower-nonneg*)
next
case *ge* **then show** *?thesis* **using** *cospoly-lower-nonneg* [*of* $-x$]
by *simp*
qed

lemma *cospoly-upper-nonneg*:
assumes $0 \leq x$
shows *cos* $x \leq$ *cospoly* (*Suc* (*4*∗*n*)) x
proof (*cases n*)
case *0* **then show** *?thesis*
by (*simp add*: *cospoly-def*)
next
case (*Suc m*)
then show *?thesis*
using *sinpoly-lower-nonneg* [*of - m*] *assms*
by (*auto simp*: *sinpoly-lb-imp-cospoly-ub*)
qed

lemma *cospoly-upper*:
cos $x \leq$ *cospoly* (*Suc* (*4*∗*n*)) x
proof (*cases rule*: *le-cases* [*of 0 x*])
case *le* **then show** *?thesis*
by (*simp add*: *cospoly-upper-nonneg*)
next
case *ge* **then show** *?thesis* **using** *cospoly-upper-nonneg* [*of* $-x$]
by *simp*
qed

end

Chapter 6

Square Root Upper and Lower Bounds

```
theory Sqrt-Bounds
imports Bounds-Lemmas
        HOL−Library.Sum-of-Squares

begin
```

Covers all bounds used in sqrt-upper.ax, sqrt-lower.ax and sqrt-extended.ax.

6.1 Upper bounds

```
primrec sqrtu :: [real,nat] ⇒ real where
  sqrtu x 0 = (x+1)/2
| sqrtu x (Suc n) = (sqrtu x n + x/sqrtu x n) / 2

lemma sqrtu-upper: x ≤ sqrtu x n ^ 2
proof (induction n)
  case 0 show ?case
    apply (simp add: power2-eq-square)
    apply (sos (((A<0 ∗ R<1) + (R<1 ∗ (R<1 ∗ [~1∗x + 1]^2)))))
    done
next
  case (Suc n)
  have xy: ⋀y. ⟦x ≤ y ∗ y; y ≠ 0⟧ ⟹ x ∗ (2 ∗ (y ∗ y)) ≤ x ∗ x + y ∗ (y ∗ (y ∗ y))
     by (sos (((((A<0 ∗ A<1) ∗ R<1) + ((A<0 ∗ R<1) ∗ (R<1 ∗ [~1∗y^2 + x]^2)))) &
                ((((A<0 ∗ A<1) ∗ R<1) + ((A<0 ∗ R<1) ∗ (R<1 ∗ [~1∗y^2 + x]^2))))))
  show ?case using Suc
    by (auto simp: power2-eq-square algebra-simps divide-simps xy)
qed
```

lemma sqrtu-numeral:
sqrtu x (numeral n) = (sqrtu x (pred-numeral n) + x/sqrtu x (pred-numeral n)) / 2
by (simp add: numeral-eq-Suc)

lemma sqrtu-gt-0: x ≥ 0 ⟹ sqrtu x n > 0
apply (induct n)
apply (auto simp: field-simps)
by (metis add-strict-increasing2 mult-zero-left not-real-square-gt-zero)

theorem gen-sqrt-upper: 0 ≤ x ⟹ sqrt x ≤ sqrtu x n
using real-sqrt-le-mono [OF sqrtu-upper [of x n]]
by auto (metis abs-of-nonneg dual-order.strict-iff-order sqrtu-gt-0)

lemma sqrt-upper-bound-0:
assumes x ≥ 0 **shows** sqrt x ≤ (x+1)/2 (**is** - ≤ ?rhs)
proof −
have sqrt x ≤ sqrtu x 0
by (metis assms gen-sqrt-upper)
also have ... = ?rhs
by (simp add: divide-simps)
finally show ?thesis **.**
qed

lemma sqrt-upper-bound-1:
assumes x ≥ 0 **shows** sqrt x ≤ (1/4)*(x^2+6*x+1) / (x+1) (**is** - ≤ ?rhs)
proof −
have sqrt x ≤ sqrtu x 1
by (metis assms gen-sqrt-upper)
also have ... = ?rhs
by (simp add: divide-simps) algebra
finally show ?thesis **.**
qed

lemma sqrtu-2-eq:
sqrtu x 2 = (1/8)*(x^4 + 28*x^3 + 70*x^2 + 28*x + 1) / ((x + 1)*(x^2 + 6*x + 1))
by (simp add: sqrtu-numeral divide-simps) algebra

lemma sqrt-upper-bound-2:
assumes x ≥ 0
shows sqrt x ≤ (1/8)*(x^4 + 28*x^3 + 70*x^2 + 28*x + 1) / ((x + 1)*(x^2 + 6*x + 1))
by (metis assms gen-sqrt-upper sqrtu-2-eq)

lemma sqrtu-4-eq:
x ≥ 0 ⟹
sqrtu x 4 = (1/32)*(225792840*x^6+64512240*x^5+601080390*x^8+471435600*x^7+496*x+1+35960
/ ((x+1)*(x^2+6*x+1)*(x^4+28*x^3+70*x^2+28*x+1)*(1820*x^6+8008*x^5+x^8+120*x^7+

by (*simp add*: *sqrtu-numeral divide-simps add-nonneg-eq-0-iff*) *algebra*

lemma *sqrt-upper-bound-4*:
assumes *x* ≥ *0*
shows *sqrt x* ≤ (*1*/*32*)∗(*225792840*∗*x*^*6*+*64512240*∗*x*^*5*+*601080390*∗*x*^*8*+*471435600*∗*x*^*7*+*496*∗*x*+*1*+*35*
/ ((*x*+*1*)∗(*x*^*2*+*6*∗*x*+*1*)∗(*x*^*4*+*28*∗*x*^*3*+*70*∗*x*^*2*+*28*∗*x*+*1*)∗(*1820*∗*x*^*6*+*8008*∗*x*^*5*+*x*^*8*+*120*∗*x*^*7*+
by (*metis assms gen-sqrt-upper sqrtu-4-eq*)

lemma *gen-sqrt-upper-scaled*:
assumes *0* ≤ *x* *0* < *u*
shows *sqrt x* ≤ *sqrtu* (*x*∗*u*^*2*) *n* / *u*
proof −
have *sqrt x* = *sqrt x* ∗ *sqrt* (*u*^*2*) / *u*
using *assms*
by *simp*
also have ... = *sqrt* (*x*∗*u*^*2*) / *u*
by (*metis real-sqrt-mult*)
also have ... ≤ *sqrtu* (*x*∗*u*^*2*) *n* / *u*
using *assms*
by (*simp add*: *divide-simps*) (*metis gen-sqrt-upper zero-le-mult-iff zero-le-power2*)
finally show *?thesis* **.**
qed

lemma *sqrt-upper-bound-2-small*:
assumes *0* ≤ *x*
shows *sqrt x* ≤ (*1*/*32*)∗(*65536*∗*x*^*4* + *114688*∗*x*^*3* + *17920*∗*x*^*2* + *448*∗*x* + *1*) / ((*16*∗*x* + *1*)∗(*256*∗*x*^*2* + *96*∗*x* + *1*))
apply (*rule order-trans* [*OF gen-sqrt-upper-scaled* [*of x 4 2*] *eq-refl*])
using *assms*
apply (*auto simp*: *sqrtu-2-eq*)
apply (*simp add*: *divide-simps*)
apply *algebra*
done

lemma *sqrt-upper-bound-2-large*:
assumes *0* ≤ *x*
shows *sqrt x* ≤ (*1*/*32*)∗(*65536* + *114688*∗*x* + *17920*∗*x*^*2* + *448*∗*x*^*3* + *x*^*4*) / ((*x* + *16*)∗(*256* + *96*∗*x* + *x*^*2*))
apply (*rule order-trans* [*OF gen-sqrt-upper-scaled* [*of x 1*/*4 2*] *eq-refl*])
using *assms*
apply (*auto simp*: *sqrtu-2-eq*)
apply (*simp add*: *divide-simps*)
apply *algebra*
done

6.2 Lower bounds

lemma *sqrt-lower-bound-id*:
assumes *0* ≤ *x* *x* ≤ *1*

shows *x* ≤ *sqrt x*
proof −
have *x^2* ≤ *x* **using** *assms*
by (*metis one-le-numeral power-decreasing power-one-right*)
then show *?thesis*
by (*metis real-le-rsqrt*)
qed

definition *sqrtl* :: [*real*,*nat*] ⇒ *real* **where**
sqrtl x n = *x*/*sqrtu x n*

lemma *sqrtl-lower*: *0* ≤ *x* ⟹ *sqrtl x n* ^ *2* ≤ *x*
unfolding *sqrtl-def* **using** *sqrtu-upper* [*of x n*]
by (*auto simp*: *power2-eq-square divide-simps mult-left-mono*)

theorem *gen-sqrt-lower*: *0* ≤ *x* ⟹ *sqrtl x n* ≤ *sqrt x*
using *real-sqrt-le-mono* [*OF sqrtl-lower* [*of x n*]]
by *auto*

lemma *sqrt-lower-bound-0*:
assumes *x* ≥ *0* **shows** *2*∗*x*/(*x*+*1*) ≤ *sqrt x* (**is** *?lhs* ≤ -)
proof −
have *?lhs* = *sqrtl x 0*
by (*simp add*: *sqrtl-def*)
also have ... ≤ *sqrt x*
by (*metis assms gen-sqrt-lower*)
finally show *?thesis* **.**
qed

lemma *sqrt-lower-bound-1*:
assumes *x* ≥ *0* **shows** *4*∗*x*∗(*x*+*1*) / (*x^2*+*6*∗*x*+*1*) ≤ *sqrt x* (**is** *?lhs* ≤ -)
proof −
have *?lhs* = *sqrtl x 1* **using** *assms*
by (*simp add*: *sqrtl-def power-eq-if algebra-simps divide-simps*)
also have ... ≤ *sqrt x*
by (*metis assms gen-sqrt-lower*)
finally show *?thesis* **.**
qed

lemma *sqrtl-2-eq*: *sqrtl x 2* =
8∗*x*∗(*x* + *1*)∗(*x^2* + *6*∗*x* + *1*) / (*x^4* + *28*∗*x^3* + *70*∗*x^2* + *28*∗*x* + *1*)
using *sqrtu-gt-0* [*of x 2*]
by (*simp add*: *sqrtl-def sqrtu-2-eq*)

lemma *sqrt-lower-bound-2*:
assumes *x* ≥ *0*
shows *8*∗*x*∗(*x* + *1*)∗(*x^2* + *6*∗*x* + *1*) / (*x^4* + *28*∗*x^3* + *70*∗*x^2* + *28*∗*x* +
1) ≤ *sqrt x*
by (*metis assms sqrtl-2-eq gen-sqrt-lower*)

lemma *sqrtl-4-eq*: $x \geq 0 \Longrightarrow$
sqrtl x 4
= (*32**x*(x+1)*(x^2+6*x+1)*(x^4+28*x^3+70*x^2+28*x+1)*(1820*x^6+8008*x^5+x^8+120*x^7+*
/ (*225792840*x^6+64512240*x^5+601080390*x^8+471435600*x^7+496*x+1+35960*x^2+906192*x^3*
using *sqrtu-gt-0* [*of x 4*]
by (*simp add*: *sqrtl-def sqrtu-4-eq*)

lemma *sqrt-lower-bound-4*:
assumes $x \geq 0$
shows (*32*x*(x+1)*(x^2+6*x+1)*(x^4+28*x^3+70*x^2+28*x+1)*(1820*x^6+8008*x^5+x^8+120**
/ (*225792840*x^6+64512240*x^5+601080390*x^8+471435600*x^7+496*x+1+35960*x^2+906192*
≤ *sqrt x*
by (*metis assms sqrtl-4-eq gen-sqrt-lower*)

lemma *gen-sqrt-lower-scaled*:
assumes $0 \leq x$ $0 < u$
shows *sqrtl* (*x*u^2*) *n* / *u* ≤ *sqrt x*
proof −
have *sqrtl* (*x*u^2*) *n* / *u* ≤ *sqrt* (*x*u^2*) / *u*
using *assms*
by (*simp add*: *divide-simps*) (*metis gen-sqrt-lower zero-le-mult-iff zero-le-power2*)
also have ... = *sqrt x* ∗ *sqrt* (*u^2*) / *u*
by (*metis real-sqrt-mult*)
also have ... = *sqrt x*
using *assms*
by *simp*
finally show *?thesis* **.**
qed

lemma *sqrt-lower-bound-2-small*:
assumes $0 \leq x$
shows *32*x*(16*x + 1)*(256*x^2 + 96*x + 1) / (65536*x^4 + 114688*x^3 + 17920*x^2 + 448*x + 1)* ≤ *sqrt x*
apply (*rule order-trans* [*OF eq-refl gen-sqrt-lower-scaled* [*of x 4 2*]])
using *assms*
apply (*auto simp*: *sqrtl-2-eq*)
apply (*simp add*: *divide-simps*)
apply *algebra*
done

lemma *sqrt-lower-bound-2-large*:
assumes $0 \leq x$
shows *32*x*(x + 16)*(x^2 + 96*x + 256) / (x^4 + 448*x^3 + 17920*x^2 + 114688*x + 65536)* ≤ *sqrt x*
apply (*rule order-trans* [*OF eq-refl gen-sqrt-lower-scaled* [*of x 1/4 2*]])
using *assms*
apply (*auto simp*: *sqrtl-2-eq*)
apply (*simp add*: *divide-simps*)

done

end

Bibliography

[1] B. Akbarpour and L. Paulson. MetiTarski: An automatic theorem prover for real-valued special functions. 44(3):175–205, Mar. 2010.