Height Bounds for Height-Balanced Trees

Manuel Eberl

July 7, 2026

Abstract

This article takes a fresh look at the class of binary trees known as *height-balanced trees*, where for each node, the height difference between the left and right subtree is bounded by some fixed integer $d > 0$. An interesting question from an algorithmic perspective is how bad the imbalance of such trees can be in the worst case. Luccio and Pagli [1] showed that the worst-case size of a tree of height h is roughly $B_d \cdot C_d^h$ for large h for some specific real numbers $B_d, C_d > 1$.

This formalisation contains:

- A simpler proof that the worst-case size is at least C_d^h for all h
- An explicit closed-form expression for the worst-case size
- A closed-form expression for B_d in terms of C_d and vice versa
- Explicit bounds for C_d in terms of d
- Asymptotic expansions for C_d and B_d as $d \to \infty$

Contents

1 Auxiliary material

theory *HBT-Lemma-Bucket*
imports
Complex-Main
HOL−Real-Asymp.Real-Asymp
HOL−Library.Function-Algebras
HOL−Library.Set-Algebras
Linear-Recurrences.Factorizations
Polynomial-Factorization.Fundamental-Theorem-Algebra-Factorized
HOL−Computational-Algebra.Field-as-Ring
begin

1.1 Polynomials and formal power series

lemma *rsquarefree-card-degree*:
fixes p :: *complex poly*
assumes *rsquarefree p*
shows *card* $\{z.$ *poly* $p\ z = 0\}$ = *degree* p
⟨*proof*⟩

lemmas [*simp del*] = *div-mult-self1 div-mult-self2 div-mult-self3 div-mult-self4*

lemma *pCons-conv-monom*: *pCons a p* = [:*a*::*'a*::*comm-semiring-1*:] + *monom 1 1* ∗ *p*
⟨*proof*⟩

lemma *pCons-conv-monom'*: *NO-MATCH 0 p* ⟹ *pCons a p* = [:*a*::*'a*::*comm-semiring-1*:] + *monom 1 1* ∗ *p*
⟨*proof*⟩

lemma *rsquarefree-def'*: *rsquarefree p* ⟷ $p \neq 0 \wedge (\forall a.$ *order a p* $\leq 1)$
⟨*proof*⟩

lemma *poly-div*: *poly q x* ≠ *0* ⟹ *q dvd p* ⟹ *poly* (*p div q*) *x* = *poly p x* / *poly q* (*x* :: *'a* :: *field*)
⟨*proof*⟩

lemma *pderiv-power*:
pderiv (*p* ^ *n* :: *'a* :: {*comm-semiring-1*,*semiring-no-zero-divisors*} *poly*) =
of-nat n ∗ *p* ^ (*n* − *1*) ∗ *pderiv p*
⟨*proof*⟩

lemma *pderiv-monom*: *pderiv* (*monom c n*) = *monom* (*of-nat n* ∗ *c*) (*n* − *1*)
⟨*proof*⟩

lemma *degree-div*:
fixes *p* :: *'a* :: *field poly*

assumes $q\ dvd\ p\ p \neq 0$
shows $degree\ p = degree\ (p\ div\ q) + degree\ q$
$\langle proof \rangle$

lemma *order-linear-poly* [*simp*]:
assumes $a \neq 0 \vee b \neq 0$
shows $order\ x\ [:a,\ b:] = (if\ a + b * x = 0\ then\ 1\ else\ 0)$
$\langle proof \rangle$

lemma *smult-sum-left*: $smult\ (\sum x \in A.\ f\ x)\ P = (\sum x \in A.\ smult\ (f\ x)\ P)$
$\langle proof \rangle$

lemma *prod-const-poly*: $(\prod x \in A.\ [:f\ x:]) = [:\prod x \in A.\ f\ x:]$
$\langle proof \rangle$

lemma *prod-uminus*: $(\prod x \in A.\ -f\ x :: 'a :: comm\text{-}ring\text{-}1) = (-1)\ \hat{}\ card\ A * prod\ f\ A$
$\langle proof \rangle$

lemma *poly-pderiv-root*:
fixes $p :: complex\ poly$
assumes $poly\ p\ x = 0$
shows $poly\ (pderiv\ p)\ x = poly\ (p\ div\ [:-x,\ 1:])\ x$
$\langle proof \rangle$

lemma *poly-pderiv-root'*:
fixes $p :: complex\ poly$
assumes $poly\ p\ (1\ /\ x) = 0\ x \neq 0$
shows $poly\ (pderiv\ p)\ (1\ /\ x) = -x * poly\ (p\ div\ [:1,\ -x:])\ (1\ /\ x)$
$\langle proof \rangle$

lemma *degree-prod-eq* [*simp*]:
fixes $f :: 'a \Rightarrow 'b::idom\ poly$
shows $(\bigwedge x.\ x \in S \Longrightarrow f\ x \neq 0) \Longrightarrow degree\ (prod\ f\ S) = (\sum x \in S.\ degree\ (f\ x))$
$\langle proof \rangle$

lemma *partial-fraction-decomposition-fps-of-poly-linear-factors*:
fixes $P\ Q :: complex\ poly$ **and** $A :: complex\ set$ **and** $c :: complex$
defines $Q \equiv smult\ c\ (\prod x \in A.\ [:-x,\ 1:])$
defines $Q' \equiv pderiv\ Q$
assumes *deg*: $degree\ P < card\ A$
assumes $0 \notin A$ **and** [*simp*]: $c \neq 0$ **and** [*simp*]: $finite\ A$
shows $fps\text{-}of\text{-}poly\ P\ /\ fps\text{-}of\text{-}poly\ Q =$
$(\sum c \in A.\ fps\text{-}const\ (poly\ P\ c\ /\ poly\ Q'\ c)\ /\ (fps\text{-}X - fps\text{-}const\ c))$
$\langle proof \rangle$

lemma *partial-fraction-decomposition-fps-of-poly-linear-factors'*:
fixes $P\ Q :: complex\ poly$ **and** $A :: complex\ set$ **and** $c :: complex$
defines $Q \equiv smult\ c\ (\prod x \in A.\ [:1,\ -x:])$

defines $Q' \equiv pderiv\ Q$
assumes deg: $degree\ P < card\ A$
assumes [$simp$]: $0 \notin A$ **and** [$simp$]: $c \neq 0$ **and** [$simp$]: $finite\ A$
shows $fps\text{-}of\text{-}poly\ P\ /\ fps\text{-}of\text{-}poly\ Q =$
$(\sum x{\in}A.\ fps\text{-}const\ (-\ x * poly\ P\ (1\ /\ x)\ /\ poly\ Q'\ (1\ /\ x))\ /\ (1 - fps\text{-}const\ x * fps\text{-}X))$
$\langle proof \rangle$

lemma *partial-fraction-decomposition-fps-of-poly-linear-factors′-monic*:
fixes $P\ Q$ :: $complex\ poly$ **and** A :: $complex\ set$ **and** c :: $complex$
defines $Q \equiv (\prod x{\in}A.\ [:1, -x:])$
defines $Q' \equiv pderiv\ Q$
assumes deg: $degree\ P < card\ A$
assumes [$simp$]: $0 \notin A$ **and** [$simp$]: $finite\ A$
shows $fps\text{-}of\text{-}poly\ P\ /\ fps\text{-}of\text{-}poly\ Q =$
$(\sum x{\in}A.\ fps\text{-}const\ (-\ x * poly\ P\ (1\ /\ x)\ /\ poly\ Q'\ (1\ /\ x))\ /\ (1 - fps\text{-}const\ x * fps\text{-}X))$
$\langle proof \rangle$

1.2 Asymptotics

lemma *asymp-equiv-weaken*:
assumes $(\lambda x.\ f\ x - g\ x) \sim[F]\ (h)$
assumes $h \in o[F](g)$
shows $f \sim[F]\ (g)$
$\langle proof \rangle$

lemma *elt-set-plusI*:
assumes $(\lambda x.\ f\ x - g\ x :: 'a :: ab\text{-}group\text{-}add) \in A$
shows $f =o\ g\ +o\ A$
$\langle proof \rangle$

lemma *elt-set-plus-altdef*:
fixes f :: $\text{-} \Rightarrow 'a :: ab\text{-}group\text{-}add$
shows $f =o\ g\ +o\ A \longleftrightarrow (\lambda x.\ f\ x - g\ x) \in A$
$\langle proof \rangle$

lemma *elt-set-plus-bigo-trans*:
fixes f :: $\text{-} \Rightarrow 'a :: real\text{-}normed\text{-}field$
assumes $f =o\ g\ +o\ O(l)\ g =o\ h\ +o\ O(l)$
shows $f =o\ h\ +o\ O(l)$
$\langle proof \rangle$

lemma *elt-set-plus-bigo-add*:
assumes $f1 =o\ g1\ +o\ O(l)\ f2 =o\ g2\ +o\ O(l)$
shows $(\lambda x.\ f1\ x + f2\ x) =o\ (\lambda x.\ g1\ x + g2\ x)\ +o\ O(l)$
$\langle proof \rangle$

lemma *elt-set-times-bigo*:

assumes $f1 =o\ g1 +o\ O(l1)$ $f2 =o\ g2 +o\ O(l2)$
assumes $(\lambda x.\ l1\ x * g2\ x) \in O(l)$ $(\lambda x.\ g1\ x * l2\ x) \in O(l)$ $(\lambda x.\ l1\ x * l2\ x) \in O(l)$
shows $(\lambda x.\ f1\ x * f2\ x) =o\ (\lambda x.\ g1\ x * g2\ x) +o\ O(l)$
⟨*proof*⟩

lemma *one-over-one-plus-bigo-asymptotics*:
fixes $f\ h :: real \Rightarrow real$
assumes $f =o\ 1 + O(h)$ **and** $h \in o(\lambda\text{-}.\ 1)$
shows $(\lambda x.\ 1\ /\ f\ x) =o\ 1 + O(h)$
⟨*proof*⟩

lemma *one-over-one-plus-bigo-asymptotics′*:
fixes $f\ g\ h :: real \Rightarrow real$
assumes $f =o\ g +o\ O(h)$ $h \in o(g)$ **and** *nz*: *eventually* $(\lambda x.\ g\ x \neq 0)$ *at-top*
shows $(\lambda x.\ 1\ /\ f\ x) \in (\lambda x.\ 1\ /\ g\ x) +o\ O(\lambda x.\ h\ x\ /\ (g\ x)^2)$
⟨*proof*⟩

1.3 Real numbers

lemma *of-real-of-rat* [*simp*]:
of-real (*of-rat* x) = (*of-rat* x :: ′*a* :: {*field-char-0*, *real-div-algebra*})
⟨*proof*⟩

lemma *Rats-abs-int-div-natE*:
assumes $x \in \mathbb{Q}$
obtains $m :: int$ **and** $n :: nat$
where $n \neq 0$ **and** $x =$ *of-int* m / *real* n **and** *coprime* m (*int* n)
⟨*proof*⟩

lemma *cmod-add-real-less*:
assumes $z \notin \mathbb{R}$ **and** $x \neq 0$
shows *norm* (z + *complex-of-real* x) < *norm* $z + |x|$
⟨*proof*⟩

lemma *ln-add1-gt-alt*:
assumes $x > 0$ $x \neq (1 :: real)$
shows $ln\ x > (x - 1)\ /\ x$
⟨*proof*⟩

lemma *ln-add1-gt-alt′*:
assumes $x > (0 :: real)$
shows $ln\ x \geq (x - 1)\ /\ x$
⟨*proof*⟩

lemma *ln-add1-over-self-less*:
fixes $x\ y :: real$
assumes $0 < x$ $x < y$

shows *ln* (*1* + *x*) / *x* > *ln* (*1* + *y*) / *y*
⟨*proof*⟩

lemma *ln-add1-over-self-le*:
fixes *x y* :: *real*
assumes *0* < *x x* ≤ *y*
shows *ln* (*1* + *x*) / *x* ≥ *ln* (*1* + *y*) / *y*
⟨*proof*⟩

lemma *x-add2-powr-le-x-add1-powr-x-add1*:
fixes *x* :: *real*
assumes *x* ≥ *1*
shows (*x*+*2*) *powr x* < (*x*+*1*) *powr* (*x*+*1*)
⟨*proof*⟩

lemma *eventually-at-right-dense*:
assumes *x* < (*y* :: *'a* :: {*linorder-topology*, *dense-order*})
shows *eventually P* (*at-right x*) ⟷ (∃ *b*>*x*. ∀ *y*>*x*. *y* ≤ *b* ⟶ *P y*)
⟨*proof*⟩

lemma *eventually-at-left-dense*:
assumes *x* > (*y* :: *'a* :: {*linorder-topology*, *dense-order*})
shows *eventually P* (*at-left x*) ⟷ (∃ *b*<*x*. ∀ *y*≥*b*. *y* < *x* ⟶ *P y*)
⟨*proof*⟩

lemma *ln-gt-minus-one-over-self*:
assumes (*x* :: *real*) > *0*
shows *ln x* > −*1* / *x*
⟨*proof*⟩

lemma *powr-times-log-less*:
assumes *1* ≤ *x e* ≥ *0 x* < (*y* :: *real*)
shows *x powr e* ∗ *ln x* < *y powr e* ∗ *ln y*
⟨*proof*⟩

lemma *powr-times-log-less'*:
assumes *x* > *0 e* ≥ (*0* ::*real*)
shows *x powr e* ∗ *ln x* < (*x* + *1*) *powr e* ∗ *ln* (*x* + *1*)
⟨*proof*⟩

lemma *ln-over-ln-add1-gt*:
assumes (*x*::*real*) ∈ {*0*<..<*1*}
shows *ln x* / *ln* (*x* + *1*) > (*x* − *1*) ∗ (*x* + *1*) / *x* ^ *2*
⟨*proof*⟩

lemma *ln-over-ln-add-const-less*:
assumes *0* < *x x* < (*y*::*real*) **and** *a*: *a* ≥ *1*
shows *ln x* / *ln* (*x* + *a*) < *ln y* / *ln* (*y* + *a*)
⟨*proof*⟩

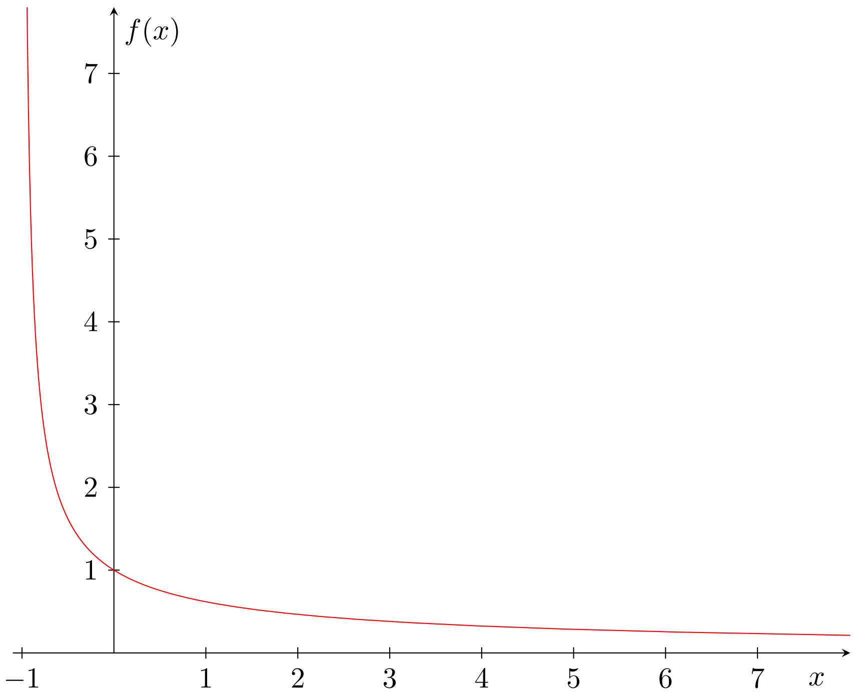

Figure 1: A plot of $f(x)$, the inverse function to $y \mapsto -\ln y/\ln(y+1)$.

end

2 An inverse function for $-\ln x/\ln(x+1)$

theory *HBT-Fun*
imports
  Complex-Main
  HOL−Real-Asymp.Real-Asymp
  Lambert-W.Lambert-W
  HBT-Lemma-Bucket
begin

lemmas [*simp del*] = *div-mult-self1 div-mult-self2 div-mult-self3 div-mult-self4*

In this section we will introduce the function $f(x)$ for $x > 0$ where $f(x)$ is the unique positive real y such that $x = -\ln y/\ln(y+1)$. For a plot of $f(x)$, see Figure 1.

2.1 Definition and basic properties

definition *hbt-f* :: *real* ⇒ *real* **where**

$hbt\text{-}f\ x = (if\ x \le -1\ then\ 1\ else\ (THE\ y.\ y > 0 \wedge -ln\ y\ /\ ln\ (y + 1) = x))$

context
fixes f
defines $f \equiv (\lambda y{::}real.\ -ln\ y\ /\ ln\ (y + 1))$
begin

lemma *hbt-f-exists-aux*:
fixes $x :: real$
assumes $x > -1$
shows $\exists\, y{>}0.\ f\ y = x$
⟨*proof*⟩

lemma *hbt-f-exists*:
fixes $x :: real$
assumes $x > -1$
shows $\exists !y.\ y > 0 \wedge f\ y = x$
⟨*proof*⟩

lemma *hbt-f-correct*: $x > -1 \Longrightarrow f\ (hbt\text{-}f\ x) = x$
⟨*proof*⟩

lemma *hbt-f-pos*: $hbt\text{-}f\ x > 0$
⟨*proof*⟩

lemma *hbt-f-nz* [*simp*]: $hbt\text{-}f\ x \neq 0$
⟨*proof*⟩

lemma *hbt-f-eqI*:
assumes $y > 0\ f\ y = x$
shows $hbt\text{-}f\ x = y$
⟨*proof*⟩

lemma *hbt-f-eqI'*:
assumes $x > -1\ y > 0\ y * (1 + y)\ powr\ x = 1$
shows $hbt\text{-}f\ x = y$
⟨*proof*⟩

lemma *hbt-f-correct'*: $x > 0 \Longrightarrow hbt\text{-}f\ (f\ x) = x$
⟨*proof*⟩

lemma *ln-hbt-f*: $x > -1 \Longrightarrow ln\ (hbt\text{-}f\ x) = -x * ln\ (1 + hbt\text{-}f\ x)$
⟨*proof*⟩

lemma *x-times-ln-hbt-f-plus-one*: $x > -1 \Longrightarrow x * ln\ (1 + hbt\text{-}f\ x) = -ln\ (hbt\text{-}f\ x)$
⟨*proof*⟩

lemma *hbt-f-characteristic-equation*: $x > -1 \Longrightarrow hbt\text{-}f\ x * (1 + hbt\text{-}f\ x)\ powr\ x = 1$

⟨*proof*⟩

lemma *hbt-f-strict-antimono*:
assumes $-1 < x$ $x < y$
shows *hbt-f* $x >$ *hbt-f* y
⟨*proof*⟩

lemma *hbt-f-antimono*:
assumes $-1 < x$ $x \leq y$
shows *hbt-f* $x \geq$ *hbt-f* y
⟨*proof*⟩

lemma *hbt-f-less-iff*: $-1 < x \Longrightarrow -1 < y \Longrightarrow$ *hbt-f* $x <$ *hbt-f* $y \longleftrightarrow x > y$
⟨*proof*⟩

lemma *hbt-f-le-iff*: $-1 < x \Longrightarrow -1 < y \Longrightarrow$ *hbt-f* $x \leq$ *hbt-f* $y \longleftrightarrow x \geq y$
⟨*proof*⟩

lemma *hbt-f-eq-iff*: $-1 < x \Longrightarrow -1 < y \Longrightarrow$ *hbt-f* $x =$ *hbt-f* $y \longleftrightarrow x = y$
⟨*proof*⟩

lemma *hbt-f-gtI*:
assumes $x < f\ y$ $x > -1$
shows *hbt-f* $x > y$
⟨*proof*⟩

lemma *hbt-f-lessI*:
assumes $x > f\ y$ $y > 0$
shows *hbt-f* $x < y$
⟨*proof*⟩

lemma *hbt-f-geI*:
assumes $x \leq f\ y$ $x > -1$
shows *hbt-f* $x \geq y$
⟨*proof*⟩

lemma *hbt-f-leI*:
assumes $x \geq f\ y$ $y > 0$
shows *hbt-f* $x \leq y$
⟨*proof*⟩

lemma *hbt-f-0* [*simp*]: *hbt-f* $0 = 1$
⟨*proof*⟩

lemma *hbt-f-1*: *hbt-f* $1 = (\mathit{sqrt}\ 5 - 1)\ /\ 2$
⟨*proof*⟩

lemma *hbt-f-eq-1-iff* [*simp*]: $x > -1 \Longrightarrow$ *hbt-f* $x = 1 \longleftrightarrow x = 0$
⟨*proof*⟩

lemma *hbt-f-gt-1-iff* [*simp*]: $x > -1 \Longrightarrow hbt\text{-}f\ x > 1 \longleftrightarrow x < 0$
⟨*proof*⟩

lemma *hbt-f-less-1-iff* [*simp*]: $x > -1 \Longrightarrow hbt\text{-}f\ x < 1 \longleftrightarrow x > 0$
⟨*proof*⟩

lemma *hbt-f-ge-1-iff* [*simp*]: $x > -1 \Longrightarrow hbt\text{-}f\ x \geq 1 \longleftrightarrow x \leq 0$
⟨*proof*⟩

lemma *hbt-f-le-1-iff* [*simp*]: $x > -1 \Longrightarrow hbt\text{-}f\ x \leq 1 \longleftrightarrow x \geq 0$
⟨*proof*⟩

lemma *filterlim-hbt-f-at-top*: *filterlim hbt-f* (*at-right 0*) *at-top*
⟨*proof*⟩

2.2 Asymptotics

Using some standard tricks for inverting asymptotic series, we derive the first few terms of the series for $f(x)$ as $x \to \infty$, namely

$$f(x) = \frac{W(x)}{x} - \frac{W(x)}{2x^2(1 + 1/W(x))} + O((\log x)^3/x^3)$$

where $W(x)$ is the Lambert W function, i.e. the unique solution y to $y \cdot e^y = x$.

lemma *hbt-f-asymp-equivI*:
assumes *asymp*: $\bigwedge c'.\ c' \neq c \Longrightarrow (\lambda x.\ f\ (a\ x + c' * b\ x) - x) \sim[at\text{-}top]\ (\lambda x.\ (c - c') * h\ x)$
assumes *h*: *eventually* $(\lambda x.\ h\ x > 0)$ *at-top*
assumes *a*: *eventually* $(\lambda x.\ a\ x > 0)$ *at-top*
assumes *b*: $b \in o(a)$
assumes [*simp*]: $c \neq 0$
shows $(\lambda x.\ hbt\text{-}f\ x - a\ x) \sim[at\text{-}top]\ (\lambda x.\ c * b\ x)$
⟨*proof*⟩

lemma *hbt-f-asymp-equiv-aux*:
assumes *c*: $c \neq 1\ /\ 6$
defines $a \equiv (\lambda u{::}real.\ 1\ /\ u + 1\ /\ (2 * u\ \hat{}\ 2 * (1 + 1\ /\ ln\ u)))$
shows $(\lambda u.\ f\ (a\ u + c * (1\ /\ u\ \hat{}\ 3)) - u * ln\ u) \sim[at\text{-}top]\ (\lambda u.\ (1\ /\ 6 - c) * (ln\ u\ /\ u))$
⟨*proof*⟩

notation *Lambert-W* (*W*)

theorem *hbt-f-asymp-equiv*:
$(\lambda x.\ hbt\text{-}f\ x - W\ x\ /\ x - W\ x\ \hat{}\ 2\ /\ (2 * x\ \hat{}\ 2 * (1 + 1\ /\ W\ x))) \sim[at\text{-}top]$
$(\lambda x.\ 1/6 * ln\ x\ \hat{}\ 3\ /\ x\ \hat{}\ 3)$
⟨*proof*⟩

lemma *hbt-f-asymp-equiv'*:
$(\lambda x.\ hbt\text{-}f\ x - W\ x\ /\ x) \sim[at\text{-}top]\ (\lambda x.\ 1\ /\ 2 * (ln\ x\ \hat{}\ 2\ /\ x\ \hat{}\ 2))$
⟨*proof*⟩

lemma *hbt-f-asymp-equiv''*:
$hbt\text{-}f \sim[at\text{-}top]\ (\lambda x.\ ln\ x\ /\ x)$
⟨*proof*⟩

We also show $\ln f(x) = W(x) + O(\log x/x)$ since we will need this later.

lemma *ln-hbt-f-asymp-equiv*:
$(\lambda x.\ ln\ (hbt\text{-}f\ x) + W\ x) \in O(\lambda x.\ ln\ x\ /\ x)$
⟨*proof*⟩

lemma *ln-hbt-f-asymp-equiv'*: $(\lambda x.\ ln\ (hbt\text{-}f\ x)) \sim[at\text{-}top]\ (\lambda x.\ -ln\ x)$
⟨*proof*⟩

2.3 Non-asymptotic bounds

Lastly, we will show two non-asymptotic bounds, namely that $f(x)$ can be approximated by $e^{-W(x)}$ from below and by $e^{-W(x)} + \frac{1}{2}e^{-2W(x)}$ from above.

lemma *hbt-f-gt-approx*:
assumes *x*: $x > 0$
shows $hbt\text{-}f\ x > exp\ (-W\ x)$
⟨*proof*⟩

lemma *hbt-f-lt-approx-aux1*:
fixes $x :: real$
assumes $x > 0$
shows $ln\ (1 + x + x\ \hat{}\ 2\ /\ 2) < x$
⟨*proof*⟩

lemma *hbt-f-lt-approx-aux2*:
fixes $x :: real$
assumes $x > 0$
shows $x * (2 + x - x^2)\ /\ (x + 2) < ln\ (1 + x + x^2\ /\ 2)$
⟨*proof*⟩

lemma *hbt-f-lt-approx-aux3*:
fixes $x :: real$
assumes $x > 0$
shows $ln\ x * ln\ (1 + x + x\ \hat{}\ 2\ /\ 2) < x * ln\ (x + x\ \hat{}\ 2\ /\ 2)$
⟨*proof*⟩

lemma *hbt-f-lt-approx*:
assumes $x > 0$
shows $hbt\text{-}f\ x < exp\ (-W\ x) + exp\ (-2 * W\ x)\ /\ 2$
⟨*proof*⟩

end

no-notation *Lambert-W* (*W*)

end

3 Bounds on height-balanced trees

theory *HBT-Bounds*
imports
  HOL−Library.Tree
  Linear-Recurrences.Rational-FPS-Solver
  Linear-Recurrences.Linear-Homogenous-Recurrences
  HBT-Fun
begin

lemmas [*simp del*] = *div-mult-self1 div-mult-self2 div-mult-self3 div-mult-self4*

3.1 Definition

We define height-balanced trees, where the heights of the left and right subtree must not differ by more than some constant d at any node in the tree:

inductive *hbt* :: *nat* ⇒ *'a tree* ⇒ *bool* **for** *d* :: *nat* **where**
  hbt d Leaf
| *hbt d l* ⟹ *hbt d r* ⟹ |*int* (*height l*) − *int* (*height r*)| ≤ *int d* ⟹ *hbt d* (*Node l a r*)

lemma *hbt-Leaf* [*simp*]: *hbt d Leaf*
  ⟨*proof*⟩

lemma *hbt-Node* [*simp*]:
  hbt d (*Node l a r*) ⟷
    hbt d l ∧ *hbt d r* ∧ |*int* (*height l*) − *int* (*height r*)| ≤ *d*
  ⟨*proof*⟩

3.2 A recurrence for the "size vs height" bound

We will now derive a recurrence for a lower bound on the number of leaves in the tree in terms of its height.

locale *height-balanced-tree-bound* =
  fixes *d* :: *nat*
  assumes *d*: *d* > *0*
begin

The following recurrence is a generalised variant of the Fibonacci numbers. Note that the way it is written below, $n-d$ actually corresponds to $\min(0, n-$

d). Another way to write the recurrence is as follows:

$$f(n) = \begin{cases} n+1 & \text{if } n \leq d \\ f(n-1) + f(n-d-1) & \text{otherwise} \end{cases}$$

fun *hbt-lb* :: *nat* ⇒ *nat* **where**
hbt-lb 0 = *1*
| *hbt-lb* (*Suc n*) = *hbt-lb n* + *hbt-lb* (*n* − *d*)

lemma *hbt-lb-base-cases* [*simp*]: *n* ≤ *d* ⟹ *hbt-lb n* = *n* + *1*
⟨*proof*⟩

lemma *mono-hbt-lb*: *m* ≤ *n* ⟹ *hbt-lb m* ≤ *hbt-lb n*
⟨*proof*⟩

We now show that for any height-balanced tree of height h, its number of leaves is at most *hbt-lb h*. Since we will later show that *hbt-lb* grows exponentially, this shows that the heigh is logarithmic in the number of leaves (and therefore also the number of nodes overall).

theorem *hbt-size1-height-bound*:
assumes *hbt d t*
shows *hbt-lb* (*height t*) ≤ *size1 t*
⟨*proof*⟩

Next we show that this lower bound is tight by constructing a sequence of trees $(t_h)_{n\geq 0}$ such that t_h has height h and *hbt-wc h* nodes.

fun *hbt-wc* :: *nat* ⇒ *unit tree* **where**
hbt-wc 0 = *Leaf*
| *hbt-wc* (*Suc n*) = *Node* (*hbt-wc* (*n* − *d*)) () (*hbt-wc n*)

lemma *height-hbt-wc* [*simp*]: *height* (*hbt-wc n*) = *n*
⟨*proof*⟩

lemma *hbt-hbt-wc* [*simp*]: *hbt d* (*hbt-wc n*)
⟨*proof*⟩

lemma *size1-hbt-wc* [*simp*]: *size1* (*hbt-wc n*) = *hbt-lb n*
⟨*proof*⟩

We now show more explicitly that *hbt-lb h* is exactly the minimum number of nodes in any HBT of height h.

definition *trees-of-height* :: *nat* ⇒ *unit tree set*
where *trees-of-height h* = {*t*. *height t* = *h*}

primrec *trees-of-height-upto* :: *nat* ⇒ *unit tree set* **where**
trees-of-height-upto 0 = {*Leaf*}
| *trees-of-height-upto* (*Suc n*) = *Set.insert Leaf*

$((\lambda(l,r).\ Node\ l\ ()\ r)$ ‘ $(trees\text{-}of\text{-}height\text{-}upto\ n \times trees\text{-}of\text{-}height\text{-}upto\ n))$

lemma *finite-trees-of-height-upto* [*intro*]: *finite* (*trees-of-height-upto h*)
⟨*proof*⟩

lemma *trees-of-height-upto-altdef*: $trees\text{-}of\text{-}height\text{-}upto\ h = \{t.\ height\ t \leq h\}$
⟨*proof*⟩

lemma *finite-trees-of-height* [*intro*]: *finite* (*trees-of-height h*)
⟨*proof*⟩

lemma *hbt-lb-altdef*: $hbt\text{-}lb\ h = (MIN\ t \in \{t \in trees\text{-}of\text{-}height\ h.\ hbt\ d\ t\}.\ size1\ t)$
⟨*proof*⟩

end

3.3 A more explicit "size vs height" lower bound

Unfortunately, this recurrence does not have a pleasant closed form. Even its asymptotics are somewhat difficult to write down: the bound grows exponentially in *n*, but the basis of the exponential is, generally, a complicated algebraic real number.

We therefore derive a more palatable, albeit less explicit lower bound next. In particular, we will show that c^n is a lower bound for any real number $c > 1$ with $c^{d+1} \leq c^d + 1$.

context *height-balanced-tree-bound*
begin

We introduce the following *characteristic function* (which also happend to be the characteristic polynomial of the recurrence we just derived):

definition *hbt-charfun* :: *real* ⇒ *real* **where**
$hbt\text{-}charfun\ x = x\ \hat{}\ (d+1) - x\hat{}d - 1$

It is easy to see that this characteristic function is negative between 0 and 1 and equal to -1 at $x = 1$. It is strictly increasing for $x \geq 1$ and tends to infinity as $x \to \infty$.

lemma *hbt-charfun-neg*:
assumes $x \in \{0..1\}$
shows $hbt\text{-}charfun\ x < 0$
⟨*proof*⟩

lemma *hbt-charfun-1* [*simp*]: $hbt\text{-}charfun\ 1 = -1$
⟨*proof*⟩

lemma *filterlim-hbt-charfun*: *filterlim hbt-charfun at-top at-top*
⟨*proof*⟩

lemma *hbt-charfun-mono*:
  assumes $1 \leq x$ $x \leq y$
  shows *hbt-charfun* $x \leq$ *hbt-charfun* y
⟨*proof*⟩

lemma *hbt-charfun-strict-mono*:
  assumes $1 \leq x$ $x < y$
  shows *hbt-charfun* $x <$ *hbt-charfun* y
⟨*proof*⟩

lemma *hbt-charfun-eq-iff*:
  assumes $x \geq 1$ $y \geq 1$
  shows *hbt-charfun* $x =$ *hbt-charfun* $y \longleftrightarrow x = y$
  ⟨*proof*⟩

lemma *hbt-charfun-le-iff* [*simp*]:
  assumes $x \geq 1$ $y \geq 1$
  shows *hbt-charfun* $x \leq$ *hbt-charfun* $y \longleftrightarrow x \leq y$
  ⟨*proof*⟩

lemma *hbt-charfun-less-iff* [*simp*]:
  assumes $x \geq 1$ $y \geq 1$
  shows *hbt-charfun* $x <$ *hbt-charfun* $y \longleftrightarrow x < y$
  ⟨*proof*⟩

lemma *strict-antimono-hbt-charfun-nonpos*:
  assumes *odd d* $y \leq 0$ $x < y$
  shows *hbt-charfun* $x >$ *hbt-charfun* y
⟨*proof*⟩

lemma *inj-on-hbt-charfun-nonpos*:
  assumes *odd d*
  shows *inj-on hbt-charfun* {..*0*}
⟨*proof*⟩

end

Now consider any real number $c > 1$ for which the characteristic function is nonpositive. We will show that c^n is then a lower bound for our recurrence.

locale *hbt-lower-bound-constant* = *height-balanced-tree-bound* +
  fixes *c* :: *real*
  assumes *c*: $c > 1$ *hbt-charfun* $c \leq 0$
begin

First of all, note that $c \leq \sqrt[d]{d+1}$.

lemma *c-less-root*: $c <$ *root d* $(d + 1)$
⟨*proof*⟩

It follows that $c^n \leq n + 1$ for any $n \in [1, d]$:

lemma *c-power-less*:
  assumes $n \in \{1..d\}$
  shows $c \,\hat{}\, n < real\ n + 1$
$\langle proof \rangle$

By a simple induction, it then follows that c^n is indeed a lower bound.

lemma *hbt-lb-ge*: $real\ (hbt\text{-}lb\ n) \geq c \,\hat{}\, n$
$\langle proof \rangle$

Putting everything together, we obtain that the height of a generalised AVL tree is at most logarithmic in its number of leaves:

theorem *hbt-height-bound*:
  assumes *hbt d t*
  shows $height\ t \leq log\ c\ (size1\ t)$
$\langle proof \rangle$

end

Next, we derive the optimal value for c, namely the unique positive real root of the characteristic function. We call this the characteristic constant for HBTs with height bound d and write it as C_d.

This constant can be approximated easily in practice using Newton's method: Applying it to the polynomial gives successively more accurate upper bounds for c, while applying it to the reflected polynomial gives successively better lower bounds.

definition *hbt-const* :: *nat* ⇒ *real* **where**
  $hbt\text{-}const\ d = (THE\ x.\ x \geq 0 \wedge x \,\hat{}\, (d+1) - x \,\hat{}\, d - 1 = 0)$

context *height-balanced-tree-bound*
begin

lemma *hbt-const*: $hbt\text{-}const\ d > 1$ $hbt\text{-}charfun\ (hbt\text{-}const\ d) = 0$
  and *hbt-const-unique*: $hbt\text{-}charfun\ x = 0 \Longrightarrow x \geq 0 \Longrightarrow hbt\text{-}const\ d = x$
$\langle proof \rangle$

Note that we can express the height bound d easily in terms of C_d.

lemma *d-conv-hbt-const*:
  $real\ d = -ln\ (hbt\text{-}const\ d - 1)\ /\ ln\ (hbt\text{-}const\ d)$
$\langle proof \rangle$

lemma *hbt-const-gtI*: $hbt\text{-}charfun\ x < 0 \Longrightarrow hbt\text{-}const\ d > x$
  $\langle proof \rangle$

lemma *hbt-const-lessI*:
  assumes $hbt\text{-}charfun\ x > 0$ $x \geq 0$
  shows $hbt\text{-}const\ d < x$
$\langle proof \rangle$

lemma *hbt-const-less-2*: *hbt-const d < 2*
⟨*proof*⟩

The following lower bound is obtained from applying Newton's method to the reflected polynomial once:

theorem *hbt-const-gt*: *hbt-const d > 1 + 1 / (d + 1)*
⟨*proof*⟩

lemma *hbt-charfun-root-imp-irrational*:
assumes *hbt-charfun x = 0*
shows *x* ∉ ℚ
⟨*proof*⟩

lemma *hbt-const-irrational*: *hbt-const d* ∉ ℚ
⟨*proof*⟩

sublocale *opt*: *hbt-lower-bound-constant d hbt-const d*
⟨*proof*⟩

end

Normal AVL trees are HBTs with $d = 1$, in which case the recurrence becomes the Fibonacci numbers and the characteristic constant is the golden ratio.

lemma *hbt-const-1*: *hbt-const (Suc 0) = (1 + sqrt 5) / 2*
⟨*proof*⟩

3.4 Basic asymptotics of the characteristic constant

We will now analyse how this constant c behaves for large values of d. In fact, we will show that $c \sim 1 + \frac{\ln d}{d}$.

lemma *eventually-hbt-const-gt*:
assumes *e*: *e < 1*
shows *eventually (λd. hbt-const d > 1 + e * ln d / d) at-top*
⟨*proof*⟩

lemma *eventually-hbt-const-less*:
assumes *e*: *e > 1*
shows *eventually (λd. hbt-const d < 1 + e * ln d / d) at-top*
⟨*proof*⟩

theorem *hbt-const-asymptotics*: *(λd. hbt-const d − 1) ∼[at-top] (λd. ln d / d)*
⟨*proof*⟩

3.5 More on the characteristic polynomial

We now consider at the characteristic function $\chi_d(X) = X^{d+1} - X^d + 1$ as a complex polynomial and take a closer look at its roots.

context *height-balanced-tree-bound*
begin

definition *hbt-lb-charpoly-coeffs* :: *complex list* **where**
hbt-lb-charpoly-coeffs = [−*1*] @ *replicate* (*d* − *1*) *0* @ [−*1*, *1*]

definition *hbt-lb-charpoly* :: *complex poly* **where**
hbt-lb-charpoly = *Poly hbt-lb-charpoly-coeffs*

lemma *length-hbt-lb-charpoly-coeffs* [*simp*]: *length hbt-lb-charpoly-coeffs* = *d* + *2*
⟨*proof*⟩

lemma *nth-hbt-lb-charpoly-coeffs* [*simp*]:
hbt-lb-charpoly-coeffs ! *0* = −*1*
hbt-lb-charpoly-coeffs ! *d* = −*1*
hbt-lb-charpoly-coeffs ! *Suc d* = *1*
k ∈ {*0*<..<*d*} ⟹ *hbt-lb-charpoly-coeffs* ! *k* = *0*
⟨*proof*⟩

lemma *hbt-lb-charpoly-coeffs-zero-iff* [*simp*]: *k* ≤ *Suc d* ⟹ *hbt-lb-charpoly-coeffs* ! *k* = *0* ⟷ *k* ∈ {*0*<..<*d*}
⟨*proof*⟩

lemma *hbt-lb-charpoly-altdef*:
hbt-lb-charpoly = *monom 1* (*d*+*1*) − *monom 1 d* − *1*
⟨*proof*⟩

lemma *poly-hbt-lb-charpoly-of-real* [*simp*]:
poly hbt-lb-charpoly (*of-real x*) = *hbt-charfun x*
⟨*proof*⟩

lemma *hbt-lb-charpoly-nz* [*simp*]: *hbt-lb-charpoly* ≠ *0*
⟨*proof*⟩

lemma *degree-hbt-lb-charpoly* [*simp*]: *degree hbt-lb-charpoly* = *d* + *1*
⟨*proof*⟩

Since $\chi_d(X)$ shares no roots with its derivative, it is squarefree and all its roots have multiplicity 1.

lemma *rsquarefree-hbt-lb-charpoly*: *rsquarefree hbt-lb-charpoly*
⟨*proof*⟩

3.5.1 The other real root

We have already shown that $\chi_d(X)$ has exactly one positive real root and derived various bounds on it. We now show that

- if d is even, this is the only real root
- if d is odd, there is precisely one additional negative root x_0' with $-1 < x_0' \leq \frac{1-\sqrt{5}}{2} < 1$

lemma *even-imp-no-neg-root*:
  assumes *even d hbt-charfun x = 0 x ≤ 0*
  shows *False*
⟨*proof*⟩

lemma *neg-root-exists*:
  assumes *odd d*
  shows ∃*x. x < 0* ∧ *x > −1* ∧ *x ≤ (1 − sqrt 5) / 2* ∧ *hbt-charfun x = 0*
⟨*proof*⟩

lemma *neg-root-unique*:
  assumes *odd d*
  shows ∃!*x. x ≤ 0* ∧ *hbt-charfun x = 0*
⟨*proof*⟩

3.5.2 The nonreal complex roots

Since $\chi_d(X)$ has *d + 1* complex roots in total and we have already handled the 1 (resp. 2) real roots if d is even (resp. odd), there are $2\lfloor\frac{d}{2}\rfloor$ non-real roots. Since $\chi_d(X)$ has real coefficients, these come in pairs of conjugates (a fact that we do not need and therefore will not prove).

What is important is that these roots all lie within an open disc around the origin with radius C_d, i. e. the unique positive root C_d is the one with the largest absolute value and therefore the dominant one.

lemma *complex-root-norm-less*:
  assumes *poly hbt-lb-charpoly z = 0 z* ∉ ℝ
  shows *norm z < hbt-const d*
⟨*proof*⟩

3.5.3 Summary

We now define the set of complex roots of $\chi_d(X)$ and put all the previous results together to classify the roots of $\chi_d(X)$.

definition *roots* **where** *roots = {z::complex. poly hbt-lb-charpoly z = 0}*

definition *other-real-root* **where** *other-real-root = (THE x. x ≤ 0* ∧ *hbt-charfun x = 0)*

This set is either empty or a singleton, depending on the parity of *d*.

definition *other-real-roots* **where** *other-real-roots* = $\{x.\ x \leq 0 \wedge$ *hbt-charfun* $x = 0\}$

definition *complex-roots* **where** *complex-roots* = $\{z.\ z \notin \mathbb{R} \wedge$ *poly hbt-lb-charpoly* $z = 0\}$

lemma *other-real-rootsD*:
  assumes $x \in$ *other-real-roots*
  shows *odd d* $x < 0$ $x > -1$ $x \leq (1 - sqrt\ 5)\ /\ 2$
⟨*proof*⟩

lemma *roots-decompose*:
  roots = *Set.insert* (*of-real* (*hbt-const d*)) (*of-real* ‘ *other-real-roots* ∪ *complex-roots*)
  (**is** - = *?rhs*)
⟨*proof*⟩

lemma *finite-roots* [*intro*]: *finite roots*
  ⟨*proof*⟩

lemma *finite-complex-roots* [*simp*]: *finite complex-roots*
  ⟨*proof*⟩

lemma *finite-other-real-roots* [*simp*]: *finite other-real-roots*
  ⟨*proof*⟩

lemma *zero-not-in-roots*: $0 \notin$ *roots*
  ⟨*proof*⟩

lemma *one-not-in-roots*: $1 \notin$ *roots*
  ⟨*proof*⟩

lemma *card-roots* [*simp*]: *card roots* = $d + 1$
  ⟨*proof*⟩

lemma *roots-irrational*: *roots* $\cap\ \mathbb{Q} = \{\}$
⟨*proof*⟩

3.6 A closed form for the “size vs height” bound

We now employ the theory of linear recurrences and rational generating functions to derive the asymptotics of *hbt-lb*.

First, we define the generating function of *hbt-lb*.

definition *hbt-lb-fps* :: *complex fps* **where**
  hbt-lb-fps = *Abs-fps* (*of-nat* ∘ *hbt-lb*)

The following polynomials are the numerator and denominator of the generating function.

definition *hbt-lb-fps-num* :: *complex poly* **where**
hbt-lb-fps-num = *lhr-fps-numerator 0 hbt-lb-charpoly-coeffs* (*of-nat* ∘ *hbt-lb*)

definition *hbt-lb-fps-denom* :: *complex poly* **where**
hbt-lb-fps-denom = *lr-fps-denominator hbt-lb-charpoly-coeffs*

The denominator of the generating function is the reflection of $\chi_d(X)$, namely $1 - X - X^{d+1}$:

lemma *hbt-lb-fps-denom-altdef1*:
hbt-lb-fps-denom = *Poly* ([*1*, −*1*] @ *replicate* (*d* − *1*) *0* @ [−*1*])
⟨*proof*⟩

lemma *hbt-lb-fps-denom-altdef2*: *hbt-lb-fps-denom* = *1* − *monom 1 1* − *monom 1* (*d*+*1*)
⟨*proof*⟩

lemma *reflect-poly-hbt-lb-fps-denom* [*simp*]:
reflect-poly hbt-lb-fps-denom = *hbt-lb-charpoly*
⟨*proof*⟩

lemma *hbt-lb-fps-denom-conv-roots*: *hbt-lb-fps-denom* = (∏ *c*∈*roots*. [:*1*, −*c*:])
⟨*proof*⟩

We now use the machinery from the AFP to show that the generating function indeed has this form.

interpretation *rec*: *linear-homogenous-recurrence*
(*of-nat* :: *nat* ⇒ *complex*) ∘ *hbt-lb*
hbt-lb-charpoly-coeffs
map of-nat [*1*..<*d*+*2*]
⟨*proof*⟩

lemma *hbt-lb-fps-altdef*:
hbt-lb-fps = *fps-of-poly hbt-lb-fps-num* / *fps-of-poly hbt-lb-fps-denom*
⟨*proof*⟩

lemma [*cong*]:
m = *n* ⟹ *height-balanced-tree-bound.hbt-lb a m* = *height-balanced-tree-bound.hbt-lb a n*
height-balanced-tree-bound.hbt-lb-charpoly-coeffs a = *height-balanced-tree-bound.hbt-lb-charpoly-coeffs a*
⟨*proof*⟩

lemma *hbt-lb-fps-num-altdef*: *hbt-lb-fps-num* = *Poly* (*replicate* (*d*+*1*) *1*)
⟨*proof*⟩

lemma *hbt-lb-fps-num-altdef2*: *hbt-lb-fps-num* = (∑ *n*≤*d*. *monom 1 n*)
⟨*proof*⟩

lemma *hbt-lb-fps-num-altdef3′*: (*1* − [:*0*, *1*:] ^ *Suc d*) = [:*1*, −*1*:] ∗ *hbt-lb-fps-num*

and *hbt-lb-fps-num-altdef3*: *hbt-lb-fps-num* = (*1* − [:*0*, *1*:] ^ *Suc d*) *div* ([:*1*, −*1*:])
⟨*proof*⟩

lemma *poly-hbt-lb-fps-num*:
poly hbt-lb-fps-num x =
(*if x* = *1 then of-nat* (*d* + *1*) *else* (*1* − *x* ^ *Suc d*) / (*1* − *x*))
⟨*proof*⟩

lemma *degree-hbt-lb-fps-num* [*simp*]: *degree hbt-lb-fps-num* = *d*
⟨*proof*⟩

lemma *coprime-hbt-lb-fps*: *coprime hbt-lb-fps-num hbt-lb-fps-denom*
⟨*proof*⟩

end

lemma *hbt-const-strict-antimono*:
assumes *0* < *d1 d1* < *d2*
shows *hbt-const d1* > *hbt-const d2*
⟨*proof*⟩

definition *hbt-lb-coeff-gen* :: *nat* ⇒ *complex* ⇒ *complex* **where**
hbt-lb-coeff-gen d x = *1* / ((*x* − *1*) ∗ (*1* + *of-nat d* ∗ (*1* − *1* / *x*)))

definition *hbt-lb-coeff* :: *nat* ⇒ *real* **where**
hbt-lb-coeff d =
(*let x* = *hbt-const d in 1* / ((*x* − *1*) ∗ (*1* − *ln* (*x* − *1*) / *ln x* ∗ (*1* − *1* / *x*))))

context *height-balanced-tree-bound*
begin

lemma *hbt-lb-coeff-gen-hbt-const*:
defines *x* ≡ *hbt-const d*
shows *hbt-lb-coeff d* = *hbt-lb-coeff-gen d x*
⟨*proof*⟩

lemma *hbt-lb-coeff-gt-1*: *hbt-lb-coeff d* > *1*
⟨*proof*⟩

Finally, we show that the generating function of *hbt-lb* is

$$\sum_z \frac{u(z)}{1 - zX}$$

where z ranges over the roots of the characteristic polynomial and

$$u(z) = \left[(z-1)\left(1 + d\left(1 - \tfrac{1}{z}\right)\right)\right]^{-1}$$

theorem *hbt-lb-fps-eq*:
hbt-lb-fps = ($\sum$ *x*∈*roots*. *fps-const* (*hbt-lb-coeff-gen d x*) / (*1* − *fps-const x* ∗ *fps-X*))
⟨*proof*⟩

Consequently, *hbt-lb n* has the closed form $\sum_z u(z)z^n$.

corollary *hbt-lb-closed-form*:
hbt-lb n = ($\sum$ *x*∈*roots*. *hbt-lb-coeff-gen d x* ∗ *x* ^ *n*)
⟨*proof*⟩

C_d, being the root with the largest absolute value, dominates asymptotically. This gives us the asymptotic estimate $B_d \cdot C_d^n$, where:

$$B_d = u(C_d) = \left[(C_d - 1)\left(1 + d\big(1 - \tfrac{1}{C_d}\big)\right)\right]^{-1}$$

theorem *asymp-equiv-hbt-lb*:
hbt-lb ∼[*at-top*] (λ*n*. *hbt-lb-coeff d* ∗ *hbt-const d* ^ *n*)
⟨*proof*⟩

If we don't care about the coefficient, we get the following "big theta" bound.

corollary *bigtheta-hbt-lb*:
(λ*n*. *real* (*hbt-lb n*)) ∈ Θ(λ*n*. *hbt-const d* ^ *n*)
⟨*proof*⟩

It also follows fairly obviously from these asymptotics that the bound $f(n) \geq C_d^n$ is in fact the best possible lower bound of the form $a\,b^n$.

corollary *hbt-const-optimal*:
fixes *A C* :: *real*
assumes ∀ *n*. *hbt-lb n* ≥ *A* ∗ *C* ^ *n A* > *0*
shows *A* ≤ *1 C* ≤ *hbt-const d*
⟨*proof*⟩

end

3.7 The asymptotics of the characteristic constant and coefficient

notation *Lambert-W* (*W*)

We recall our auxiliary function $f(x)$, which is defined as the unique positive real y such that $-\ln y/\ln(y+x) = x$. We can then write $C_d = 1 + f(d)$.

lemma (**in** *height-balanced-tree-bound*) *hbt-const-conv-hbt-f*: *hbt-const d* = *1* + *hbt-f d*
⟨*proof*⟩

Via the asymptotics we have already derived for $f(x)$, we now find that

$$C_d = 1 + \frac{W(d)}{d} - \frac{W(d)^2}{2d^2(1 + 1/W(d))} + O((\log d)^3/d^3)$$

theorem *hbt-const-asymptotics-precise*:
  ($\lambda d.$ *hbt-const d* − *1* − *W d* / *d* − (*W d*)^2 / (*2* ∗ *d*^{2} ∗ (*1* + *1* / *W d*))) ∼[*at-top*]
    ($\lambda d.$ *1* / *6* ∗ *ln d* ^ *3* / *d* ^ *3*)
⟨*proof*⟩

Next, we turn to the asymptotics of the coefficient B_d. We first introduce the auxiliary function

$$h(x) = \left[x\left(1 - \frac{\ln x}{\ln(x+1)}\left(1 - \frac{1}{x+1}\right)\right)\right]^{-1}$$

and note that $B_d = h(f(d))$.

definition *hbt-h* :: *real* ⇒ *real*
  where *hbt-h x* = *1* / (*x* ∗ (*1* − *ln x* / *ln* (*x* + *1*) ∗ (*1* − *1* / (*x* + *1*))))

lemma (**in** *height-balanced-tree-bound*) *hbt-lb-coeff-conv-hbt-f*:
  hbt-lb-coeff d = *hbt-h* (*hbt-f d*)
  ⟨*proof*⟩

Straightforward asymptotic estimates tell us that $h(x) = \frac{1}{x(1-\ln x)} + O(1/\log x)$ at $x \to 0$.

lemma *hbt-h-asymptotics*: ($\lambda x.$ *hbt-h x* − *1* / (*x* ∗ (*1* − *ln x*))) ∈ *O*[*at-right 0*]($\lambda x.$ *1* / *ln x*)
  ⟨*proof*⟩

lemma *Lambert-W-bigo-at-top*: *Lambert-W* ∈ *O*($\lambda x.$ *ln x*)
  ⟨*proof*⟩

By plugging the asymptotics of *hbt-f* into those of *hbt-h*, we obtain an asymptotic estimate for *hbt-lb-coeff*:

lemma *hbt-lb-coeff-asymptotics-aux*:
  ($\lambda x.$ *hbt-h* (*hbt-f x*)) =*o* ($\lambda x.$ *x* / (*W x* ∗ (*1* + *W x*))) +*o* *O*($\lambda x.$ *1* / *ln x*)
⟨*proof*⟩

theorem *hbt-lb-coeff-asymptotics*:
  hbt-lb-coeff =*o* ($\lambda d.$ *d* / (*W d* ∗ (*1* + *W d*))) +*o* *O*($\lambda d.$ *1* / *ln d*)
⟨*proof*⟩

corollary *hbt-lb-coeff-asymptotics′*: *hbt-lb-coeff* ∼[*at-top*] ($\lambda d.$ *d* / *ln d* ^ *2*)
⟨*proof*⟩

no-notation *Lambert-W* (*W*)

end

References

[1] F. Luccio and L. Pagli. On the height of height-balanced trees. *IEEE Transactions on Computers*, C-25(1):87–90, 1976.