

# The Akra–Bazzi theorem and the Master theorem

Manuel Eberl

February 20, 2024

## Abstract

This article contains a formalisation of the Akra–Bazzi method [1] based on a proof by Leighton [2]. It is a generalisation of the well-known Master Theorem for analysing the complexity of Divide & Conquer algorithms. We also include a generalised version of the Master theorem based on the Akra–Bazzi theorem, which is easier to apply than the Akra–Bazzi theorem itself.

Some proof methods that facilitate applying the Master theorem are also included. For a more detailed explanation of the formalisation and the proof methods, see the accompanying paper (publication forthcoming).

## Contents

<b>1 Auxiliary lemmas</b>	<b>2</b>
<b>2 Asymptotic bounds</b>	<b>5</b>
<b>3 The continuous Akra-Bazzi theorem</b>	<b>8</b>
<b>4 The discrete Akra-Bazzi theorem</b>	<b>18</b>
<b>5 The Master theorem</b>	<b>25</b>
<b>6 Evaluating expressions with rational numerals</b>	<b>28</b>
<b>7 The proof methods</b>	<b>31</b>
7.1 Master theorem and termination . . . . .	31
<b>8 Examples</b>	<b>38</b>
8.1 Merge sort . . . . .	39
8.2 Karatsuba multiplication . . . . .	39
8.3 Strassen matrix multiplication . . . . .	39
8.4 Deterministic select . . . . .	40
8.5 Decreasing function . . . . .	40

8.6	Example taken from Drmota and Szpakowski . . . . .	40
8.7	Transcendental exponents . . . . .	41
8.8	Functions in locale contexts . . . . .	41
8.9	Non-curried functions . . . . .	42
8.10	Ham-sandwich trees . . . . .	42

## 1 Auxiliary lemmas

**theory** Akra-Bazzi-Library

**imports**

Complex-Main

Landau-Symbols.Landau-More

Pure-ex.Guess

**begin**

**lemma** ln-mono:  $0 < x \implies 0 < y \implies x \leq y \implies \ln(x::real) \leq \ln y$   
 $\langle proof \rangle$

**lemma** ln-mono-strict:  $0 < x \implies 0 < y \implies x < y \implies \ln(x::real) < \ln y$   
 $\langle proof \rangle$

**declare** DERIV-power[THEN DERIV-chain2, derivative-intros]

**lemma** sum-pos':  
**assumes** finite I  
**assumes**  $\exists x \in I. f x > (0 :: - :: linordered-ab-group-add)$   
**assumes**  $\bigwedge x. x \in I \implies f x \geq 0$   
**shows**  $\sum f I > 0$   
 $\langle proof \rangle$

**lemma** min-mult-left:  
**assumes**  $(x::real) > 0$   
**shows**  $x * \min y z = \min(x*y) (x*z)$   
 $\langle proof \rangle$

**lemma** max-mult-left:  
**assumes**  $(x::real) > 0$   
**shows**  $x * \max y z = \max(x*y) (x*z)$   
 $\langle proof \rangle$

**lemma** DERIV-nonneg-imp-mono:  
**assumes**  $\bigwedge t. t \in \{x..y\} \implies (f \text{ has-field-derivative } f' t) \text{ (at } t\text{)}$   
**assumes**  $\bigwedge t. t \in \{x..y\} \implies f' t \geq 0$   
**assumes**  $(x::real) \leq y$   
**shows**  $(f x :: real) \leq f y$

$\langle proof \rangle$

**lemma** *eventually-conjE*: *eventually* ( $\lambda x. P x \wedge Q x$ )  $F \implies (\text{eventually } P F \implies \text{eventually } Q F \implies R) \implies R$   
 $\langle proof \rangle$

**lemma** *real-natfloor-nat*:  $x \in \mathbb{N} \implies \text{real}(\text{nat}\lfloor x \rfloor) = x$   $\langle proof \rangle$

**lemma** *eventually-natfloor*:  
  **assumes** *eventually*  $P$  (*at-top* :: *nat filter*)  
  **shows** *eventually* ( $\lambda x. P(\text{nat}\lfloor x \rfloor)$ ) (*at-top* :: *real filter*)  
 $\langle proof \rangle$

**lemma** *tendsto-0-smallo-1*:  $f \in o(\lambda x. 1 :: \text{real}) \implies (f \xrightarrow{} 0) \text{ at-top}$   
 $\langle proof \rangle$

**lemma** *smallo-1-tendsto-0*:  $(f \xrightarrow{} 0) \text{ at-top} \implies f \in o(\lambda x. 1 :: \text{real})$   
 $\langle proof \rangle$

**lemma** *filterlim-at-top-smallomega-1*:  
  **assumes**  $f \in \omega[F](\lambda x. 1 :: \text{real})$  *eventually* ( $\lambda x. f x > 0$ )  $F$   
  **shows** *filterlim*  $f$  *at-top*  $F$   
 $\langle proof \rangle$

**lemma** *smallo-imp-abs-less-real*:  
  **assumes**  $f \in o[F](g)$  *eventually* ( $\lambda x. g x > (0 :: \text{real})$ )  $F$   
  **shows** *eventually* ( $\lambda x. |f x| < g x$ )  $F$   
 $\langle proof \rangle$

**lemma** *smallo-imp-less-real*:  
  **assumes**  $f \in o[F](g)$  *eventually* ( $\lambda x. g x > (0 :: \text{real})$ )  $F$   
  **shows** *eventually* ( $\lambda x. f x < g x$ )  $F$   
 $\langle proof \rangle$

**lemma** *smallo-imp-le-real*:  
  **assumes**  $f \in o[F](g)$  *eventually* ( $\lambda x. g x \geq (0 :: \text{real})$ )  $F$   
  **shows** *eventually* ( $\lambda x. f x \leq g x$ )  $F$   
 $\langle proof \rangle$

**lemma** *filterlim-at-right*:  
  *filterlim*  $f$  (*at-right*  $a$ )  $F \longleftrightarrow \text{eventually } (\lambda x. f x > a) F \wedge \text{filterlim } f(\text{nhds } a) F$   
 $\langle proof \rangle$

**lemma** *one-plus-x-powr-approx-ex*:  
  **assumes**  $x: \text{abs}(x :: \text{real}) \leq 1/2$   
  **obtains**  $t$  **where**  $\text{abs } t < 1/2$   $(1 + x)^{\text{powr } p} =$   
     $1 + p * x + p * (p - 1) * (1 + t)^{\text{powr } (p - 2)} / 2 * x^2$

$\langle proof \rangle$

**lemma** *powr-lower-bound*:  $\llbracket (l::real) > 0; l \leq x; x \leq u \rrbracket \implies \min(l \text{ powr } z) \leq x \text{ powr } z$   
 $\langle proof \rangle$

**lemma** *powr-upper-bound*:  $\llbracket (l::real) > 0; l \leq x; x \leq u \rrbracket \implies \max(l \text{ powr } z) \geq x \text{ powr } z$   
 $\langle proof \rangle$

**lemma** *one-plus-x-powr-Taylor2*:  
  **obtains**  $k$  **where**  $\bigwedge x. \text{abs}(x::real) \leq 1/2 \implies \text{abs}((1+x) \text{ powr } p - 1 - p*x) \leq k*x^2$   
 $\langle proof \rangle$

**lemma** *one-plus-x-powr-Taylor2-bigo*:  
  **assumes**  $\lim(f \longrightarrow 0) F$   
  **shows**  $(\lambda x. (1 + f x) \text{ powr } (p::real) - 1 - p * f x) \in O[F](\lambda x. f x \wedge 2)$   
 $\langle proof \rangle$

**lemma** *one-plus-x-powr-Taylor1-bigo*:  
  **assumes**  $\lim(f \longrightarrow 0) F$   
  **shows**  $(\lambda x. (1 + f x) \text{ powr } (p::real) - 1) \in O[F](\lambda x. f x)$   
 $\langle proof \rangle$

**lemma** *x-times-x-minus-1-nonneg*:  $x \leq 0 \vee x \geq 1 \implies (x::linordered-idom) * (x - 1) \geq 0$   
 $\langle proof \rangle$

**lemma** *x-times-x-minus-1-nonpos*:  $x \geq 0 \implies x \leq 1 \implies (x::linordered-idom) * (x - 1) \leq 0$   
 $\langle proof \rangle$

**lemma** *powr-mono'*:  
  **assumes**  $(x::real) > 0 x \leq 1 a \leq b$   
  **shows**  $x \text{ powr } b \leq x \text{ powr } a$   
 $\langle proof \rangle$

**lemma** *powr-less-mono'*:  
  **assumes**  $(x::real) > 0 x < 1 a < b$   
  **shows**  $x \text{ powr } b < x \text{ powr } a$   
 $\langle proof \rangle$

**lemma** *real-powr-at-bot*:  
  **assumes**  $(a::real) > 1$   
  **shows**  $((\lambda x. a \text{ powr } x) \longrightarrow 0) \text{ at-bot}$   
 $\langle proof \rangle$

**lemma** *real-powr-at-bot-neg*:

```

assumes (a::real) > 0 a < 1
shows filterlim (λx. a powr x) at-top at-bot
⟨proof⟩

lemma real-powr-at-top-neg:
assumes (a::real) > 0 a < 1
shows ((λx. a powr x) —→ 0) at-top
⟨proof⟩

lemma eventually-nat-real:
assumes eventually P (at-top :: real filter)
shows eventually (λx. P (real x)) (at-top :: nat filter)
⟨proof⟩

end

```

## 2 Asymptotic bounds

```

theory Akra-Bazzi-Asymptotics
imports
  Complex-Main
  Akra-Bazzi-Library
  HOL-Library.Landau-Symbols
begin

locale akra-bazzi-asymptotics-bep =
  fixes b e p hb :: real
  assumes bep: b > 0 b < 1 e > 0 hb > 0
begin

context
begin

Functions that are negligible w.r.t.  $\ln(b*x)$   $\text{powr}(e/2 + 1)$ .
private abbreviation (input) negl :: (real ⇒ real) ⇒ bool where
  negl f ≡ f ∈ o(λx. ln(b*x) powr(-(e/2 + 1)))

private lemma neglD: negl f ⇒ c > 0 ⇒ eventually (λx. |f x| ≤ c / ln(b*x)
  powr(e/2+1)) at-top
  ⟨proof⟩ lemma negl-mult: negl f ⇒ negl g ⇒ negl (λx. f x * g x)
  ⟨proof⟩ lemma ev4:
    assumes g: negl g
    shows eventually (λx. ln(b*x) powr(-e/2) - ln x powr(-e/2) ≥ g x) at-top
  ⟨proof⟩ lemma ev1:
    negl (λx. (1 + c * inverse b * ln x powr(-(1+e))) powr p - 1)
  ⟨proof⟩ lemma ev2-aux:
    defines f ≡ λx. (1 + 1/ln(b*x) * ln(1 + hb / b * ln x powr(-1-e))) powr
    (-e/2)
    obtains h where eventually (λx. f x ≥ 1 + h x) at-top h ∈ o(λx. 1 / ln x)

```

```

⟨proof⟩ lemma ev2:
  defines f ≡ λx. ln (b * x + hb * x / ln x powr (1 + e)) powr (-e/2)
  obtains h where
    negl h
    eventually (λx. f x ≥ ln (b * x) powr (-e/2) + h x) at-top
    eventually (λx. |ln (b * x) powr (-e/2) + h x| < 1) at-top
⟨proof⟩ lemma ev21:
  obtains g where
    negl g
    eventually (λx. 1 + ln (b * x + hb * x / ln x powr (1 + e)) powr (-e/2) ≥
      1 + ln (b * x) powr (-e/2) + g x) at-top
    eventually (λx. 1 + ln (b * x) powr (-e/2) + g x > 0) at-top
⟨proof⟩ lemma ev22:
  obtains g where
    negl g
    eventually (λx. 1 - ln (b * x + hb * x / ln x powr (1 + e)) powr (-e/2) ≤
      1 - ln (b * x) powr (-e/2) - g x) at-top
    eventually (λx. 1 - ln (b * x) powr (-e/2) - g x > 0) at-top
⟨proof⟩

lemma asymptotics1:
  shows eventually (λx.
    (1 + c * inverse b * ln x powr -(1+e)) powr p *
    (1 + ln (b * x + hb * x / ln x powr (1 + e)) powr (- e / 2)) ≥
    1 + (ln x powr (-e/2))) at-top
⟨proof⟩

lemma asymptotics2:
  shows eventually (λx.
    (1 + c * inverse b * ln x powr -(1+e)) powr p *
    (1 - ln (b * x + hb * x / ln x powr (1 + e)) powr (- e / 2)) ≤
    1 - (ln x powr (-e/2))) at-top
⟨proof⟩

lemma asymptotics3: eventually (λx. (1 + (ln x powr (-e/2))) / 2 ≤ 1) at-top
  (is eventually (λx. ?f x ≤ 1) -)
⟨proof⟩

lemma asymptotics4: eventually (λx. (1 - (ln x powr (-e/2))) * 2 ≥ 1) at-top
  (is eventually (λx. ?f x ≥ 1) -)
⟨proof⟩

lemma asymptotics5: eventually (λx. ln (b*x - hb*x*ln x powr -(1+e)) powr
  (-e/2) < 1) at-top
⟨proof⟩

lemma asymptotics6: eventually (λx. hb / ln x powr (1 + e) < b/2) at-top
  and asymptotics7: eventually (λx. hb / ln x powr (1 + e) < (1 - b) / 2) at-top

```

```

and asymptotics8: eventually ( $\lambda x. x*(1 - b - hb / \ln x \text{ powr } (1 + e)) > 1$ )
at-top
⟨proof⟩

end
end

definition akra-bazzi-asymptotic1 b hb e p x  $\longleftrightarrow$ 

$$(1 - hb * \text{inverse } b * \ln x \text{ powr } -(1+e)) \text{ powr } p * (1 + \ln(b*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$$


$$\geq 1 + (\ln x \text{ powr } (-e/2) :: \text{real})$$

definition akra-bazzi-asymptotic1' b hb e p x  $\longleftrightarrow$ 

$$(1 + hb * \text{inverse } b * \ln x \text{ powr } -(1+e)) \text{ powr } p * (1 + \ln(b*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$$


$$\geq 1 + (\ln x \text{ powr } (-e/2) :: \text{real})$$

definition akra-bazzi-asymptotic2 b hb e p x  $\longleftrightarrow$ 

$$(1 + hb * \text{inverse } b * \ln x \text{ powr } -(1+e)) \text{ powr } p * (1 - \ln(b*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$$


$$\leq 1 - \ln x \text{ powr } (-e/2 :: \text{real})$$

definition akra-bazzi-asymptotic2' b hb e p x  $\longleftrightarrow$ 

$$(1 - hb * \text{inverse } b * \ln x \text{ powr } -(1+e)) \text{ powr } p * (1 - \ln(b*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$$


$$\leq 1 - \ln x \text{ powr } (-e/2 :: \text{real})$$

definition akra-bazzi-asymptotic3 e x  $\longleftrightarrow$   $(1 + (\ln x \text{ powr } (-e/2))) / 2 \leq (1 :: \text{real})$ 
definition akra-bazzi-asymptotic4 e x  $\longleftrightarrow$   $(1 - (\ln x \text{ powr } (-e/2))) * 2 \geq (1 :: \text{real})$ 
definition akra-bazzi-asymptotic5 b hb e x  $\longleftrightarrow$ 

$$\ln(b*x - hb*x*\ln x \text{ powr } -(1+e)) \text{ powr } (-e/2 :: \text{real}) < 1$$


definition akra-bazzi-asymptotic6 b hb e x  $\longleftrightarrow$   $hb / \ln x \text{ powr } (1 + e :: \text{real}) < b/2$ 
definition akra-bazzi-asymptotic7 b hb e x  $\longleftrightarrow$   $hb / \ln x \text{ powr } (1 + e :: \text{real}) < (1 - b) / 2$ 
definition akra-bazzi-asymptotic8 b hb e x  $\longleftrightarrow$   $x*(1 - b - hb / \ln x \text{ powr } (1 + e :: \text{real})) > 1$ 

definition akra-bazzi-asymptotics b hb e p x  $\longleftrightarrow$ 
akra-bazzi-asymptotic1 b hb e p x  $\wedge$  akra-bazzi-asymptotic1' b hb e p x  $\wedge$ 
akra-bazzi-asymptotic2 b hb e p x  $\wedge$  akra-bazzi-asymptotic2' b hb e p x  $\wedge$ 
akra-bazzi-asymptotic3 e x  $\wedge$  akra-bazzi-asymptotic4 e x  $\wedge$  akra-bazzi-asymptotic5
b hb e x  $\wedge$ 
akra-bazzi-asymptotic6 b hb e x  $\wedge$  akra-bazzi-asymptotic7 b hb e x  $\wedge$ 
akra-bazzi-asymptotic8 b hb e x

lemmas akra-bazzi-asymptotic-defs =
akra-bazzi-asymptotic1-def akra-bazzi-asymptotic1'-def
akra-bazzi-asymptotic2-def akra-bazzi-asymptotic2'-def akra-bazzi-asymptotic3-def

```

```

akra-bazzi-asymptotic4-def akra-bazzi-asymptotic5-def akra-bazzi-asymptotic6-def
akra-bazzi-asymptotic7-def akra-bazzi-asymptotic8-def akra-bazzi-asymptotics-def

```

```

lemma akra-bazzi-asymptotics:
  assumes  $\bigwedge b. b \in \text{set } bs \implies b \in \{0 <.. < 1\}$ 
  assumes  $hb > 0 \text{ and } e > 0$ 
  shows eventually  $(\lambda x. \forall b \in \text{set } bs. \text{akra-bazzi-asymptotics } b \text{ } hb \text{ } e \text{ } p \text{ } x)$  at-top
  ⟨proof⟩
end

```

### 3 The continuous Akra-Bazzi theorem

**theory** Akra-Bazzi-Real

**imports**

Complex-Main

Akra-Bazzi-Asymptotics

**begin**

We want to be generic over the integral definition used; we fix some arbitrary notions of integrability and integral and assume just the properties we need. The user can then instantiate the theorems with any desired integral definition.

```

locale akra-bazzi-integral =
  fixes integrable ::  $(\text{real} \Rightarrow \text{real}) \Rightarrow \text{real} \Rightarrow \text{real} \Rightarrow \text{bool}$ 
  and integral ::  $(\text{real} \Rightarrow \text{real}) \Rightarrow \text{real} \Rightarrow \text{real} \Rightarrow \text{real}$ 
  assumes integrable-const:  $c \geq 0 \implies \text{integrable } (\lambda x. c) a b$ 
  and integral-const:  $c \geq 0 \implies a \leq b \implies \text{integral } (\lambda x. c) a b = (b - a) * c$ 
  and integrable-subinterval:
    integrable f a b  $\implies a \leq a' \implies b' \leq b \implies \text{integrable } f a' b'$ 
  and integral-le:
    integrable f a b  $\implies \text{integrable } g a b \implies (\lambda x. x \in \{a..b\} \implies f x \leq g x)$ 
   $\implies$ 
    integral f a b  $\leq$  integral g a b
  and integral-combine:
     $a \leq c \leq b \implies \text{integrable } f a b \implies \text{integral } f a c + \text{integral } f c b = \text{integral } f a b$ 
begin
  lemma integral-nonneg:
     $a \leq b \implies \text{integrable } f a b \implies (\lambda x. x \in \{a..b\} \implies f x \geq 0) \implies \text{integral } f a b \geq 0$ 
    ⟨proof⟩
end

```

**declare** sum.cong[fundef-cong]

**lemma** strict-mono-imp-ex1-real:

```

fixes f :: real  $\Rightarrow$  real
assumes lim-neg-inf: LIM x at-bot. f x  $:>$  at-top
assumes lim-inf: (f  $\longrightarrow$  z) at-top
assumes mono:  $\bigwedge a b. a < b \implies f b < f a$ 
assumes cont:  $\bigwedge x. \text{isCont } f x$ 
assumes y-greater-z: z < y
shows  $\exists !x. f x = y$ 
⟨proof⟩

```

The parameter  $p$  in the Akra-Bazzi theorem always exists and is unique.

```

definition akra-bazzi-exponent :: real list  $\Rightarrow$  real list  $\Rightarrow$  real where
akra-bazzi-exponent as bs  $\equiv$  (THE p. ( $\sum i < \text{length as}. \text{as}!i * \text{bs}!i \text{ powr } p$ ) = 1)

```

```

locale akra-bazzi-params =
fixes k :: nat and as bs :: real list
assumes length-as: length as = k
and length-bs: length bs = k
and k-not-0: k  $\neq$  0
and a-ge-0: a  $\in$  set as  $\implies$  a  $\geq$  0
and b-bounds: b  $\in$  set bs  $\implies$  b  $\in$  {0 < .. < 1}
begin

```

```

abbreviation p :: real where p  $\equiv$  akra-bazzi-exponent as bs

```

```

lemma p-def: p = (THE p. ( $\sum i < k. \text{as}!i * \text{bs}!i \text{ powr } p$ ) = 1)
⟨proof⟩

```

```

lemma b-pos: b  $\in$  set bs  $\implies$  b > 0 and b-less-1: b  $\in$  set bs  $\implies$  b < 1
⟨proof⟩

```

```

lemma as-nonempty [simp]: as  $\neq []$  and bs-nonempty [simp]: bs  $\neq []$ 
⟨proof⟩

```

```

lemma a-in-as[intro, simp]: i < k  $\implies$  as ! i  $\in$  set as
⟨proof⟩

```

```

lemma b-in-bs[intro, simp]: i < k  $\implies$  bs ! i  $\in$  set bs
⟨proof⟩

```

```

end

```

```

locale akra-bazzi-params-nonzero =
fixes k :: nat and as bs :: real list
assumes length-as: length as = k
and length-bs: length bs = k
and a-ge-0: a  $\in$  set as  $\implies$  a  $\geq$  0
and ex-a-pos:  $\exists a \in \text{set as}. a > 0$ 
and b-bounds: b  $\in$  set bs  $\implies$  b  $\in$  {0 < .. < 1}

```

```

begin

sublocale akra-bazzi-params k as bs
  ⟨proof⟩

lemma akra-bazzi-p-strict-mono:
  assumes x < y
  shows (∑ i < k. as!i * bs!i powr y) < (∑ i < k. as!i * bs!i powr x)
  ⟨proof⟩

lemma akra-bazzi-p-mono:
  assumes x ≤ y
  shows (∑ i < k. as!i * bs!i powr y) ≤ (∑ i < k. as!i * bs!i powr x)
  ⟨proof⟩

lemma akra-bazzi-p-unique:
  ∃!p. (∑ i < k. as!i * bs!i powr p) = 1
  ⟨proof⟩

lemma p-props: (∑ i < k. as!i * bs!i powr p) = 1
  and p-unique: (∑ i < k. as!i * bs!i powr p') = 1 ⇒ p = p'
  ⟨proof⟩

lemma p-greaterI: 1 < (∑ i < k. as!i * bs!i powr p') ⇒ p' < p
  ⟨proof⟩

lemma p-lessI: 1 > (∑ i < k. as!i * bs!i powr p') ⇒ p' > p
  ⟨proof⟩

lemma p-geI: 1 ≤ (∑ i < k. as!i * bs!i powr p') ⇒ p' ≤ p
  ⟨proof⟩

lemma p-leI: 1 ≥ (∑ i < k. as!i * bs!i powr p') ⇒ p' ≥ p
  ⟨proof⟩

lemma p-boundsI: (∑ i < k. as!i * bs!i powr x) ≤ 1 ∧ (∑ i < k. as!i * bs!i powr y)
  ≥ 1 ⇒ p ∈ {y..x}
  ⟨proof⟩

lemma p-boundsI': (∑ i < k. as!i * bs!i powr x) < 1 ∧ (∑ i < k. as!i * bs!i powr y)
  > 1 ⇒ p ∈ {y < .. < x}
  ⟨proof⟩

lemma p-nonneg: sum-list as ≥ 1 ⇒ p ≥ 0
  ⟨proof⟩

end

```

```

locale akra-bazzi-real-recursion =
  fixes as bs :: real list and hs :: (real  $\Rightarrow$  real) list and k :: nat and x0 x1 hb e p
  :: real
  assumes length-as: length as = k
  and length-bs: length bs = k
  and length-hs: length hs = k
  and k-not-0: k ≠ 0
  and a-ge-0: a ∈ set as  $\implies a \geq 0
  and b-bounds: b ∈ set bs  $\implies b \in \{0 <.. < 1\}

  and x0-ge-1: x0 ≥ 1
  and x0-le-x1: x0 ≤ x1
  and x1-ge: b ∈ set bs  $\implies x_1 \geq 2 * x_0 * \text{inverse } b

  and e-pos: e > 0
  and h-bounds: x ≥ x1  $\implies h \in \text{set hs} \implies |h x| \leq hb * x / \ln x \text{ powr } (1 + e)

  and asymptotics: x ≥ x0  $\implies b \in \text{set bs} \implies \text{akra-bazzi-asymptotics } b \text{ hb e p } x
begin

sublocale akra-bazzi-params k as bs
  ⟨proof⟩

lemma h-in-hs[intro, simp]: i < k  $\implies \text{hs ! } i \in \text{set hs}
  ⟨proof⟩

lemma x1-gt-1: x1 > 1
  ⟨proof⟩

lemma x1-ge-1: x1 ≥ 1 ⟨proof⟩

lemma x1-pos: x1 > 0 ⟨proof⟩

lemma bx-le-x: x ≥ 0  $\implies b \in \text{set bs} \implies b * x \leq x
  ⟨proof⟩

lemma x0-pos: x0 > 0 ⟨proof⟩

lemma
  assumes x ≥ x0 b ∈ set bs
  shows x0-hb-bound0: hb / ln x powr (1 + e) < b/2
  and x0-hb-bound1: hb / ln x powr (1 + e) < (1 - b) / 2
  and x0-hb-bound2: x * (1 - b - hb / ln x powr (1 + e)) > 1
  ⟨proof⟩

lemma step-diff:$$$$$$$ 
```

```

assumes  $i < k \ x \geq x_1$ 
shows  $bs ! i * x + (hs ! i) \ x + 1 < x$ 
⟨proof⟩

lemma step-le-x:  $i < k \implies x \geq x_1 \implies bs ! i * x + (hs ! i) \ x \leq x$ 
⟨proof⟩

lemma x0-hb-bound0':  $\bigwedge x \ b. \ x \geq x_0 \implies b \in set \ b \implies hb / ln \ x \ powr (1 + e) < b$ 
⟨proof⟩

lemma step-pos:
assumes  $i < k \ x \geq x_1$ 
shows  $bs ! i * x + (hs ! i) \ x > 0$ 
⟨proof⟩

lemma step-nonneg:  $i < k \implies x \geq x_1 \implies bs ! i * x + (hs ! i) \ x \geq 0$ 
⟨proof⟩

lemma step-nonneg':  $i < k \implies x \geq x_1 \implies bs ! i + (hs ! i) \ x / x \geq 0$ 
⟨proof⟩

lemma hb-nonneg:  $hb \geq 0$ 
⟨proof⟩

lemma x0-hb-bound3:
assumes  $x \geq x_1 \ i < k$ 
shows  $x - (bs ! i * x + (hs ! i) \ x) \leq x$ 
⟨proof⟩

lemma x0-hb-bound4:
assumes  $x \geq x_1 \ i < k$ 
shows  $(bs ! i + (hs ! i) \ x / x) > bs ! i / 2$ 
⟨proof⟩

lemma x0-hb-bound4':  $x \geq x_1 \implies i < k \implies (bs ! i + (hs ! i) \ x / x) \geq bs ! i / 2$ 
⟨proof⟩

lemma x0-hb-bound5:
assumes  $x \geq x_1 \ i < k$ 
shows  $(bs ! i + (hs ! i) \ x / x) \leq bs ! i * 3/2$ 
⟨proof⟩

lemma x0-hb-bound6:
assumes  $x \geq x_1 \ i < k$ 
shows  $x * ((1 - bs ! i) / 2) \leq x - (bs ! i * x + (hs ! i) \ x)$ 
⟨proof⟩

lemma x0-hb-bound7:

```

**assumes**  $x \geq x_1 i < k$   
**shows**  $bs!i*x + (hs!i) x > x_0$   
 $\langle proof \rangle$

**lemma**  $x0\text{-hb-bound7}': x \geq x_1 \implies i < k \implies bs!i*x + (hs!i) x > 1$   
 $\langle proof \rangle$

**lemma**  $x0\text{-hb-bound8}:$   
**assumes**  $x \geq x_1 i < k$   
**shows**  $bs!i*x - hb * x / \ln x \text{ powr } (1+e) > x_0$   
 $\langle proof \rangle$

**lemma**  $x0\text{-hb-bound8}':$   
**assumes**  $x \geq x_1 i < k$   
**shows**  $bs!i*x + hb * x / \ln x \text{ powr } (1+e) > x_0$   
 $\langle proof \rangle$

**lemma**  
**assumes**  $x \geq x_0$   
**shows**  $\text{asymptotics1}: i < k \implies 1 + \ln x \text{ powr } (-e/2) \leq$   
 $(1 - hb * \text{inverse}(bs!i) * \ln x \text{ powr } -(1+e)) \text{ powr } p *$   
 $(1 + \ln (bs!i*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$   
**and**  $\text{asymptotics2}: i < k \implies 1 - \ln x \text{ powr } (-e/2) \geq$   
 $(1 + hb * \text{inverse}(bs!i) * \ln x \text{ powr } -(1+e)) \text{ powr } p *$   
 $(1 - \ln (bs!i*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$   
**and**  $\text{asymptotics1}': i < k \implies 1 + \ln x \text{ powr } (-e/2) \leq$   
 $(1 + hb * \text{inverse}(bs!i) * \ln x \text{ powr } -(1+e)) \text{ powr } p *$   
 $(1 + \ln (bs!i*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$   
**and**  $\text{asymptotics2}': i < k \implies 1 - \ln x \text{ powr } (-e/2) \geq$   
 $(1 - hb * \text{inverse}(bs!i) * \ln x \text{ powr } -(1+e)) \text{ powr } p *$   
 $(1 - \ln (bs!i*x + hb*x/\ln x \text{ powr } (1+e)) \text{ powr } (-e/2))$   
**and**  $\text{asymptotics3}: (1 + \ln x \text{ powr } (-e/2)) / 2 \leq 1$   
**and**  $\text{asymptotics4}: (1 - \ln x \text{ powr } (-e/2)) * 2 \geq 1$   
**and**  $\text{asymptotics5}: i < k \implies \ln (bs!i*x - hb*x*\ln x \text{ powr } -(1+e)) \text{ powr } (-e/2) < 1$   
 $\langle proof \rangle$

**lemma**  $x0\text{-hb-bound9}:$   
**assumes**  $x \geq x_1 i < k$   
**shows**  $\ln (bs!i*x + (hs!i) x) \text{ powr } -(e/2) < 1$   
 $\langle proof \rangle$

**definition**  $\text{akra-bazzi-measure} :: \text{real} \Rightarrow \text{nat}$  **where**  
 $\text{akra-bazzi-measure } x = \text{nat} \lceil x \rceil$

**lemma**  $\text{akra-bazzi-measure-decreases}:$   
**assumes**  $x \geq x_1 i < k$

```

shows akra-bazzi-measure (bs!i*x + (hs!i) x) < akra-bazzi-measure x
⟨proof⟩

```

```

lemma akra-bazzi-induct[consumes 1, case-names base rec]:
assumes x ≥ x0
assumes base:  $\bigwedge x. x \geq x_0 \implies x \leq x_1 \implies P x$ 
assumes rec:  $\bigwedge x. x > x_1 \implies (\bigwedge i. i < k \implies P (bs!i*x + (hs!i) x)) \implies P x$ 
shows P x
⟨proof⟩

```

```
end
```

```

locale akra-bazzi-real = akra-bazzi-real-recursion +
fixes integrable integral
assumes integral: akra-bazzi-integral integrable integral
fixes f :: real ⇒ real
and g :: real ⇒ real
and C :: real
assumes p-props:  $(\sum i < k. as!i * bs!i powr p) = 1$ 
and f-base:  $x \geq x_0 \implies x \leq x_1 \implies f x \geq 0$ 
and f-rec:  $x > x_1 \implies f x = g x + (\sum i < k. as!i * f (bs!i * x + (hs!i)$ 
x))
and g-nonneg:  $x \geq x_0 \implies g x \geq 0$ 
and C-bound:  $b \in set bs \implies x \geq x_1 \implies C*x \leq b*x - hb*x/\ln x powr$ 
(1+e)
and g-integrable:  $x \geq x_0 \implies \text{integrable } (\lambda u. g u / u powr (p + 1)) x_0 x$ 
begin

```

```
interpretation akra-bazzi-integral integrable integral ⟨proof⟩
```

```

lemma akra-bazzi-integrable:
a ≥ x0 ⇒ a ≤ b ⇒ integrable ( $\lambda x. g x / x powr (p + 1)$ ) a b
⟨proof⟩

```

```

definition g-approx :: nat ⇒ real ⇒ real where
g-approx i x = x powr p * integral ( $\lambda u. g u / u powr (p + 1)$ ) (bs!i * x + (hs!i)
x) x

```

```

lemma f-nonneg: x ≥ x0 ⇒ f x ≥ 0
⟨proof⟩

```

```

definition f-approx :: real ⇒ real where
f-approx x = x powr p * (1 + integral ( $\lambda u. g u / u powr (p + 1)$ ) x0 x)

```

```

lemma f-approx-aux:
assumes x ≥ x0

```

**shows**  $1 + \text{integral}(\lambda u. g u / u \text{ powr } (p + 1)) x_0 x \geq 1$   
 $\langle \text{proof} \rangle$

**lemma**  $f\text{-approx-pos}$ :  $x \geq x_0 \implies f\text{-approx } x > 0$   
 $\langle \text{proof} \rangle$

**lemma**  $f\text{-approx-nonneg}$ :  $x \geq x_0 \implies f\text{-approx } x \geq 0$   
 $\langle \text{proof} \rangle$

**lemma**  $f\text{-approx-bounded-below}$ :  
**obtains**  $c$  **where**  $\bigwedge x. x \geq x_0 \implies x \leq x_1 \implies f\text{-approx } x \geq c$   $c > 0$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{asymptotics-aux}$ :  
**assumes**  $x \geq x_1 i < k$   
**assumes**  $s \equiv (\text{if } p \geq 0 \text{ then } 1 \text{ else } -1)$   
**shows**  $(bs!i*x - s*hb*x*\ln x \text{ powr } -(1+e)) \text{ powr } p \leq (bs!i*x + (hs!i) x) \text{ powr } p$  (**is**  $?thesis1$ )  
**and**  $(bs!i*x + (hs!i) x) \text{ powr } p \leq (bs!i*x + s*hb*x*\ln x \text{ powr } -(1+e)) \text{ powr } p$  (**is**  $?thesis2$ )  
 $\langle \text{proof} \rangle$

**lemma**  $\text{asymptotics1}'$ :  
**assumes**  $x \geq x_1 i < k$   
**shows**  $(bs!i*x) \text{ powr } p * (1 + \ln x \text{ powr } (-e/2)) \leq$   
 $(bs!i*x + (hs!i) x) \text{ powr } p * (1 + \ln (bs!i*x + (hs!i) x) \text{ powr } (-e/2))$   
 $\langle \text{proof} \rangle$

**lemma**  $\text{asymptotics2}'$ :  
**assumes**  $x \geq x_1 i < k$   
**shows**  $(bs!i*x + (hs!i) x) \text{ powr } p * (1 - \ln (bs!i*x + (hs!i) x) \text{ powr } (-e/2)) \leq$   
 $(bs!i*x) \text{ powr } p * (1 - \ln x \text{ powr } (-e/2))$   
 $\langle \text{proof} \rangle$

**lemma**  $Cx\text{-le-step}$ :  
**assumes**  $i < k x \geq x_1$   
**shows**  $C*x \leq bs!i*x + (hs!i) x$   
 $\langle \text{proof} \rangle$

**end**

**locale**  $\text{akra-bazzi-nat-to-real} = \text{akra-bazzi-real-recursion} +$   
**fixes**  $f :: \text{nat} \Rightarrow \text{real}$   
**and**  $g :: \text{real} \Rightarrow \text{real}$   
**assumes**  $f\text{-base}: \text{real } x \geq x_0 \implies \text{real } x \leq x_1 \implies f x \geq 0$

```

and       $f\text{-rec: } \text{real } x > x_1 \implies f x = g(\text{real } x) + (\sum i < k. \text{as!}i * f(\text{nat } \lfloor bs!i * x + (hs!i)(\text{real } x) \rfloor))$ 
and       $x_0\text{-int: } \text{real } (\text{nat } \lfloor x_0 \rfloor) = x_0$ 
begin

function  $f' :: \text{real} \Rightarrow \text{real}$  where
 $x \leq x_1 \implies f' x = f(\text{nat } \lfloor x \rfloor)$ 
 $| x > x_1 \implies f' x = g x + (\sum i < k. \text{as!}i * f' (bs!i * x + (hs!i) x))$ 
⟨proof⟩
termination ⟨proof⟩

lemma  $f'\text{-base: } x \geq x_0 \implies x \leq x_1 \implies f' x \geq 0$ 
⟨proof⟩

lemmas  $f'\text{-rec} = f'.\text{simps}(2)$ 

end

locale akra-bazzi-real-lower = akra-bazzi-real +
fixes  $fb2 \text{ } gb2 \text{ } c2 :: \text{real}$ 
assumes  $f\text{-base2: } x \geq x_0 \implies x \leq x_1 \implies f x \geq fb2$ 
and       $fb2\text{-pos: } fb2 > 0$ 
and       $g\text{-growth2: } \forall x \geq x_1. \forall u \in \{C*x..x\}. c2 * g x \geq g u$ 
and       $c2\text{-pos: } c2 > 0$ 
and       $g\text{-bounded: } x \geq x_0 \implies x \leq x_1 \implies g x \leq gb2$ 
begin

interpretation akra-bazzi-integral integrable integral ⟨proof⟩

lemma  $gb2\text{-nonneg: } gb2 \geq 0$  ⟨proof⟩

lemma  $g\text{-growth2}':$ 
assumes  $x \geq x_1 \ i < k \ u \in \{bs!i*x+(hs!i) x..x\}$ 
shows    $c2 * g x \geq g u$ 
⟨proof⟩

lemma  $g\text{-bounds2: }$ 
obtains  $c4$  where  $\bigwedge x. i. x \geq x_1 \implies i < k \implies g\text{-approx } i x \leq c4 * g x$   $c4 > 0$ 
⟨proof⟩

lemma  $f\text{-approx-bounded-above: }$ 
obtains  $c$  where  $\bigwedge x. x \geq x_0 \implies x \leq x_1 \implies f\text{-approx } x \leq c$   $c > 0$ 
⟨proof⟩

lemma  $f\text{-bounded-below: }$ 
assumes  $c' : c' > 0$ 
obtains  $c$  where  $\bigwedge x. x \geq x_0 \implies x \leq x_1 \implies 2 * (c * f\text{-approx } x) \leq f x$   $c \leq c'$ 

```

$c > 0$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-lower*:

**obtains**  $c5$  **where**  $\bigwedge x. x \geq x_0 \implies f x \geq c5 * f\text{-approx } x$   $c5 > 0$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-bigomega*:

$f \in \Omega(\lambda x. x \text{ powr } p * (1 + \text{integral } (\lambda u. g u / u \text{ powr } (p + 1)) x_0 x))$   
 $\langle proof \rangle$

**end**

**locale** *akra-bazzi-real-upper* = *akra-bazzi-real* +  
**fixes**  $fb1\ c1 :: real$   
**assumes** *f-base1*:  $x \geq x_0 \implies x \leq x_1 \implies f x \leq fb1$   
**and** *g-growth1*:  $\forall x \geq x_1. \forall u \in \{C*x..x\}. c1 * g x \leq g u$   
**and** *c1-pos*:  $c1 > 0$   
**begin**

**interpretation** *akra-bazzi-integral integrable integral*  $\langle proof \rangle$

**lemma** *g-growth1'*:

**assumes**  $x \geq x_1$   $i < k$   $u \in \{bs!i*x + (hs!i) x..x\}$   
**shows**  $c1 * g x \leq g u$   
 $\langle proof \rangle$

**lemma** *g-bounds1*:

**obtains**  $c3$  **where**  
 $\bigwedge x i. x \geq x_1 \implies i < k \implies c3 * g x \leq g\text{-approx } i x$   $c3 > 0$   
 $\langle proof \rangle$

**lemma** *f-bounded-above*:

**assumes**  $c': c' > 0$   
**obtains**  $c$  **where**  $\bigwedge x. x \geq x_0 \implies x \leq x_1 \implies f x \leq (1/2) * (c * f\text{-approx } x)$   $c \geq c'$   $c > 0$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-upper*:

**obtains**  $c6$  **where**  $\bigwedge x. x \geq x_0 \implies f x \leq c6 * f\text{-approx } x$   $c6 > 0$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-bigo*:

$f \in O(\lambda x. x \text{ powr } p * (1 + \text{integral } (\lambda u. g u / u \text{ powr } (p + 1)) x_0 x))$   
 $\langle proof \rangle$

```
end
```

```
end
```

## 4 The discrete Akra-Bazzi theorem

```
theory Akra-Bazzi
```

```
imports
```

```
Complex-Main
```

```
HOL-Library.Landau-Symbols
```

```
Akra-Bazzi-Real
```

```
begin
```

```
lemma ex-mono: ( $\exists x. P x \Rightarrow (\bigwedge x. P x \Rightarrow Q x) \Rightarrow (\exists x. Q x)$ )  $\langle proof \rangle$ 
```

```
lemma x-over-ln-mono:
```

```
assumes (e::real) > 0
```

```
assumes x > exp e
```

```
assumes x ≤ y
```

```
shows x / ln x powr e ≤ y / ln y powr e
```

```
 $\langle proof \rangle$ 
```

```
definition akra-bazzi-term :: nat ⇒ nat ⇒ real ⇒ (nat ⇒ nat) ⇒ bool where
```

```
akra-bazzi-term x0 x1 b t =
```

```
( $\exists e h. e > 0 \wedge (\lambda x. h x) \in O(\lambda x. real x / ln (real x) powr (1+e)) \wedge$ 
```

```
( $\forall x \geq x_1. t x \geq x_0 \wedge t x < x \wedge b*x + h x = real (t x))$ 
```

```
lemma akra-bazzi-termI [intro?]:
```

```
assumes e > 0 ( $\lambda x. h x) \in O(\lambda x. real x / ln (real x) powr (1+e))$ 
```

```
 $\bigwedge x. x \geq x_1 \Rightarrow t x \geq x_0 \bigwedge x. x \geq x_1 \Rightarrow t x < x$ 
```

```
 $\bigwedge x. x \geq x_1 \Rightarrow b*x + h x = real (t x)$ 
```

```
shows akra-bazzi-term x0 x1 b t
```

```
 $\langle proof \rangle$ 
```

```
lemma akra-bazzi-term-imp-less:
```

```
assumes akra-bazzi-term x0 x1 b t x ≥ x1
```

```
shows t x < x
```

```
 $\langle proof \rangle$ 
```

```
lemma akra-bazzi-term-imp-less':
```

```
assumes akra-bazzi-term x0 (Suc x1) b t x > x1
```

```
shows t x < x
```

```
 $\langle proof \rangle$ 
```

```
locale akra-bazzi-recursion =
```

```
fixes x0 x1 k :: nat and as bs :: real list and ts :: (nat ⇒ nat) list and f :: nat ⇒ real
```

```

assumes k-not-0:  $k \neq 0$ 
and   length-as:  $\text{length } as = k$ 
and   length-bs:  $\text{length } bs = k$ 
and   length-ts:  $\text{length } ts = k$ 
and   a-ge-0:  $a \in \text{set } as \implies a \geq 0$ 
and   b-bounds:  $b \in \text{set } bs \implies b \in \{0 <.. < k\}$ 
and   ts:  $i < \text{length } bs \implies \text{akra-bazzi-term } x_0 \ x_1 \ (bs!i) \ (ts!i)$ 
begin

sublocale akra-bazzi-params k as bs
⟨proof⟩

lemma ts-nonempty:  $ts \neq []$  ⟨proof⟩

definition e-hs :: real × (nat ⇒ real) list where
e-hs = (SOME (e,hs).
 $e > 0 \wedge \text{length } hs = k \wedge (\forall h \in \text{set } hs. (\lambda x. h x) \in O(\lambda x. \text{real } x / \ln(\text{real } x) \text{powr } (1+e))) \wedge$ 
 $(\forall t \in \text{set } ts. \forall x \geq x_1. t x \geq x_0 \wedge t x < x) \wedge$ 
 $(\forall i < k. \forall x \geq x_1. (bs!i)*x + (hs!i) x = \text{real } ((ts!i) x))$ 
)

definition e = (case e-hs of (e,-) ⇒ e)
definition hs = (case e-hs of (-,hs) ⇒ hs)

lemma filterlim-powr-zero-cong:
filterlim ( $\lambda x. P (x::\text{real}) (x \text{powr } (0::\text{real}))$ ) F at-top = filterlim ( $\lambda x. P x 1$ ) F
at-top
⟨proof⟩

lemma e-hs-aux:
 $0 < e \wedge \text{length } hs = k \wedge$ 
 $(\forall h \in \text{set } hs. (\lambda x. h x) \in O(\lambda x. \text{real } x / \ln(\text{real } x) \text{powr } (1 + e))) \wedge$ 
 $(\forall t \in \text{set } ts. \forall x \geq x_1. x_0 \leq t x \wedge t x < x) \wedge$ 
 $(\forall i < k. \forall x \geq x_1. (bs!i)*x + (hs!i) x = \text{real } ((ts!i) x))$ 
⟨proof⟩

lemma
e-pos:  $e > 0$  and length-hs:  $\text{length } hs = k$  and
hs-growth:  $\bigwedge h. h \in \text{set } hs \implies (\lambda x. h x) \in O(\lambda x. \text{real } x / \ln(\text{real } x) \text{powr } (1 + e))$  and
step-ge-x0:  $\bigwedge t. t \in \text{set } ts \implies x \geq x_1 \implies x_0 \leq t x$  and
step-less:  $\bigwedge t. t \in \text{set } ts \implies x \geq x_1 \implies t x < x$  and
decomp:  $\bigwedge i. i < k \implies x \geq x_1 \implies (bs!i)*x + (hs!i) x = \text{real } ((ts!i) x)$ 
⟨proof⟩

lemma h-in-hs [intro, simp]:  $i < k \implies hs ! i \in \text{set } hs$ 
⟨proof⟩

```

```

lemma t-in-ts [intro, simp]:  $i < k \implies ts ! i \in set ts$ 
  ⟨proof⟩

lemma x0-less-x1:  $x_0 < x_1$  and x0-le-x1:  $x_0 \leq x_1$ 
  ⟨proof⟩

lemma akra-bazzi-induct [consumes 1, case-names base rec]:
  assumes  $x \geq x_0$ 
  assumes base:  $\bigwedge x. x \geq x_0 \implies x < x_1 \implies P x$ 
  assumes rec:  $\bigwedge x. x \geq x_1 \implies (\bigwedge t. t \in set ts \implies P (t x)) \implies P x$ 
  shows  $P x$ 
  ⟨proof⟩

end

locale akra-bazzi-function = akra-bazzi-recursion +
  fixes integrable integral
  assumes integral: akra-bazzi-integral integrable integral
  fixes g :: nat ⇒ real
  assumes f-nonneg-base:  $x \geq x_0 \implies x < x_1 \implies f x \geq 0$ 
  and f-rec:  $x \geq x_1 \implies f x = g x + (\sum i < k. as!i * f ((ts!i) x))$ 
  and g-nonneg:  $x \geq x_1 \implies g x \geq 0$ 
  and ex-pos-a:  $\exists a \in set as. a > 0$ 
begin

lemma ex-pos-a':  $\exists i < k. as!i > 0$ 
  ⟨proof⟩

sublocale akra-bazzi-params-nonzero
  ⟨proof⟩

definition g-real :: real ⇒ real where g-real x = g (nat ⌊x⌋)

lemma g-real-real[simp]: g-real (real x) = g x ⟨proof⟩

lemma f-nonneg:  $x \geq x_0 \implies f x \geq 0$ 
  ⟨proof⟩

definition hs' = map (λ h x. h (nat ⌊x::real⌋)) hs

lemma length-hs': length hs' = k ⟨proof⟩

lemma hs'-real:  $i < k \implies (hs'!i) (real x) = (hs!i) x$ 
  ⟨proof⟩

lemma h-bound:
  obtains hb where hb > 0 and

```

*eventually* ( $\lambda x. \forall h \in set hs'. |h x| \leq hb * x / ln x powr (1 + e)$ ) *at-top*  
*(proof)*

**lemma** *C-bound*:

**assumes**  $\bigwedge b. b \in set bs \implies C < b \cdot hb > 0$

**shows** *eventually* ( $\lambda x::real. \forall b \in set bs. C * x \leq b * x - hb * x / ln x powr (1 + e)$ )

*at-top*

*(proof)*

**end**

**locale** *akra-bazzi-lower* = *akra-bazzi-function* +

**fixes**  $g' :: real \Rightarrow real$

**assumes** *f-pos*: *eventually* ( $\lambda x. f x > 0$ ) *at-top*

**and** *g-growth2*:  $\exists C c2. c2 > 0 \wedge C < Min (set bs) \wedge$

*eventually* ( $\lambda x. \forall u \in \{C * x..x\}. g' u \leq c2 * g' x$ ) *at-top*

**and** *g'-integrable*:  $\exists a. \forall b \geq a. integrable (\lambda u. g' u / u powr (p + 1)) a b$

**and** *g'-bounded*: *eventually* ( $\lambda a::real. (\forall b > a. \exists c. \forall x \in \{a..b\}. g' x \leq c)$ ) *at-top*

**and** *g-bigomega*:  $g \in \Omega(\lambda x. g' (real x))$

**and** *g'-nonneg*: *eventually* ( $\lambda x. g' x \geq 0$ ) *at-top*

**begin**

**definition**  $gc2 \equiv SOME gc2. gc2 > 0 \wedge$  *eventually* ( $\lambda x. g x \geq gc2 * g' (real x)$ )  
*at-top*

**lemma**  $gc2: gc2 > 0$  *eventually* ( $\lambda x. g x \geq gc2 * g' (real x)$ ) *at-top*  
*(proof)*

**definition**  $gx0 \equiv max x_1 (SOME gx0. \forall x \geq gx0. g x \geq gc2 * g' (real x) \wedge f x > 0 \wedge g' (real x) \geq 0)$

**definition**  $gx1 \equiv max gx0 (SOME gx1. \forall x \geq gx1. \forall i < k. (ts!i) x \geq gx0)$

**lemma** *gx0*:

**assumes**  $x \geq gx0$

**shows**  $g x \geq gc2 * g' (real x) \wedge f x > 0 \wedge g' (real x) \geq 0$

*(proof)*

**lemma** *gx1*:

**assumes**  $x \geq gx1$

**shows**  $(ts!i) x \geq gx0$

*(proof)*

**lemma** *gx0-ge-x1*:  $gx0 \geq x_1$  *(proof)*

**lemma** *gx0-le-gx1*:  $gx0 \leq gx1$  *(proof)*

**function**  $f2' :: nat \Rightarrow real$  **where**

$x < gx1 \implies f2' x = max 0 (f x / gc2)$

|  $x \geq gx1 \implies f2' x = g' (\text{real } x) + (\sum i < k. as!i * f2' ((ts!i) x))$   
 $\langle proof \rangle$

**termination**  $\langle proof \rangle$

**lemma**  $f2'$ -nonneg:  $x \geq gx0 \implies f2' x \geq 0$   
 $\langle proof \rangle$

**lemma**  $f2'$ -le-f:  $x \geq x_0 \implies gc2 * f2' x \leq f x$   
 $\langle proof \rangle$

**lemma**  $f2'$ -pos: eventually  $(\lambda x. f2' x > 0)$  at-top  
 $\langle proof \rangle$

**lemma** bigomega-f-aux:

**obtains**  $a$  **where**  $a \geq A \forall a' \geq a. a' \in \mathbb{N} \longrightarrow$   
 $f \in \Omega(\lambda x. x \text{ powr } p * (1 + \text{integral } (\lambda u. g' u / u \text{ powr } (p + 1)) a' x))$   
 $\langle proof \rangle$

**lemma** bigomega-f:

**obtains**  $a$  **where**  $a \geq A$   $f \in \Omega(\lambda x. x \text{ powr } p * (1 + \text{integral } (\lambda u. g' u / u \text{ powr } (p + 1)) a x))$   
 $\langle proof \rangle$

end

**locale** akra-bazzi-upper = akra-bazzi-function +  
**fixes**  $g' :: \text{real} \Rightarrow \text{real}$   
**assumes**  $g'$ -integrable:  $\exists a. \forall b \geq a. \text{integrable } (\lambda u. g' u / u \text{ powr } (p + 1)) a b$   
**and**  $g$ -growth1:  $\exists C c1. c1 > 0 \wedge C < \text{Min } (\text{set } bs) \wedge$   
 $\text{eventually } (\lambda x. \forall u \in \{C * x .. x\}. g' u \geq c1 * g' x)$  at-top  
**and**  $g$ -bigo:  $g \in O(g')$   
**and**  $g'$ -nonneg: eventually  $(\lambda x. g' x \geq 0)$  at-top  
**begin**

**definition**  $gc1 \equiv \text{SOME } gc1. gc1 > 0 \wedge \text{eventually } (\lambda x. g x \leq gc1 * g' (\text{real } x))$   
at-top

**lemma**  $gc1 > 0$  eventually  $(\lambda x. g x \leq gc1 * g' (\text{real } x))$  at-top  
 $\langle proof \rangle$

**definition**  $gx3 \equiv \max x_1 (\text{SOME } gx0. \forall x \geq gx0. g x \leq gc1 * g' (\text{real } x))$

**lemma**  $gx3$ :

**assumes**  $x \geq gx3$   
**shows**  $g x \leq gc1 * g' (\text{real } x)$

```

⟨proof⟩

lemma gx3-ge-x1: gx3 ≥ x1 ⟨proof⟩

function f' :: nat ⇒ real where
  x < gx3 ⇒ f' x = max 0 (f x / gc1)
  | x ≥ gx3 ⇒ f' x = g' (real x) + (∑ i<k. as!i*f' ((ts!i) x))
  ⟨proof⟩
termination ⟨proof⟩

lemma f'-ge-f: x ≥ x0 ⇒ gc1 * f' x ≥ f x
⟨proof⟩

lemma bigo-f-aux:
  obtains a where a ≥ A ∀ a'≥a. a' ∈ ℙ →
    f ∈ O(λx. x powr p *(1 + integral (λu. g' u / u powr (p + 1)) a' x))
  ⟨proof⟩

lemma bigo-f:
  obtains a where a > A f ∈ O(λx. x powr p *(1 + integral (λu. g' u / u powr (p + 1)) a x))
  ⟨proof⟩

end

locale akra-bazzi = akra-bazzi-function +
  fixes g' :: real ⇒ real
  assumes f-pos: eventually (λx. f x > 0) at-top
  and g'-nonneg: eventually (λx. g' x ≥ 0) at-top
  assumes g'-integrable: ∃ a. ∀ b≥a. integrable (λu. g' u / u powr (p + 1)) a b
  and g-growth1: ∃ C c1. c1 > 0 ∧ C < Min (set bs) ∧
    eventually (λx. ∀ u∈{C*x..x}. g' u ≥ c1 * g' x) at-top
  and g-growth2: ∃ C c2. c2 > 0 ∧ C < Min (set bs) ∧
    eventually (λx. ∀ u∈{C*x..x}. g' u ≤ c2 * g' x) at-top
  and g-bounded: eventually (λa::real. (∀ b>a. ∃ c. ∀ x∈{a..b}. g' x ≤ c)) at-top
  and g-bigtheta: g ∈ Θ(g')
begin

sublocale akra-bazzi-lower ⟨proof⟩
sublocale akra-bazzi-upper ⟨proof⟩

lemma bigtheta-f:
  obtains a where a > A f ∈ Θ(λx. x powr p *(1 + integral (λu. g' u / u powr (p + 1)) a x))
  ⟨proof⟩

end

```

**named-theorems** *akra-bazzi-term-intros introduction rules for Akra–Bazzi terms*

**lemma** *akra-bazzi-term-floor-add* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 + c \ c < (1 - b) * real \ x_1 \ x_1 > 0$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lfloor b * real \ x + c \rfloor)$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-term-floor-add'* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 + real \ c \ real \ c < (1 - b) * real \ x_1 \ x_1 > 0$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lfloor b * real \ x \rfloor + c)$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-term-floor-subtract* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 - c \ 0 < c + (1 - b) * real \ x_1 \ x_1 > 0$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lfloor b * real \ x - c \rfloor)$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-term-floor-subtract'* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 - real \ c \ 0 < real \ c + (1 - b) * real \ x_1 \ x_1 > 0$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lfloor b * real \ x \rfloor - c)$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-term-floor* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 \ 0 < (1 - b) * real \ x_1 \ x_1 > 0$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lfloor b * real \ x \rfloor)$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-term-ceiling-add* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 + c \ c + 1 \leq (1 - b) * x_1$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lceil b * real \ x + c \rceil)$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-term-ceiling-add'* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 + real \ c \ real \ c + 1 \leq (1 - b) * x_1$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lceil b * real \ x \rceil + c)$   
 $\langle proof \rangle$

**lemma** *akra-bazzi-term-ceiling-subtract* [*akra-bazzi-term-intros*]:

**assumes**  $(b::real) > 0 \ b < 1 \ real \ x_0 \leq b * real \ x_1 - c \ 1 \leq c + (1 - b) * x_1$   
**shows** *akra-bazzi-term*  $x_0 \ x_1 \ b (\lambda x. nat \lceil b * real \ x - c \rceil)$   
 $\langle proof \rangle$

```

lemma akra-bazzi-term-ceiling-subtract' [akra-bazzi-term-intros]:
  assumes (b::real) > 0 b < 1 real x0 ≤ b * real x1 – real c 1 ≤ real c + (1 – b)
  * x1
  shows akra-bazzi-term x0 x1 b (λx. nat ⌈b*real x⌉ – c)
  ⟨proof⟩

lemma akra-bazzi-term-ceiling [akra-bazzi-term-intros]:
  assumes (b::real) > 0 b < 1 real x0 ≤ b * real x1 1 ≤ (1 – b) * x1
  shows akra-bazzi-term x0 x1 b (λx. nat ⌈b*real x⌉)
  ⟨proof⟩

end

```

## 5 The Master theorem

```

theory Master-Theorem
imports
  HOL-Analysis.Equivalence-Lebesgue-Henstock-Integration
  Akra-Bazzi-Library
  Akra-Bazzi
begin

lemma fundamental-theorem-of-calculus-real:
  a ≤ b ⇒ ∀x ∈ {a..b}. (f has-real-derivative f' x) (at x within {a..b}) ⇒
    (f' has-integral (f b – f a)) {a..b}
  ⟨proof⟩

lemma integral-powr:
  y ≠ -1 ⇒ a ≤ b ⇒ a > 0 ⇒ integral {a..b} (λx. x powr y :: real) =
    inverse (y + 1) * (b powr (y + 1) – a powr (y + 1))
  ⟨proof⟩

lemma integral-ln-powr-over-x:
  y ≠ -1 ⇒ a ≤ b ⇒ a > 1 ⇒ integral {a..b} (λx. ln x powr y / x :: real) =
    inverse (y + 1) * (ln b powr (y + 1) – ln a powr (y + 1))
  ⟨proof⟩

lemma integral-one-over-x-ln-x:
  a ≤ b ⇒ a > 1 ⇒ integral {a..b} (λx. inverse (x * ln x) :: real) = ln (ln b)
  – ln (ln a)
  ⟨proof⟩

lemma akra-bazzi-integral-kurzweil-henstock:
  akra-bazzi-integral (λf a b. f integrable-on {a..b}) (λf a b. integral {a..b} f)
  ⟨proof⟩

```

```

locale master-theorem-function = akra-bazzi-recursion +
  fixes g :: nat  $\Rightarrow$  real
  assumes f-nonneg-base:  $x \geq x_0 \Rightarrow x < x_1 \Rightarrow f x \geq 0$ 
  and f-rec:  $x \geq x_1 \Rightarrow f x = g x + (\sum i < k. as!i * f ((ts!i) x))$ 
  and g-nonneg:  $x \geq x_1 \Rightarrow g x \geq 0$ 
  and ex-pos-a:  $\exists a \in set as. a > 0$ 
begin

interpretation akra-bazzi-integral  $\lambda f a b. f$  integrable-on {a..b}  $\lambda f a b. integral$ 
{a..b} f
⟨proof⟩

sublocale akra-bazzi-function x0 x1 k as bs ts f  $\lambda f a b. f$  integrable-on {a..b}
 $\lambda f a b. integral$  {a..b} f g
⟨proof⟩

context
begin

private lemma g-nonneg': eventually ( $\lambda x. g x \geq 0$ ) at-top
⟨proof⟩ lemma g-pos:
assumes g ∈ Ω(h)
assumes eventually ( $\lambda x. h x > 0$ ) at-top
shows eventually ( $\lambda x. g x > 0$ ) at-top
⟨proof⟩ lemma f-pos:
assumes g ∈ Ω(h)
assumes eventually ( $\lambda x. h x > 0$ ) at-top
shows eventually ( $\lambda x. f x > 0$ ) at-top
⟨proof⟩

lemma bs-lower-bound:  $\exists C > 0. \forall b \in set bs. C < b$ 
⟨proof⟩ lemma powr-growth2:
 $\exists C c2. 0 < c2 \wedge C < Min (set bs) \wedge$ 
  eventually ( $\lambda x. \forall u \in \{C * x..x\}. c2 * x powr p' \geq u powr p'$ ) at-top
⟨proof⟩ lemma powr-growth1:
 $\exists C c1. 0 < c1 \wedge C < Min (set bs) \wedge$ 
  eventually ( $\lambda x. \forall u \in \{C * x..x\}. c1 * x powr p' \leq u powr p'$ ) at-top
⟨proof⟩ lemma powr-ln-powr-lower-bound:
 $a > 1 \Rightarrow a \leq x \Rightarrow x \leq b \Rightarrow$ 
  min (a powr p) (b powr p) * min (ln a powr p') (ln b powr p')  $\leq x powr p * ln$ 
  x powr p'
⟨proof⟩ lemma powr-ln-powr-upper-bound:
 $a > 1 \Rightarrow a \leq x \Rightarrow x \leq b \Rightarrow$ 
  max (a powr p) (b powr p) * max (ln a powr p') (ln b powr p')  $\geq x powr p * ln$ 
  x powr p'
⟨proof⟩ lemma powr-ln-powr-upper-bound':
  eventually ( $\lambda a. \forall b > a. \exists c. \forall x \in \{a..b\}. x powr p * ln x powr p' \leq c$ ) at-top
⟨proof⟩ lemma powr-upper-bound':
  eventually ( $\lambda a :: real. \forall b > a. \exists c. \forall x \in \{a..b\}. x powr p' \leq c$ ) at-top

```

$\langle proof \rangle$

```
lemmas bounds =
  powr-ln-powr-lower-bound powr-ln-powr-upper-bound powr-ln-powr-upper-bound'
  powr-upper-bound'
```

**private lemma** eventually-ln-const:

assumes ( $C::real$ )  $> 0$

shows eventually  $(\lambda x. ln(C*x) / ln x > 1/2)$  at-top

$\langle proof \rangle$  lemma powr-ln-powr-growth1:  $\exists C c1. 0 < c1 \wedge C < Min (set bs) \wedge$   
 $\text{eventually } (\lambda x. \forall u \in \{C * x..x\}. c1 * (x \text{ powr } r * ln x \text{ powr } r') \leq u \text{ powr } r * ln$   
 $u \text{ powr } r')$  at-top

$\langle proof \rangle$  lemma powr-ln-powr-growth2:  $\exists C c1. 0 < c1 \wedge C < Min (set bs) \wedge$   
 $\text{eventually } (\lambda x. \forall u \in \{C * x..x\}. c1 * (x \text{ powr } r * ln x \text{ powr } r') \geq u \text{ powr } r * ln$   
 $u \text{ powr } r')$  at-top

$\langle proof \rangle$

```
lemmas growths = powr-growth1 powr-growth2 powr-ln-powr-growth1 powr-ln-powr-growth2
```

**private lemma** master-integrable:

$\exists a::real. \forall b \geq a. (\lambda u. u \text{ powr } r * ln u \text{ powr } s / u \text{ powr } t) \text{ integrable-on } \{a..b\}$

$\exists a::real. \forall b \geq a. (\lambda u. u \text{ powr } r / u \text{ powr } s) \text{ integrable-on } \{a..b\}$

$\langle proof \rangle$  lemma master-integral:

fixes  $a p p' :: real$

assumes  $p: p \neq p'$  and  $a: a > 0$

obtains  $c d$  where  $c \neq 0 p > p' \rightarrow d \neq 0$

$(\lambda x::nat. x \text{ powr } p * (1 + \text{integral } \{a..x\} (\lambda u. u \text{ powr } p' / u \text{ powr } (p+1)))) \in$   
 $\Theta(\lambda x::nat. d * x \text{ powr } p + c * x \text{ powr } p')$

$\langle proof \rangle$  lemma master-integral':

fixes  $a p p' :: real$

assumes  $p': p' \neq 0$  and  $a: a > 1$

obtains  $c d :: real$  where  $p' < 0 \rightarrow c \neq 0 d \neq 0$

$(\lambda x::nat. x \text{ powr } p * (1 + \text{integral } \{a..x\} (\lambda u. u \text{ powr } p * ln u \text{ powr } (p'-1) / u$   
 $\text{powr } (p+1)))) \in$

$\Theta(\lambda x::nat. c * x \text{ powr } p + d * x \text{ powr } p * ln x \text{ powr } p')$

$\langle proof \rangle$  lemma master-integral'':

fixes  $a p p' :: real$

assumes  $a: a > 1$

shows  $(\lambda x::nat. x \text{ powr } p * (1 + \text{integral } \{a..x\} (\lambda u. u \text{ powr } p * ln u \text{ powr } -$   
 $1 / u \text{ powr } (p+1)))) \in$

$\Theta(\lambda x::nat. x \text{ powr } p * ln(ln x))$

$\langle proof \rangle$

**lemma** master1-bigo:

assumes g-bigo:  $g \in O(\lambda x. \text{real } x \text{ powr } p')$

```

assumes less-p': ( $\sum i < k. as!i * bs!i \text{ powr } p'$ ) > 1
shows  $f \in O(\lambda x. \text{real } x \text{ powr } p)$ 
⟨proof⟩

```

**lemma** master1:

```

assumes g-bigo:  $g \in O(\lambda x. \text{real } x \text{ powr } p')$ 
assumes less-p': ( $\sum i < k. as!i * bs!i \text{ powr } p'$ ) > 1
assumes f-pos: eventually  $(\lambda x. f x > 0)$  at-top
shows  $f \in \Theta(\lambda x. \text{real } x \text{ powr } p)$ 
⟨proof⟩

```

**lemma** master2-3:

```

assumes g-bigtheta:  $g \in \Theta(\lambda x. \text{real } x \text{ powr } p * \ln(\text{real } x) \text{ powr } (p' - 1))$ 
assumes p':  $p' > 0$ 
shows  $f \in \Theta(\lambda x. \text{real } x \text{ powr } p * \ln(\text{real } x) \text{ powr } p')$ 
⟨proof⟩

```

**lemma** master2-1:

```

assumes g-bigtheta:  $g \in \Theta(\lambda x. \text{real } x \text{ powr } p * \ln(\text{real } x) \text{ powr } p')$ 
assumes p':  $p' < -1$ 
shows  $f \in \Theta(\lambda x. \text{real } x \text{ powr } p)$ 
⟨proof⟩

```

**lemma** master2-2:

```

assumes g-bigtheta:  $g \in \Theta(\lambda x. \text{real } x \text{ powr } p / \ln(\text{real } x))$ 
shows  $f \in \Theta(\lambda x. \text{real } x \text{ powr } p * \ln(\ln(\text{real } x)))$ 
⟨proof⟩

```

**lemma** master3:

```

assumes g-bigtheta:  $g \in \Theta(\lambda x. \text{real } x \text{ powr } p')$ 
assumes p'-greater': ( $\sum i < k. as!i * bs!i \text{ powr } p'$ ) < 1
shows  $f \in \Theta(\lambda x. \text{real } x \text{ powr } p')$ 
⟨proof⟩

```

end

end

end

## 6 Evaluating expressions with rational numerals

**theory** Eval-Numeral

**imports**

Complex-Main

**begin**

**lemma** real-numeral-to-Ratreal:

$(0::\text{real}) = \text{Ratreal}(\text{Frct}(0, 1))$

```

(1::real) = Ratreal (Frct (1, 1))
(numeral x :: real) = Ratreal (Frct (numeral x, 1))
(1::int) = numeral Num.One
⟨proof⟩

lemma real-equals-code: Ratreal x = Ratreal y  $\longleftrightarrow$  x = y
⟨proof⟩

lemma Rat-normalize-idempotent: Rat.normalize (Rat.normalize x) = Rat.normalize x
⟨proof⟩

lemma uminus-pow-Numeral1:  $-(x::\text{monoid-mult}) \wedge \text{Numeral1} = -x$  ⟨proof⟩

lemmas power-numeral-simps = power-0 uminus-pow-Numeral1 power-minus-Bit0
power-minus-Bit1

lemma Fract-normalize: Fract (fst (Rat.normalize (x,y))) (snd (Rat.normalize (x,y))) = Fract x y
⟨proof⟩

lemma Frct-add: Frct (a, numeral b) + Frct (c, numeral d) =
Frct (Rat.normalize (a * numeral d + c * numeral b, numeral (b*d)))
⟨proof⟩

lemma Frct-uminus:  $-(Frct (a,b)) = Frct (-a,b)$  ⟨proof⟩

lemma Frct-diff: Frct (a, numeral b) - Frct (c, numeral d) =
Frct (Rat.normalize (a * numeral d - c * numeral b, numeral (b*d)))
⟨proof⟩

lemma Frct-mult: Frct (a, numeral b) * Frct (c, numeral d) = Frct (a*c, numeral (b*d))
⟨proof⟩

lemma Frct-inverse: inverse (Frct (a, b)) = Frct (b, a) ⟨proof⟩

lemma Frct-divide: Frct (a, numeral b) / Frct (c, numeral d) = Frct (a*numeral d, numeral b * c)
⟨proof⟩

lemma Frct-pow: Frct (a, numeral b)  $\wedge$  c = Frct (a  $\wedge$  c, numeral b  $\wedge$  c)
⟨proof⟩

lemma Frct-less: Frct (a, numeral b) < Frct (c, numeral d)  $\longleftrightarrow$  a * numeral d < c * numeral b

```

$\langle proof \rangle$

**lemma** *Frct-le*:  $Frct(a, \text{numeral } b) \leq Frct(c, \text{numeral } d) \longleftrightarrow a * \text{numeral } d \leq c * \text{numeral } b$   
 $\langle proof \rangle$

**lemma** *Frct-equals*:  $Frct(a, \text{numeral } b) = Frct(c, \text{numeral } d) \longleftrightarrow a * \text{numeral } d = c * \text{numeral } b$   
 $\langle proof \rangle$

**lemma** *real-power-code*:  $(Ratreal x) \wedge y = Ratreal(x \wedge y)$   $\langle proof \rangle$

**lemmas** *real-arith-code* =  
real-plus-code real-minus-code real-times-code real-uminus-code real-inverse-code  
real-divide-code real-power-code real-less-code real-less-eq-code real-equals-code

**lemmas** *rat-arith-code* =  
Frct-add Frct-uminus Frct-diff Frct-mult Frct-inverse Frct-divide Frct-pow  
Frct-less Frct-le Frct>equals

**lemma** *gcd-numeral-red*:  $gcd(\text{numeral } x::\text{int}) (\text{numeral } y) = gcd(\text{numeral } y)$   
 $(\text{numeral } x \text{ mod } \text{numeral } y)$   
 $\langle proof \rangle$

**lemma** *divmod-one*:  
divmod (Num.One) (Num.One) = (Numeral1, 0)  
divmod (Num.One) (Num.Bit0 x) = (0, Numeral1)  
divmod (Num.One) (Num.Bit1 x) = (0, Numeral1)  
divmod x (Num.One) = (numeral x, 0)  
 $\langle proof \rangle$

**lemmas** *divmod-numeral-simps* =  
div-0 div-by-0 mod-0 mod-by-0  
fst-divmod [symmetric]  
snd-divmod [symmetric]  
divmod-cancel  
divmod-steps [simplified rel-simps if-True] divmod-trivial  
rel-simps

**lemma** *Suc-0-to-numeral*:  $Suc 0 = \text{Numeral1}$   $\langle proof \rangle$   
**lemmas** *Suc-to-numeral* = *Suc-0-to-numeral* Num.Suc-1 Num.Suc-numeral

**lemma** *rat-powr*:  
 $0 \text{ powr } y = 0$   
 $x > 0 \implies x \text{ powr } Ratreal(Frct(0, \text{Numeral1})) = Ratreal(Frct(\text{Numeral1}, \text{Numeral1}))$   
 $x > 0 \implies x \text{ powr } Ratreal(Frct(\text{numeral } a, \text{Numeral1})) = x \wedge \text{numeral } a$   
 $x > 0 \implies x \text{ powr } Ratreal(Frct(-\text{numeral } a, \text{Numeral1})) = \text{inverse}(x \wedge \text{numeral } a)$

$\langle proof \rangle$

```
lemmas eval-numeral-simps =
  real-numeral-to-Ratreal real-arith-code rat-arith-code Num.arith-simps
  Rat.normalize-def fst-conv snd-conv gcd-0-int gcd-0-left-int gcd.bottom-right-bottom
  gcd.bottom-left-bottom
  gcd.neg1-int gcd.neg2-int gcd-numeral-red zmod-numeral-Bit0 zmod-numeral-Bit1
  power-numeral-simps
  divmod-numeral-simps numeral-One [symmetric] Groups.Let-0 Num.Let-numeral
  Suc-to-numeral power-numeral
  greaterThanLessThan-iff atLeastAtMost-iff atLeastLessThan-iff greaterThanAt-
  Most-iff rat-powr
  Num.pow.simps Num.sqr.simps Product-Type.split of-int-numeral of-int-neg-numeral
  of-nat-numeral
```

$\langle ML \rangle$

```
lemma 21254387548659589512*314213523632464357453884361*2342523623324234*56432743858724173474
      12561712738645824362329316482973164398214286 powr 2 /
      (1130246312978423123+231212374631082764842731842*122474378389424362347451251263)
>
      (12313244512931247243543279768645745929475829310651205623844::real)
  <proof>
```

end

## 7 The proof methods

### 7.1 Master theorem and termination

theory Akra-Bazzi-Method

imports

Complex-Main

Akra-Bazzi

Master-Theorem

Eval-Numerical

begin

```
lemma landau-symbol-ge-3-cong:
  assumes landau-symbol L L' Lr
  assumes  $\bigwedge x::'a::linordered-semidom. x \geq 3 \implies f x = g x$ 
  shows L at-top (f) = L at-top (g)
  <proof>
```

```
lemma exp-1-lt-3: exp (1::real) < 3
  <proof>
```

```
lemma ln-ln-pos:
  assumes (x::real)  $\geq 3$ 
```

**shows**  $\ln(\ln x) > 0$   
 $\langle proof \rangle$

**definition akra-bazzi-terms where**

$akra\text{-bazzi-terms } x_0\ x_1\ bs\ ts = (\forall i < \text{length } bs. \text{ akra-bazzi-term } x_0\ x_1\ (bs!i)\ (ts!i))$

**lemma akra-bazzi-termsI:**

$(\bigwedge i. i < \text{length } bs \implies \text{ akra-bazzi-term } x_0\ x_1\ (bs!i)\ (ts!i)) \implies \text{ akra-bazzi-terms } x_0\ x_1\ bs\ ts$   
 $\langle proof \rangle$

**lemma master-theorem-functionI:**

**assumes**  $\forall x \in \{x_0 \dots x_1\}. f x \geq 0$

**assumes**  $\forall x \geq x_1. f x = g x + (\sum i < k. \text{as} ! i * f((ts ! i) x))$

**assumes**  $\forall x \geq x_1. g x \geq 0$

**assumes**  $\forall a \in \text{set as}. a \geq 0$

**assumes**  $\text{list-ex } (\lambda a. a > 0) \text{ as}$

**assumes**  $\forall b \in \text{set bs}. b \in \{0 < \dots < 1\}$

**assumes**  $k \neq 0$

**assumes**  $\text{length as} = k$

**assumes**  $\text{length bs} = k$

**assumes**  $\text{length ts} = k$

**assumes**  $\text{ akra-bazzi-terms } x_0\ x_1\ bs\ ts$

**shows**  $\text{ master-theorem-function } x_0\ x_1\ k \text{ as } bs\ ts\ f\ g$

$\langle proof \rangle$

**lemma akra-bazzi-term-measure:**

$x \geq x_1 \implies \text{ akra-bazzi-term } 0\ x_1\ b\ t \implies (t\ x, x) \in \text{ Wellfounded.measure } (\lambda n :: \text{nat}. n)$

$x > x_1 \implies \text{ akra-bazzi-term } 0\ (\text{Suc } x_1)\ b\ t \implies (t\ x, x) \in \text{ Wellfounded.measure } (\lambda n :: \text{nat}. n)$

$\langle proof \rangle$

**lemma measure-prod-conv:**

$((a, b), (c, d)) \in \text{ Wellfounded.measure } (\lambda x. t\ (\text{fst } x)) \longleftrightarrow (a, c) \in \text{ Wellfounded.measure } t$

$((e, f), (g, h)) \in \text{ Wellfounded.measure } (\lambda x. t\ (\text{snd } x)) \longleftrightarrow (f, h) \in \text{ Wellfounded.measure } t$

$\langle proof \rangle$

**lemmas**  $\text{ measure-prod-conv}' = \text{ measure-prod-conv}[\text{where } t = \lambda x. x]$

**lemma akra-bazzi-termination-simps:**

**fixes**  $x :: \text{nat}$

**shows**  $a * \text{real } x / b = a/b * \text{real } x \text{ real } x / b = 1/b * \text{real } x$

$\langle proof \rangle$

**lemma akra-bazzi-params-nonzeroI:**

$\text{length as} = \text{length bs} \implies$

$(\forall a \in \text{set } as. a \geq 0) \implies (\forall b \in \text{set } bs. b \in \{0 <.. < 1\}) \implies (\exists a \in \text{set } as. a > 0) \implies$   
*akra-bazzi-params-nonzero (length as) as bs ⟨proof⟩*

**lemmas** *akra-bazzi-p-rel-intros* =

*akra-bazzi-params-nonzero.p-lessI*[*rotated*, *OF - akra-bazzi-params-nonzeroI*]  
*akra-bazzi-params-nonzero.p-greaterI*[*rotated*, *OF - akra-bazzi-params-nonzeroI*]  
*akra-bazzi-params-nonzero.p-leI*[*rotated*, *OF - akra-bazzi-params-nonzeroI*]  
*akra-bazzi-params-nonzero.p-geI*[*rotated*, *OF - akra-bazzi-params-nonzeroI*]  
*akra-bazzi-params-nonzero.p-boundsI*[*rotated*, *OF - akra-bazzi-params-nonzeroI*]  
*akra-bazzi-params-nonzero.p-boundsI'*[*rotated*, *OF - akra-bazzi-params-nonzeroI*]

**lemma** *eval-length*:  $\text{length } [] = 0$   $\text{length } (x \# xs) = \text{Suc } (\text{length } xs)$  ⟨⟨proof⟩⟩

**lemma** *eval-akra-bazzi-sum*:

$(\sum i < 0. as!i * bs!i \text{ powr } x) = 0$   
 $(\sum i < \text{Suc } 0. (a \# as)!i * (b \# bs)!i \text{ powr } x) = a * b \text{ powr } x$   
 $(\sum i < \text{Suc } k. (a \# as)!i * (b \# bs)!i \text{ powr } x) = a * b \text{ powr } x + (\sum i < k. as!i * bs!i \text{ powr } x)$   
⟨⟨proof⟩⟩

**lemma** *eval-akra-bazzi-sum'*:

$(\sum i < 0. as!i * f ((ts!i) x)) = 0$   
 $(\sum i < \text{Suc } 0. (a \# as)!i * f (((t \# ts)!i) x)) = a * f (t x)$   
 $(\sum i < \text{Suc } k. (a \# as)!i * f (((t \# ts)!i) x)) = a * f (t x) + (\sum i < k. as!i * f ((ts!i) x))$   
⟨⟨proof⟩⟩

**lemma** *akra-bazzi-termsI'*:

*akra-bazzi-terms*  $x_0 x_1 [] []$   
*akra-bazzi-term*  $x_0 x_1 b t \implies$  *akra-bazzi-terms*  $x_0 x_1 bs ts \implies$  *akra-bazzi-terms*  
 $x_0 x_1 (b \# bs) (t \# ts)$   
⟨⟨proof⟩⟩

**lemma** *ball-set-intros*:  $(\forall x \in \text{set } []. P x) P x \implies (\forall x \in \text{set } xs. P x) \implies (\forall x \in \text{set } (x \# xs). P x)$   
⟨⟨proof⟩⟩

**lemma** *ball-set-simps*:  $(\forall x \in \text{set } []. P x) = \text{True}$   $(\forall x \in \text{set } (x \# xs). P x) = (P x \wedge \forall x \in \text{set } xs. P x)$   
⟨⟨proof⟩⟩

**lemma** *bex-set-simps*:  $(\exists x \in \text{set } []. P x) = \text{False}$   $(\exists x \in \text{set } (x \# xs). P x) = (P x \vee \exists x \in \text{set } xs. P x)$   
⟨⟨proof⟩⟩

**lemma** *eval-akra-bazzi-le-list-ex*:

*list-ex*  $P (x \# y \# xs) \longleftrightarrow P x \vee \text{list-ex } P (y \# xs)$   
*list-ex*  $P [x] \longleftrightarrow P x$   
*list-ex*  $P [] \longleftrightarrow \text{False}$

$\langle proof \rangle$

**lemma** eval-akra-bazzi-le-sum-list:

$$\begin{aligned} x \leq \text{sum-list } [] &\longleftrightarrow x \leq 0 \quad x \leq \text{sum-list } (y \# ys) \longleftrightarrow x \leq y + \text{sum-list } ys \\ x \leq z + \text{sum-list } [] &\longleftrightarrow x \leq z \quad x \leq z + \text{sum-list } (y \# ys) \longleftrightarrow x \leq z + y + \text{sum-list } ys \end{aligned}$$

$\langle proof \rangle$

**lemma** atLeastLessThanE:  $x \in \{a..<b\} \implies (x \geq a \implies x < b \implies P) \implies P$

$\langle proof \rangle$

**lemma** master-theorem-preprocess:

$$\begin{aligned} \Theta(\lambda n :: \text{nat}. \ 1) &= \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 0) \\ \Theta(\lambda n :: \text{nat}. \ \text{real } n) &= \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 1) \\ O(\lambda n :: \text{nat}. \ 1) &= O(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 0) \\ O(\lambda n :: \text{nat}. \ \text{real } n) &= O(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 1) \end{aligned}$$

$$\Theta(\lambda n :: \text{nat}. \ \ln(\ln(\text{real } n))) = \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 0 * \ln(\ln(\text{real } n)))$$

$$\Theta(\lambda n :: \text{nat}. \ \text{real } n * \ln(\ln(\text{real } n))) = \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 1 * \ln(\ln(\text{real } n)))$$

$$\Theta(\lambda n :: \text{nat}. \ \ln(\text{real } n)) = \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 0 * \ln(\text{real } n) \text{ powr } 1)$$

$$\Theta(\lambda n :: \text{nat}. \ \text{real } n * \ln(\text{real } n)) = \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 1 * \ln(\text{real } n) \text{ powr } 1)$$

$$\Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } p * \ln(\text{real } n)) = \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } p * \ln(\text{real } n) \text{ powr } 1)$$

$$\Theta(\lambda n :: \text{nat}. \ \ln(\text{real } n) \text{ powr } p') = \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 0 * \ln(\text{real } n) \text{ powr } p')$$

$$\Theta(\lambda n :: \text{nat}. \ \text{real } n * \ln(\text{real } n) \text{ powr } p') = \Theta(\lambda n :: \text{nat}. \ \text{real } n \text{ powr } 1 * \ln(\text{real } n) \text{ powr } p')$$

$\langle proof \rangle$

**lemma** akra-bazzi-term-imp-size-less:

$$x_1 \leq x \implies \text{akra-bazzi-term } 0 \ x_1 \ b \ t \implies \text{size}(t \ x) < \text{size } x$$

$$x_1 < x \implies \text{akra-bazzi-term } 0 \ (\text{Suc } x_1) \ b \ t \implies \text{size}(t \ x) < \text{size } x$$

$\langle proof \rangle$

**definition** CLAMP ( $f :: \text{nat} \Rightarrow \text{real}$ )  $x = (\text{if } x < 3 \text{ then } 0 \text{ else } f \ x)$

**definition** CLAMP' ( $f :: \text{nat} \Rightarrow \text{real}$ )  $x = (\text{if } x < 3 \text{ then } 0 \text{ else } f \ x)$

**definition** MASTER-BOUND  $a \ b \ c \ x = \text{real } x \text{ powr } a * \ln(\text{real } x) \text{ powr } b * \ln(\ln(\text{real } x)) \text{ powr } c$

**definition** MASTER-BOUND'  $a \ b \ x = \text{real } x \text{ powr } a * \ln(\text{real } x) \text{ powr } b$

**definition** MASTER-BOUND''  $a \ x = \text{real } x \text{ powr } a$

**lemma** ln-1-imp-less-3:

$$\ln x = (1 :: \text{real}) \implies x < 3$$

$\langle proof \rangle$

**lemma** ln-1-imp-less-3':  $\ln(\text{real } (x :: \text{nat})) = 1 \implies x < 3$   $\langle proof \rangle$

**lemma** ln-ln-nonneg:  $x \geq (3 :: \text{real}) \implies \ln(\ln x) \geq 0$   $\langle proof \rangle$

**lemma** ln-ln-nonneg':  $x \geq (3 :: \text{nat}) \implies \ln(\ln(\text{real } x)) \geq 0$   $\langle proof \rangle$

**lemma** *MASTER-BOUND-postproc*:

$$\begin{aligned} \text{CLAMP}(\text{MASTER-BOUND}' a 0) &= \text{CLAMP}(\text{MASTER-BOUND}'' a) \\ \text{CLAMP}(\text{MASTER-BOUND}' a 1) &= \text{CLAMP}(\lambda x. \text{CLAMP}(\text{MASTER-BOUND}'' a) x * \text{CLAMP}(\lambda x. \ln(\text{real } x)) x) \\ \text{CLAMP}(\text{MASTER-BOUND}' a (\text{numeral } n)) &= \\ &\quad \text{CLAMP}(\lambda x. \text{CLAMP}(\text{MASTER-BOUND}'' a) x * \text{CLAMP}(\lambda x. \ln(\text{real } x) \wedge \text{numeral } n) x) \\ \text{CLAMP}(\text{MASTER-BOUND}' a (-1)) &= \\ &\quad \text{CLAMP}(\lambda x. \text{CLAMP}(\text{MASTER-BOUND}'' a) x / \text{CLAMP}(\lambda x. \ln(\text{real } x)) x) \\ \text{CLAMP}(\text{MASTER-BOUND}' a (-\text{numeral } n)) &= \\ &\quad \text{CLAMP}(\lambda x. \text{CLAMP}(\text{MASTER-BOUND}'' a) x / \text{CLAMP}(\lambda x. \ln(\text{real } x) \wedge \text{numeral } n) x) \\ \text{CLAMP}(\text{MASTER-BOUND}' a b) &= \\ &\quad \text{CLAMP}(\lambda x. \text{CLAMP}(\text{MASTER-BOUND}'' a) x * \text{CLAMP}(\lambda x. \ln(\text{real } x) \text{ powr } b) x) \\ \\ \text{CLAMP}(\text{MASTER-BOUND}'' 0) &= \text{CLAMP}(\lambda x. 1) \\ \text{CLAMP}(\text{MASTER-BOUND}'' 1) &= \text{CLAMP}(\lambda x. (\text{real } x)) \\ \text{CLAMP}(\text{MASTER-BOUND}'' (\text{numeral } n)) &= \text{CLAMP}(\lambda x. (\text{real } x) \wedge \text{numeral } n) \\ \text{CLAMP}(\text{MASTER-BOUND}'' (-1)) &= \text{CLAMP}(\lambda x. 1 / (\text{real } x)) \\ \text{CLAMP}(\text{MASTER-BOUND}'' (-\text{numeral } n)) &= \text{CLAMP}(\lambda x. 1 / (\text{real } x) \wedge \text{numeral } n) \\ \text{CLAMP}(\text{MASTER-BOUND}'' a) &= \text{CLAMP}(\lambda x. (\text{real } x) \text{ powr } a) \end{aligned}$$

**and** *MASTER-BOUND-UNCLAMP*:

$$\begin{aligned} \text{CLAMP}(\lambda x. \text{CLAMP } f x * \text{CLAMP } g x) &= \text{CLAMP}(\lambda x. f x * g x) \\ \text{CLAMP}(\lambda x. \text{CLAMP } f x / \text{CLAMP } g x) &= \text{CLAMP}(\lambda x. f x / g x) \\ \text{CLAMP}(\text{CLAMP } f) &= \text{CLAMP } f \\ \langle \text{proof} \rangle & \end{aligned}$$

**context**

**begin**

**private lemma** *CLAMP-*:

$$\text{landau-symbol } L L' Lr \implies L \text{ at-top } (f :: \text{nat} \Rightarrow \text{real}) \equiv L \text{ at-top } (\lambda x. \text{CLAMP } f x)$$

$$\langle \text{proof} \rangle$$

**lemma** *UNCLAMP'-*:

$$\text{landau-symbol } L L' Lr \implies L \text{ at-top } (\text{CLAMP}'(\text{MASTER-BOUND } a b c)) \equiv L \text{ at-top } (\text{MASTER-BOUND } a b c)$$

$$\langle \text{proof} \rangle$$

**lemma** *UNCLAMP-*:

$$\text{landau-symbol } L L' Lr \implies L \text{ at-top } (\text{CLAMP } f) \equiv L \text{ at-top } (f)$$

$$\langle \text{proof} \rangle$$

**lemmas** *CLAMP* = *landau-symbols*[THEN *CLAMP-*]

**lemmas** *UNCLAMP'* = *landau-symbols*[THEN *UNCLAMP'-*]

```

lemmas UNCLAMP = landau-symbols[THEN UNCLAMP-]
end

lemma propagate-CLAMP:

$$\begin{aligned} CLAMP(\lambda x. f x * g x) &= CLAMP'(\lambda x. CLAMP f x * CLAMP g x) \\ CLAMP(\lambda x. f x / g x) &= CLAMP'(\lambda x. CLAMP f x / CLAMP g x) \\ CLAMP(\lambda x. inverse(f x)) &= CLAMP'(\lambda x. inverse(CLAMP f x)) \\ CLAMP(\lambda x. real x) &= CLAMP'(\text{MASTER-BOUND } 1 \ 0 \ 0) \\ CLAMP(\lambda x. real x powr a) &= CLAMP'(\text{MASTER-BOUND } a \ 0 \ 0) \\ CLAMP(\lambda x. real x ^ a') &= CLAMP'(\text{MASTER-BOUND } (\text{real } a') \ 0 \ 0) \\ CLAMP(\lambda x. ln(\text{real } x)) &= CLAMP'(\text{MASTER-BOUND } 0 \ 1 \ 0) \\ CLAMP(\lambda x. ln(\text{real } x) powr b) &= CLAMP'(\text{MASTER-BOUND } 0 \ b \ 0) \\ CLAMP(\lambda x. ln(\text{real } x) ^ b') &= CLAMP'(\text{MASTER-BOUND } 0 \ (\text{real } b') \ 0) \\ CLAMP(\lambda x. ln(ln(\text{real } x))) &= CLAMP'(\text{MASTER-BOUND } 0 \ 0 \ 1) \\ CLAMP(\lambda x. ln(ln(\text{real } x)) powr c) &= CLAMP'(\text{MASTER-BOUND } 0 \ 0 \ c) \\ CLAMP(\lambda x. ln(ln(\text{real } x)) ^ c') &= CLAMP'(\text{MASTER-BOUND } 0 \ 0 \ (\text{real } c')) \\ CLAMP'(CLAMP f) &= CLAMP' f \\ CLAMP'(\lambda x. CLAMP'(\text{MASTER-BOUND } a1 \ b1 \ c1) x * CLAMP'(\text{MASTER-BOUND } a2 \ b2 \ c2) x) &= \\ &\quad CLAMP'(\text{MASTER-BOUND } (a1+a2) \ (b1+b2) \ (c1+c2)) \\ CLAMP'(\lambda x. CLAMP'(\text{MASTER-BOUND } a1 \ b1 \ c1) x / CLAMP'(\text{MASTER-BOUND } a2 \ b2 \ c2) x) &= \\ &\quad CLAMP'(\text{MASTER-BOUND } (a1-a2) \ (b1-b2) \ (c1-c2)) \\ CLAMP'(\lambda x. inverse(\text{MASTER-BOUND } a1 \ b1 \ c1 x)) &= CLAMP'(\text{MASTER-BOUND } (-a1) \ (-b1) \ (-c1)) \\ \langle proof \rangle \end{aligned}$$


lemma numeral-assoc-simps:

$$\begin{aligned} ((a::\text{real}) + \text{numeral } b) + \text{numeral } c &= a + \text{numeral } (b + c) \\ (a + \text{numeral } b) - \text{numeral } c &= a + \text{neg-numeral-class.sub } b \ c \\ (a - \text{numeral } b) + \text{numeral } c &= a + \text{neg-numeral-class.sub } c \ b \\ (a - \text{numeral } b) - \text{numeral } c &= a - \text{numeral } (b + c) \ \langle proof \rangle \end{aligned}$$


lemmas CLAMP-aux =
arith-simps numeral-assoc-simps of-nat-power of-nat-mult of-nat-numeral
one-add-one numeral-One [symmetric]

lemmas CLAMP-postproc = numeral-One

context master-theorem-function
begin

lemma master1-bigo-automation:
assumes  $g \in O(\lambda x. \text{real } x \text{ powr } p')$   $1 < (\sum i < k. \text{as } ! \ i * \text{bs } ! \ i \text{ powr } p')$ 
shows  $f \in O(\text{MASTER-BOUND } p \ 0 \ 0)$ 
⟨proof⟩

lemma master1-automation:
assumes  $g \in O(\text{MASTER-BOUND}'' p')$   $1 < (\sum i < k. \text{as } ! \ i * \text{bs } ! \ i \text{ powr } p')$ 
```

```

eventually ( $\lambda x. f x > 0$ ) at-top
shows  $f \in \Theta(\text{MASTER-BOUND } p \ 0 \ 0)$ 
(proof)

lemma master2-1-automation:
assumes  $g \in \Theta(\text{MASTER-BOUND}' p \ p') \ p' < -1$ 
shows  $f \in \Theta(\text{MASTER-BOUND } p \ 0 \ 0)$ 
(proof)

lemma master2-2-automation:
assumes  $g \in \Theta(\text{MASTER-BOUND}' p \ (-1))$ 
shows  $f \in \Theta(\text{MASTER-BOUND } p \ 0 \ 1)$ 
(proof)

lemma master2-3-automation:
assumes  $g \in \Theta(\text{MASTER-BOUND}' p \ (p' - 1)) \ p' > 0$ 
shows  $f \in \Theta(\text{MASTER-BOUND } p \ p' \ 0)$ 
(proof)

lemma master3-automation:
assumes  $g \in \Theta(\text{MASTER-BOUND}'' p') \ 1 > (\sum i < k. \text{as} ! \ i * \text{bs} ! \ i \text{ powr } p')$ 
shows  $f \in \Theta(\text{MASTER-BOUND } p' \ 0 \ 0)$ 
(proof)

lemmas master-automation =
master1-automation master2-1-automation master2-2-automation
master2-2-automation master3-automation

```

$\langle ML \rangle$

**end**

```

definition arith-consts ( $x :: \text{real}$ ) ( $y :: \text{nat}$ ) =
( $\text{if } \neg(-x) + 3 / x * 5 - 1 \leq x \wedge \text{True} \vee \text{True} \longrightarrow \text{True} \text{ then}$ 
 $x < \text{inverse } 3 \text{ powr } 21 \text{ else } x = \text{real} (\text{Suc } 0 \wedge 2 +$ 
 $(\text{if } 42 - x \leq 1 \wedge 1 \text{ div } y = y \text{ mod } 2 \vee y < \text{Numeral1} \text{ then } 0 \text{ else } 0)) + \text{Numeral1}$ )

```

$\langle ML \rangle$

**hide-const** *arith-consts*

$\langle ML \rangle$

```

hide-const CLAMP CLAMP' MASTER-BOUND MASTER-BOUND' MASTER-BOUND''

end
theory Akra-Bazzi-Approximation

```

```

imports
  Complex-Main
  Akra-Bazzi-Method
  HOL-Decision-Procs.Approximation
begin

context akra-bazzi-params-nonzero
begin

lemma sum-alt:  $(\sum i < k. as!i * bs!i \text{ powr } p') = (\sum i < k. as!i * \exp(p' * \ln(bs!i)))$ 
   $\langle proof \rangle$ 

lemma akra-bazzi-p-rel-intros-aux:
   $1 < (\sum i < k. as!i * \exp(p' * \ln(bs!i))) \implies p' < p$ 
   $1 > (\sum i < k. as!i * \exp(p' * \ln(bs!i))) \implies p' > p$ 
   $1 \leq (\sum i < k. as!i * \exp(p' * \ln(bs!i))) \implies p' \leq p$ 
   $1 \geq (\sum i < k. as!i * \exp(p' * \ln(bs!i))) \implies p' \geq p$ 
   $(\sum i < k. as!i * \exp(x * \ln(bs!i))) \leq 1 \wedge (\sum i < k. as!i * \exp(y * \ln(bs!i))) \geq 1 \implies p \in \{y..x\}$ 
   $(\sum i < k. as!i * \exp(x * \ln(bs!i))) < 1 \wedge (\sum i < k. as!i * \exp(y * \ln(bs!i))) > 1 \implies p \in \{y < .. < x\}$ 
   $\langle proof \rangle$ 
end

lemmas akra-bazzi-p-rel-intros-exp =
  akra-bazzi-params-nonzero.akra-bazzi-p-rel-intros-aux[rotated, OF - akra-bazzi-params-nonzeroI]

lemma eval-akra-bazzi-sum:
   $(\sum i < 0. as!i * \exp(x * \ln(bs!i))) = 0$ 
   $(\sum i < Suc 0. (a#as)!i * \exp(x * \ln((b#bs)!i))) = a * \exp(x * \ln b)$ 
   $(\sum i < Suc k. (a#as)!i * \exp(x * \ln((b#bs)!i))) = a * \exp(x * \ln b) + (\sum i < k. as!i * \exp(x * \ln(bs!i)))$ 
   $\langle proof \rangle$ 

end

```

## 8 Examples

```

theory Master-Theorem-Examples
imports
  Complex-Main
  Akra-Bazzi-Method

```

*Akra-Bazzi-Approximation*  
**begin**

### 8.1 Merge sort

```

function merge-sort-cost :: (nat ⇒ real) ⇒ nat ⇒ real where
  merge-sort-cost t 0 = 0
  | merge-sort-cost t 1 = 1
  | n ≥ 2 ⇒ merge-sort-cost t n =
    merge-sort-cost t (nat ⌊real n / 2⌋) + merge-sort-cost t (nat ⌈real n / 2⌉) + t
  n
  ⟨proof⟩
termination ⟨proof⟩

lemma merge-sort-nonneg[simp]: (¬n. t n ≥ 0) ⇒ merge-sort-cost t x ≥ 0
  ⟨proof⟩

lemma t ∈ Θ(λn. real n) ⇒ (¬n. t n ≥ 0) ⇒ merge-sort-cost t ∈ Θ(λn. real
  n * ln (real n))
  ⟨proof⟩

```

### 8.2 Karatsuba multiplication

```

function karatsuba-cost :: nat ⇒ real where
  karatsuba-cost 0 = 0
  | karatsuba-cost 1 = 1
  | n ≥ 2 ⇒ karatsuba-cost n =
    3 * karatsuba-cost (nat ⌊real n / 2⌋) + real n
  ⟨proof⟩
termination ⟨proof⟩

lemma karatsuba-cost-nonneg[simp]: karatsuba-cost n ≥ 0
  ⟨proof⟩

lemma karatsuba-cost ∈ O(λn. real n powr log 2 3)
  ⟨proof⟩

lemma karatsuba-cost-pos: n ≥ 1 ⇒ karatsuba-cost n > 0
  ⟨proof⟩

lemma karatsuba-cost ∈ Θ(λn. real n powr log 2 3)
  ⟨proof⟩

```

### 8.3 Strassen matrix multiplication

```

function strassen-cost :: nat ⇒ real where
  strassen-cost 0 = 0
  | strassen-cost 1 = 1
  | n ≥ 2 ⇒ strassen-cost n = 7 * strassen-cost (nat ⌊real n / 2⌋) + real (n^2)
  ⟨proof⟩

```

```

termination ⟨proof⟩

lemma strassen-cost-nonneg[simp]: strassen-cost  $n \geq 0$ 
⟨proof⟩

lemma strassen-cost ∈  $O(\lambda n. \text{real } n \text{ powr } \log 2 \gamma)$ 
⟨proof⟩

lemma strassen-cost-pos:  $n \geq 1 \implies \text{strassen-cost } n > 0$ 
⟨proof⟩

lemma strassen-cost ∈  $\Theta(\lambda n. \text{real } n \text{ powr } \log 2 \gamma)$ 
⟨proof⟩

```

## 8.4 Deterministic select

```

function select-cost :: nat ⇒ real where
   $n \leq 20 \implies \text{select-cost } n = 0$ 
  |  $n > 20 \implies \text{select-cost } n =$ 
     $\text{select-cost}(\text{nat} \lfloor \text{real } n / 5 \rfloor) + \text{select-cost}(\text{nat} \lfloor \gamma * \text{real } n / 10 \rfloor + 6) + 12$ 
     $* \text{real } n / 5$ 
  ⟨proof⟩
termination ⟨proof⟩

lemma select-cost ∈  $\Theta(\lambda n. \text{real } n)$ 
⟨proof⟩

```

## 8.5 Decreasing function

```

function dec-cost :: nat ⇒ real where
   $n \leq 2 \implies \text{dec-cost } n = 1$ 
  |  $n > 2 \implies \text{dec-cost } n = 0.5 * \text{dec-cost}(\text{nat} \lfloor \text{real } n / 2 \rfloor) + 1 / \text{real } n$ 
  ⟨proof⟩
termination ⟨proof⟩

```

```

lemma dec-cost ∈  $\Theta(\lambda x::\text{nat}. \ln x / x)$ 
⟨proof⟩

```

## 8.6 Example taken from Drmota and Szpakowski

```

function drmota1 :: nat ⇒ real where
   $n < 20 \implies \text{drmota1 } n = 1$ 
  |  $n \geq 20 \implies \text{drmota1 } n = 2 * \text{drmota1}(\text{nat} \lfloor \text{real } n / 2 \rfloor) + 8/9 * \text{drmota1}(\text{nat} \lfloor 3 * \text{real } n / 4 \rfloor) + \text{real } n^2 / \ln(\text{real } n)$ 
  ⟨proof⟩
termination ⟨proof⟩

lemma drmota1 ∈  $\Theta(\lambda n::\text{real}. n^2 * \ln(\ln n))$ 
⟨proof⟩

```

```

function drmota2 :: nat  $\Rightarrow$  real where
   $n < 20 \implies \text{drmota2 } n = 1$ 
   $| n \geq 20 \implies \text{drmota2 } n = 1/3 * \text{drmota2} (\text{nat} \lfloor \text{real } n/3 + 1/2 \rfloor) + 2/3 * \text{drmota2} (\text{nat} \lfloor 2*\text{real } n/3 - 1/2 \rfloor) + 1$ 
  <proof>
termination <proof>

lemma drmota2  $\in \Theta(\lambda x. \ln(\text{real } x))$ 
  <proof>

lemma boncelet-phrase-length:
  fixes p δ :: real assumes p:  $p > 0$  p < 1 and δ:  $\delta > 0$   $\delta < 1$   $2*p + \delta < 2$ 
  fixes d :: nat  $\Rightarrow$  real
  defines q  $\equiv 1 - p$ 
  assumes d-nonneg:  $\bigwedge n. d \geq 0$ 
  assumes d-rec:  $\bigwedge n. n \geq 2 \implies d = 1 + p * d (\text{nat} \lfloor p * \text{real } n + \delta \rfloor) + q * d (\text{nat} \lfloor q * \text{real } n - \delta \rfloor)$ 
  shows d  $\in \Theta(\lambda x. \ln x)$ 
  <proof>

```

## 8.7 Transcendental exponents

```

function foo-cost :: nat  $\Rightarrow$  real where
   $n < 200 \implies \text{foo-cost } n = 0$ 
   $| n \geq 200 \implies \text{foo-cost } n = \text{foo-cost} (\text{nat} \lfloor \text{real } n / 3 \rfloor) + \text{foo-cost} (\text{nat} \lfloor 3 * \text{real } n / 4 \rfloor + 42) + \text{real } n$ 
  <proof>
termination <proof>

lemma foo-cost-nonneg [simp]:  $\text{foo-cost } n \geq 0$ 
  <proof>

```

```

lemma foo-cost  $\in \Theta(\lambda n. \text{real } n \text{ powr akrabazzi-exponent } [1,1] [1/3,3/4])$ 
  <proof>

```

```

lemma akrabazzi-exponent [1,1] [1/3,3/4]  $\in \{1.1519623..1.1519624\}$ 
  <proof>

```

## 8.8 Functions in locale contexts

```

locale det-select =
  fixes b :: real
  assumes b:  $b > 0$   $b < 7/10$ 
begin

function select-cost' :: nat  $\Rightarrow$  real where
   $n \leq 20 \implies \text{select-cost}' n = 0$ 
   $| n > 20 \implies \text{select-cost}' n =$ 

```

```

select-cost' (nat ⌊real n / 5⌋) + select-cost' (nat ⌊b * real n⌋ + 6) + 6 * real
n + 5
⟨proof⟩
termination ⟨proof⟩

lemma a ≥ 0 ⇒ select-cost' ∈ Θ(λn. real n)
⟨proof⟩

end

```

## 8.9 Non-curried functions

```

function baz-cost :: nat × nat ⇒ real where
  n ≤ 2 ⇒ baz-cost (a, n) = 0
  | n > 2 ⇒ baz-cost (a, n) = 3 * baz-cost (a, nat ⌊real n / 2⌋) + real a
  ⟨proof⟩
termination ⟨proof⟩

lemma baz-cost-nonneg [simp]: a ≥ 0 ⇒ baz-cost (a, n) ≥ 0
⟨proof⟩

lemma
  assumes a > 0
  shows (λx. baz-cost (a, x)) ∈ Θ(λx. x powr log 2 3)
⟨proof⟩

```

```

function bar-cost :: nat × nat ⇒ real where
  n ≤ 2 ⇒ bar-cost (a, n) = 0
  | n > 2 ⇒ bar-cost (a, n) = 3 * bar-cost (2 * a, nat ⌊real n / 2⌋) + real a
  ⟨proof⟩
termination ⟨proof⟩

```

## 8.10 Ham-sandwich trees

```

function ham-sandwich-cost :: nat ⇒ real where
  n < 4 ⇒ ham-sandwich-cost n = 1
  | n ≥ 4 ⇒ ham-sandwich-cost n =
    ham-sandwich-cost (nat ⌊n/4⌋) + ham-sandwich-cost (nat ⌊n/2⌋) + 1
  ⟨proof⟩
termination ⟨proof⟩

```

```

lemma ham-sandwich-cost-pos [simp]: ham-sandwich-cost n > 0
⟨proof⟩

```

The golden ratio

```

definition φ = ((1 + sqrt 5) / 2 :: real)

```

```

lemma φ-pos [simp]: φ > 0 and φ-nonneg [simp]: φ ≥ 0 and φ-nonzero [simp]:
φ ≠ 0

```

$\langle proof \rangle$

**lemma** *ham-sandwich-cost*  $\in \Theta(\lambda n. n \text{ powr} (\log 2 \varphi))$   
 $\langle proof \rangle$

**end**

## References

- [1] M. Akra and L. Bazzi. On the solution of linear recurrence equations. *Computational Optimization and Applications*, 10(2):195–210, 1998.
- [2] T. Leighton. Notes on better Master theorems for divide-and-conquer recurrences. 1996.