Noname manuscript No.
(will be inserted by the editor)

Proving Divide & Conquer Complexities in Isabelle/HOL

Manuel Eberl

Received: date / Accepted: date

Abstract The Akra–Bazzi method [1], a generalisation of the well-known Master Theorem, is a useful tool for analysing the complexity of Divide & Conquer algorithms. This work describes a formalisation of the Akra–Bazzi method (as generalised by Leighton [14]) in the interactive theorem prover Isabelle/HOL and the derivation of a generalised version of the Master Theorem from it. We also provide some automated proof methods that facilitate the application of this Master Theorem and allow mostly automatic verification of Θ-bounds for these Divide & Conquer recurrences. To our knowledge, this is the first formalisation of theorems for the analysis of such recurrences.

Keywords Isabelle/HOL · Master Theorem · Akra–Bazzi · Divide & Conquer algorithms · Recurrences · Complexity · Landau symbols

1 Introduction

The *Master Theorem* is the textbook method taught in undergraduate algorithms lectures for analysing the asymptotic run-time complexity of many Divide & Conquer algorithms. The general form of the admissible algorithms is several non-recursive base cases and one recursive case, in which a problem of size n is reduced to a fixed number a of subproblems of size n/b, which are then solved recursively and their solutions combined to a solution for the original problem. One simple example is *Merge Sort*: lists of length ≤ 1 are always trivially sorted and thus returned unchanged (the base cases); lists of size ≥ 2 are split into half, each half is sorted recursively, and the two sorted halves are then combined into a single sorted list. The recurrence relation of this algorithm's run-time function T is

$$T(n) = 2T\left(\tfrac{1}{2}n\right) + n$$

and the classical Master Theorem then states $T \in \Theta(n \ln n)$. One problem with this is that, strictly speaking, the recurrence relation for T is actually something like

$$T(n) = T\left(\left\lfloor \tfrac{1}{2}n \right\rfloor\right) + T\left(\left\lceil \tfrac{1}{2}n \right\rceil\right) + n$$

This work was supported by DFG RTG 1480 (PUMA).

Manuel Eberl
Fakultät für Informatik, Technische Universität München, Garching, Germany
E-mail: eberlm@in.tum.de

since one certainly cannot split a list of length 3 into two lists of length 1.5. Intuitively, one may think that the rounding does not change the asymptotic behaviour of the function, seeing as the rounding operations are asymptotically small disturbances. This is, in fact, the case, but proving it is not entirely trivial and is seldom done rigorously in textbooks or lectures, especially the case when both floors and ceilings are used at the same time, as in the example above.[1]

In order to establish a basis for the verified run-time analysis of such algorithms, we wanted to formalise the Master Theorem in the theorem prover Isabelle/HOL. This must, of course, include rigorous handling of rounding operations. We therefore chose not to base our formal proof on any of the literature proofs for the Master Theorem, but to instead prove a generalisation known as the *Akra–Bazzi method* [1], and derive the Master Theorem as a corollary. As a pleasant side effect, this version of the Master Theorem supports much more complex recursion patterns than the classical Master Theorem from the literature.

To make the application of the Master Theorem in the theorem prover almost as simple as on paper, we provided some proof automation machinery that facilitates the definition of functions from Akra–Bazzi-type Divide & Conquer recurrences and allows applying the Master Theorem to them in a mostly automatic way.

To provide some motivation, consider the following two recurrences, which are related to a deterministic selection algorithm and so-called Ham-Sandwich trees. They are far outside the scope of the classical Master Theorem, but their complexities can easily be found and formally proven in Isabelle/HOL with the generalised Master Theorem we formalised:

$$
\begin{aligned}
&f_1(n) = f_1\left(\left\lfloor \tfrac{n}{5} \right\rfloor\right) + f_1\left(\left\lfloor \tfrac{7n}{10} \right\rfloor + 6\right) + \tfrac{12n}{5} && \text{Result: } f_1 \in \Theta(n) \\
&f_2(n) = f_2\left(\left\lfloor \tfrac{n}{2} \right\rfloor\right) + f_2\left(\left\lfloor \tfrac{n}{4} \right\rfloor\right) + 1 && \text{Result: } f_2 \in \Theta(n^{\log_2 \varphi}) \text{ where } \varphi = \tfrac{1+\sqrt{5}}{2}
\end{aligned}
$$

Outline. In Section 2, we will list some related work on the type of recurrences that we focus on in this work. Section 3 then gives some important background information, namely about the notation used in this work and the notions of integration and Landau symbols that are used in the formal proof.

Section 4 contains a formal description of the types of recurrences for which our results – the Akra–Bazzi theorem and the generalised Master theorem – hold, and Sections 5 and 6 state these results. The formal Isabelle/HOL proof of the results is explained in Sec. 7. Section 8 gives a list of the proof automation we developed for the Master theorem as well as a few examples of its application.

Finally, Sec. 9 compares the scope of our version of the Akra–Bazzi theorem to Leighton's and our Master theorem to the 'textbook' version of the Master theorem.

[1] In *Introduction to Algorithms* [7], for example, only the 'floor' case is proven and the 'ceiling' case is stated to be analogous. The question of what happens when both floors and ceilings are used is not addressed.

2 Related Work

The original paper by Akra and Bazzi [1] that introduced the Akra–Bazzi method uses a so-called *order transform* to reduce the problem to a two-dimensional problem. Their version of the method requires very strong assumptions on the parameters of the problem: the recursive definition must be of the following form:

$$f(x) = g(x) + \sum_{i=1}^{k} a_i \cdot f(\lfloor \tfrac{x}{b_i} \rfloor) \qquad (b_i \in \mathbb{N}_{\geq 2},\ g \text{ non-decreasing})$$

In particular, recursive calls like $f(\lceil \frac{x}{2} \rceil)$, $f(\lfloor \frac{2}{3}x \rfloor)$, or $f(\lfloor \frac{1}{3}x \rfloor + 1)$ are not allowed.

Leighton [14] gives a vastly generalised version of the theorem in which the above restrictions on g and the recursive call are weakened greatly; in particular, his version allows small deviations from the linear recursive call, which, among other things, includes rounding and adding constants. Furthermore, his approach is much more direct; he gives a simple inductive proof that we deemed much more amenable to formal verification than the original proof by Akra and Bazzi. The statement of the theorem we proved and its formal proof in Isabelle/HOL are therefore modelled very closely after his; the only differences in the theorem statement are that some assumptions in our version have been weakened slightly (e. g. allowing negative values on some initial segment of the domain).

Based on these two works, Bazzi and Mitter [5] give a version of the Akra–Bazzi theorem for probabilistic recurrences, where both the factors b_i and the non-linear deviations in the recursive calls are random variables with some restrictions. The theorem then determines the asymptotic growth of the expectation of the function thus defined in a way that is very similar to the theorem given by Leighton.

Drmota and Szpankowski [8] analyse recurrences of the form

$$f(x) = g(x) + \sum_{i=1}^{m} a_i \cdot f\left(\left\lfloor b_j \cdot x + O\left(x^{1-\varepsilon}\right)\right\rfloor\right) + \sum_{i=1}^{m} \bar{a}_i \cdot f\left(\left\lceil \bar{b}_j \cdot x + O\left(x^{1-\varepsilon}\right)\right\rceil\right)$$

with the additional restriction that the $b_j \cdot x + O(x^{1-\varepsilon})$ are increasing. The class of functions that they consider is therefore smaller than Leighton's and ours. However, while Leighton and we are only interested in finding a Θ-bound for f, Drmota and Szpankowski obtain very precise approximations for f, such as $f(x) = C_2 \log n + C_2' + o(1)$ for explicitly computable constants C_2, C_2'. They also describe the oscillations that arise in f due to the rounding in the recurrence and which are not present in the corresponding continuous recurrences. For the Θ-analysis, these oscillations are irrelevant, since they are asymptotically small.

3 Preliminaries

3.1 Syntactical note

We take some liberties when presenting expressions or theorems from Isabelle/HOL here to increase readability. In particular: type coercions between real numbers and natural numbers are always omitted; schematic variables, which are implicitly universally quantified in Isabelle, are printed with an explicit $\forall$ for the sake of clarity; Isabelle-specific syntax, such as $\{0..{<}1\}$ is replaced with the standard notation $[0;1)$; lists are sometimes implicitly used as sets or as indexed sequences (e. g. as_i for the i-th element of as, starting from 1).

$f \in O(g)$	$\longleftrightarrow$	$\exists c > 0.\ \exists x_0.\ \forall x \geq x_0.\ \lvert f(x)\rvert \leq c \cdot \lvert g(x)\rvert$
$f \in o(g)$	$\longleftrightarrow$	$\forall c > 0.\ \exists x_0.\ \forall x \geq x_0.\ \lvert f(x)\rvert \leq c \cdot \lvert g(x)\rvert$
$f \in \Omega(g)$	$\longleftrightarrow$	$\exists c > 0.\ \exists x_0.\ \forall x \geq x_0.\ \lvert f(x)\rvert \geq c \cdot \lvert g(x)\rvert$
$f \in \omega(g)$	$\longleftrightarrow$	$\forall c > 0.\ \exists x_0.\ \forall x \geq x_0.\ \lvert f(x)\rvert \geq c \cdot \lvert g(x)\rvert$
$f \in \Theta(g)$	$\longleftrightarrow$	$f \in O(g) \land f \in \Omega(g)$

Table 1 Definitions of the five Landau symbols

A function taking some value x and returning some t that may depend on x will be written as $\lambda x.\ t$, following Lambda calculus syntax as opposed to the traditional mathematical syntax $x \mapsto t$. We will occasionally omit the 'λ' when it is clear from the context that we mean a function and what the function variable is, particularly in Landau symbols (e. g. $x \in O(x^2)$ instead of $(\lambda x.\ x) \in O(\lambda x.\ x^2)$).

3.2 Landau symbols

3.2.1 Definition

Before stating the Akra–Bazzi theorem, we shall give the precise definition of the Landau symbols that were used in the Isabelle formalisation. [10] Since there was no suitable formalisation of asymptotic growth in Isabelle/HOL, we created a library of Landau symbols specifically for the formalisation of the Akra–Bazzi method, but with other use cases in mind as well. The definitions we chose differ slightly from those given in *Introduction to Algorithms* by Cormen *et al.* [7]: for $f \in O(g)$ to hold, they require f and g to be positive for sufficiently large inputs, whereas we do not. According to our definitions,

$$f \in O(g) \longleftrightarrow -f \in O(g) \longleftrightarrow f \in O(-g) \longleftrightarrow -f \in O(-g)\ .$$

This choice was made because, in our experience, formal reasoning with Landau symbols – especially automatic reasoning – becomes easier this way.

Table 1 shows the definitions of our Landau symbols. In Isabelle, Landau symbols are defined for functions from any (not necessarily linearly) ordered set to any linearly-ordered field. The restriction to fields was made due to the fact that the existence of multiplicative inverses makes Landau symbols more 'well-behaved', and one may even argue that Landau symbols for functions into e. g. the natural numbers do not make much sense – one will probably always want to view such functions as functions into the reals.

There is one prior formalisation of Landau symbols in Isabelle by Avigad *et al.* [2]. It is related to their proof of the Prime Number theorem. They only defined the symbol 'O', and they did so in the following fashion:

$$O(f) = \{h \mid \exists c.\ \forall x.\ \lvert h(x)\rvert \leq c \cdot \lvert f(x)\rvert\}$$

This definition differs from the commonly used one in so far as this one requires the inequality to hold on all inputs, not just for sufficiently large inputs. If the inputs are natural numbers, the two are almost equivalent, but in the context of the Akra–Bazzi theorem, we also use Landau symbols for functions of type $\mathbb{R} \to \mathbb{R}$ and then these two definitions are quite different. We therefore deemed it necessary to create our own library of Landau symbols.

3.2.2 Decision Procedure

During the process of proving the generalised Master Theorem, and particularly in its applications, we encountered a great number of proof obligations like $x \in O(x^3)$, $x^2 \in o(x^2 \ln x)$, $\ln x \in \omega(\ln\ln x)$, etc. These problems all have the following in common:

- They are of the form $f \in L(g)$, where L is a Landau symbol and $f, g : \mathbb{R} \to \mathbb{R}$ are products of 'elementary building blocks' like the identity function and iterated logarithms.
- The building blocks can be ordered linearly w. r. t. their growth (e.g. x, $\ln x$, $\ln\ln x$, …) in such a way that the growth rate of any positive power of a function in the sequence eclipses that of any power of the subsequent one (e.g. $x^p \in \omega((\ln x)^q)$ for any $p, q \in \mathbb{R}$ with $p > 0$).
- These problems are typically very tedious to prove formally.
- Most mathematicians would dismiss them as trivial and not even bother proving them by hand in a pen-and-paper proof.

The obvious course of action was therefore to develop automation machinery to discharge these proof obligations automatically. In the following, we will sketch the decision procedure we developed for this problem without going into too much detail.

Definition 1 (Families of functions)
We call $\mathcal{F} \subseteq \mathbb{R}^\mathbb{R}$ a *family* of functions if

- $\mathcal{F}$ is closed under multiplication and multiplicative inverse
- each function in $\mathcal{F}$ is positive for all sufficiently large inputs
- $\mathcal{F}$ is linearly ordered in the sense that for any $f, \bar{f} \in \mathcal{F}$, at least one of $f \in o(\bar{f})$, $f \in \omega(\bar{f})$, and $f \in \Theta(\bar{f})$ holds

Examples for such families are $\{\lambda x.\ x^p \mid p \in \mathbb{R}\}$ or $\{\lambda x.\ a^x \mid a \in \mathbb{R}_{>0}\}$.

Definition 2 (Dominating families)
We say that a family $\mathcal{F}$ *dominates* a family $\mathcal{G}$ if

- there exists an $f \in \mathcal{F}$ such that $g \in o(f)$ for all $g \in \mathcal{G}$
- $f(x) \in o(\bar{f}(x))$ implies $f(x) \cdot g(x) \in o(\bar{f}(x) \cdot \bar{g}(x))$ for any $f, \bar{f} \in \mathcal{F}$ and $g, \bar{g} \in \mathcal{G}$

If $\mathcal{F}$ dominates $\mathcal{G}$, we immediately have for all $f, \bar{f} \in \mathcal{F}$ and $g, \bar{g} \in \mathcal{G}$:

$$
\begin{aligned}
f(x) \cdot g(x) \in o(\bar{f}(x) \cdot \bar{g}(x)) &\longleftrightarrow f \in o(\bar{f}) \vee (f \in \Theta(\bar{f}) \wedge g \in o(\bar{g})) \\
f(x) \cdot g(x) \in O(\bar{f}(x) \cdot \bar{g}(x)) &\longleftrightarrow f \in o(\bar{f}) \vee (f \in \Theta(\bar{f}) \wedge g \in O(\bar{g})) \\
f(x) \cdot g(x) \in \Theta(\bar{f}(x) \cdot \bar{g}(x)) &\longleftrightarrow f \in \Theta(\bar{f}) \wedge \bar{g} \in \Theta(\bar{g})
\end{aligned}
$$

In other words: o, O, and Θ on $\mathcal{F} \cdot \mathcal{G}$ behave analogously to $<$, $\leq$, and $=$ on pairs with lexicographic ordering.

Furthermore, it is obvious that if $\mathcal{F}$ dominates $\mathcal{G}$ and $\mathcal{G}$ dominates $\mathcal{H}$, then $\mathcal{F}$ dominates $\mathcal{G} \cdot \mathcal{H}$. This notion of transitivity implies that we can lift the above result on pairs to sequences where each family dominates the next. We can thus reduce any statement of the form

$$f_1(x) \ldots f_n(x) \in L(\bar{f}_1(x) \ldots \bar{f}_n(x)) \qquad \text{(with } L \in \{o, O, \Theta\} \text{ and } f_i, \bar{f}_i \in \mathcal{F}_i\text{)}$$

to a statement involving only Boolean connectives and expressions of the form $f_i \in l(\bar{f}_i)$ for $l \in \{o, O, \Theta\}$.

Of course, this means that if the $\mathcal{F}_i$ are chosen such that we can decide $f_i \in l(\bar{f}_i)$, we can also decide $f \in L(\bar{f})$. In our decision procedure, the admissible function families are powers of iterated logarithms; i. e. for each fixed $k \in \mathbb{N}$, the functions of the form

$$\lambda x.\ (\underbrace{\ln\ \dots\ \ln}_{k\text{ times}} x)^p \qquad \text{(for some } p \in \mathbb{R})$$

form one family. Deciding $f_k \in l(\bar{f}_k)$ for two functions in such a family can then be done easily by comparing the exponents.

Our decision procedure therefore simply analyses a goal like $x \in o(x \ln x)$ and rewrites it to $x^1 (\ln x)^0 \in o(x^1 (\ln x)^1)$. By the above result, this holds iff $1 < 1 \lor (1 = 1 \land 0 < 1)$, which Isabelle's simplifier can easily prove automatically. We integrated this decision procedure into Isabelle's simplifier, so proof obligations of this form will automatically be rewritten to necessary and sufficient conditions containing only Boolean connectives and comparisons on the exponents. Additionally, if these exponents are numeric constants, the conditions are then proven (or disproven) automatically by the simplifier. This made many of our proofs and the application of our main results much easier.

In addition to this decision procedure, we also have simplifier setup that can:

- Simplify terms like $L(f+g)$ to $L(g)$ if $f \in o(g)$ or to $L(f)$ if $g \in o(f)$
- Cancel common factors like h from $f \cdot h \in L(g \cdot h)$ if $h(x)$ is non-zero for large enough x
- Perform simplifications on functions under a Landau symbol that are valid for sufficiently large values, e. g. $\ln(2x) = \ln 2 + \ln x$. (This is not valid in Isabelle/HOL for $x \leq 0$)

We initially formalised the proof of the asymptotic inequalities described in section 7.2 in an elementary way. The resulting proofs were complex and virtually unreadable. After the introduction of Landau symbols and with heavy use of the automation we just described, we were able to more than halve the length of these proofs and make them significantly more readable.

3.3 Integration

Since the statement of the Akra–Bazzi theorem contains an integral, we need to decide on a definition of integration and a formalisation thereof. Isabelle contains a number of different integrals, most notably the *Henstock–Kurzweil integral* (also known as the *Gauge integral*) on functions from Euclidean spaces to normed real vector spaces and the *Bochner integral* (an extension of the Lebesgue integral) on functions from a measure space to the real numbers [3].

In the proof of the Akra–Bazzi theorem, we noticed that the only properties of integration that we actually needed are the following:

Integral of constant functions. if $a \leq b$ and $c \in \mathbb{R}_{\geq 0}$, the constant function $\lambda x.\ c$ is integrable on $[a;b]$ and

$$\int_a^b c\,\mathrm{d}x = (b-a) \cdot c\ .$$

Monotonicity. if f and g are integrable on $[a;b]$ and $f(x) \leq g(x)$ for all $x \in [a;b]$, then

$$\int_a^b f(x)\,\mathrm{d}x \leq \int_a^b g(x)\,\mathrm{d}x\ .$$

Integrability on sub-intervals. if f is integrable on $[a;d]$ and $a \leq b \leq c \leq d$, then f is also integrable on $[b;c]$.

Splitting. if f is integrable on $[a;c]$ and $a \leq b \leq c$, then

$$\int_a^b f(x)\,\mathrm{d}x + \int_b^c f(x)\,\mathrm{d}x = \int_a^c f(x)\,\mathrm{d}x\ .$$

We therefore proved the Akra–Bazzi theorem generically w. r. t. the integral definition: the theorem can be instantiated with any concept of integration and integrability that fulfils the above four properties. We call such an integral *admissible*.

The 'standard' integrals like the Riemann, Lebesgue, Bochner, and Henstock–Kurzweil integrals all fulfil these properties and are therefore admissible, as is the non-negative Lebesgue integral that ignores the negative part of the integrand. Notably, all of these are generalisations of the Riemann integral on non-negative functions. The natural question is then: Are there any admissible 'integrals' that are not integrals in the usual sense, i. e. not merely generalisations of the Riemann integral?

The answer to this is not immediately obvious, but it turns out that any admissible integral $\mathcal{I}$ must coincide with the Riemann integral on any function f that is both piecewise continuous and $\mathcal{I}$-integrable, but it can differ from the Riemann integral and its generalisations on non-piecewise-continuous functions. Since non-piecewise-continuous functions should rarely arise in the context of the Akra–Bazzi theorem, we shall not explore the issue further here; a more detailed explanation and a proof can be found in the appendix.

4 General setting

Let us now set up the context in which the remainder of this work will be set: For our version of the Akra–Bazzi method, we shall consider a recursively-defined function $f : \mathbb{N} \to \mathbb{R}$ with the following properties:

$$\begin{aligned} f(x) &\geq 0 && \text{for all } x \in [x_0;x_1) \\ f(x) &= g(x) + \sum_{i=1}^{k} a_i \cdot f\left(b_i \cdot x + h_i(x)\right) && \text{for all } x \geq x_1 \end{aligned}$$

for a natural number $k \in \mathbb{N} \setminus \{0\}$, a function $g : \mathbb{N} \to \mathbb{R}$, natural numbers $x_0, x_1 \in \mathbb{N}$, real coefficients $a_i \in \mathbb{R}$, $b_i \in \mathbb{R}$, functions $h_i : \mathbb{N} \to \mathbb{R}$ such that:

- $g(x) \geq 0$ for all $x \geq x_1$
- $a_i \geq 0$ for all $i \in [1;k]$ and $a_i > 0$ for at least one $i \in [1;k]$
- $b_i \in (0;1)$ for all $i \in [1;k]$
- for every $i \in [1;k]$, there exists an $\varepsilon > 0$ such that $h_i \in O(x/\ln^{1+\varepsilon} x)$
- $b_i x + h_i(x) \in \mathbb{N}$ and $x_0 \leq b_i \cdot x + h_i(x) < x$ for all $i \in [1;k]$ and all $x \geq x_1$
  (well-definedness of f)

We will now explain the meaning of these variables.

The recursion structure. The parameters x_0, x_1, k, a_i, b_i, and h_i characterise the recursion structure of the function f. To understand the role of the different parameters, it is useful to look at them in the case when f describes the cost of a Divide & Conquer algorithm: the values between x_0 and x_1 are the costs of the base cases; the cost of the recursive case is defined recursively as the sum of the costs of the recursive calls and the costs of combining the results of the calls. Each triple (a_i, b_i, h_i) corresponds to a_i recursive calls of the form $b_i \cdot x + h_i(x)$; the costs of combining the results are represented by the function g.

Variation terms. The h_i represent asymptotically small variation terms in the recursive call, allowing some deviation from the linear term $b_i \cdot x$. This is not merely a nice gimmick – it is actually necessary to have something like this, since, due to the discreteness of the natural numbers, a purely linear term in the recursive call is impossible. This approach covers rounding and other deviations in a uniform way, as opposed to making ad-hoc arguments why certain kinds of rounding do not change the result.

For example, the terms $f(\lfloor \frac{x}{2} \rfloor)$, $f(\lceil \frac{x}{2} \rceil)$, and $f(\lceil \frac{x}{2} \rceil + 42)$ would be admissible, as they can be expressed as $f(b \cdot x + h(x))$ for some $b \in (0;1)$ and some $h : \mathbb{N} \to \mathbb{R}$ where $h \in O(1)$. However, much larger deviations, such as $f(\lfloor \frac{1}{2}n - \sqrt{n} \rfloor)$, are also allowed.

Of course, enough base cases must be provided (i. e. x_0 and x_1 must be chosen large enough and far enough apart) to fulfil the well-definedness conditions; a function 'definition' like $f(x) = f(\lceil \frac{3}{4}x \rceil) + 1$ for $x_1 = 3$ cannot be allowed since $f(3) = f(\lceil \frac{9}{4} \rceil) + 1 = f(3) + 1$ is contradictory.

Leighton [14] mentions that the condition that the h_i be in $O(x/\ln^{1+\varepsilon} x)$ for some $\varepsilon > 0$ is tight in some sense, since the recurrence $f(x) = 2f(x/2 + x/\ln x)$ has the asymptotic growth $x \ln^{\Theta(1)} x$, whereas the recurrence $f(x) = 2f(x/2)$ (i. e. without the variation term) has the growth $\Theta(x)$.

5 The Akra–Bazzi method

Having established the necessary context, we will now present the main theorems of the Akra–Bazzi method for the function f. To do this, we first need to define the characteristic number of an Akra–Bazzi recurrence: The contribution of the recursion structure (without the 'recombination costs' g) to the asymptotic growth can be summarised as a single real number, which we call p. This number is defined implicitly as

$$\sum_{i=1}^{k} a_i \cdot b_i^p = 1$$

If the a_i are not all zero (which we assumed), this equation defines p uniquely. To show this, we consider the function

$$t : \mathbb{R} \to \mathbb{R}_{>0},\ x \mapsto \sum_{i=1}^{k} a_i \cdot b_i^x .$$

This function is continuous and $t(x) \xrightarrow{-\infty} \infty$ and $t(x) \xrightarrow{\infty} 0$. Therefore, by the intermediate value theorem, some p such that $\sum_{i=1}^{k} a_i \cdot b_i^p = 1$ always exists, and since t is also strictly decreasing, this p is unique.

We can now state the three variants of the Akra–Bazzi theorem, which give Ω, O, and Θ bounds on the growth of f:

Theorem 1 (Akra–Bazzi theorem, Ω version)
Fix $\bar{g} : \mathbb{R} \to \mathbb{R}$ with $g \in \Omega(\bar{g})$, i. e. $\bar{g}$ is an asymptotic lower bound for g. Assume that:

- *$f(x) > 0$ for all sufficiently large x*
- *$\bar{g}(x) \geq 0$ for all sufficiently large x*
- *there exist real constants $c > 0$ and C with $\forall i \in [1;k].\ C < b_i$ such that for all sufficiently large x: $\forall u \in [C \cdot x;x].\ \bar{g}(u) \leq c\bar{g}(x)$*
- *$\bar{g}$ is bounded above on every real interval $[a;b]$ with $a \geq a_0$ for some a_0*
- *$\bar{g}(x)/x^{p+1}$ is integrable on any interval $[a;b]$ with $a \geq a_0$ for some a_0*

Then
$$f \in \Omega\left(x^p \left(1 + \int_t^x \frac{\bar{g}(u)}{u^{p+1}} \mathrm{d}u\right)\right)$$
for any sufficiently large t.

Theorem 2 (Akra–Bazzi theorem, O version)
Fix $\bar{g} : \mathbb{R} \to \mathbb{R}$ with $g \in O(\bar{g})$, i. e. $\bar{g}$ is an asymptotic upper bound for g. Assume that:

- *$\bar{g}(x) \geq 0$ for all sufficiently large x*
- *there exist real constants $c > 0$ and C with $\forall i \in [1;k].\ C < b_i$ such that for all sufficiently large x: $\forall u \in [C \cdot x;x].\ \bar{g}(u) \geq c\bar{g}(x)$*
- *$\bar{g}(x)/x^{p+1}$ is integrable on any real interval $[a;b]$ with $a \geq a_0$ for sufficiently large a_0*

Then
$$f \in O\left(x^p \left(1 + \int_t^x \frac{\bar{g}(u)}{u^{p+1}} \mathrm{d}u\right)\right)$$
for any sufficiently large t.

Combining these two results yields:

Theorem 3 (Akra–Bazzi theorem, Θ version)
Fix $\bar{g} : \mathbb{R} \to \mathbb{R}$ with $g \in \Theta(\bar{g})$, i. e. $\bar{g}$ has the same asymptotic growth as g. Assume that:

- *$f(x) > 0$ for all sufficiently large x*
- *$\bar{g}(x) \geq 0$ for all sufficiently large x*
- *there exist real constants $c_1, c_2 > 0$ and C with $\forall i \in [1;k].\ C < b_i$ such that for all sufficiently large x: $\forall u \in [Cx;x].\ c_1\bar{g}(x) \leq \bar{g}(u) \leq c_2\bar{g}(x)$*
- *$\bar{g}$ is bounded above on every real interval $[a;b]$ with $a \geq a_0$ for some a_0*
- *$\bar{g}(x)/x^{p+1}$ is integrable on any real interval $[a;b]$ with $a \geq a_0$ for some a_0*

Then
$$f \in \Theta\left(x^p \left(1 + \int_t^x \frac{\bar{g}(u)}{u^{p+1}} \mathrm{d}u\right)\right)$$
for any sufficiently large t.

The restrictions on $\bar{g}$ are somewhat technical, especially the ones of the form $\exists c_1 > 0.\ \forall u \in [Cx;x].\ c_1\bar{g}(x) \leq \bar{g}(u)$. Leighton [14] calls these the *polynomial-growth conditions*[2] and also asserts that if $|\bar{g}'|$ is upper-bounded by a polynomial, the conditions always hold. This is incorrect, since e. g. $g(x) = 1 + \sin(x)$ is non-negative and the absolute of its derivative is upper-bounded by 1, but it does not fulfil either of the two polynomial-growth conditions.

Nevertheless, the most interesting cases are those where $\bar{g}(x)$ is of the form $x^r \ln^s x$, and as Leighton also remarks, these functions always satisfy the polynomial-growth conditions. Restricting $\bar{g}$ to this form, which we shall do in the next section, will lead us to a specialisation of the Akra–Bazzi method that is very close to the well-known Master Theorem.

Let us now analyse the conclusion of the last Akra–Bazzi theorem more closely in an informal way: Expanding the product inside the Θ yields

$$f \in \Theta\left(x^p\right) + \Theta\left(x^p \int_t^x \frac{\bar{g}(u)}{u^{p+1}}\,\mathrm{d}u\right) .$$

Clearly, the x^p in the left summand is independent from $\bar{g}$ and would still be present even for $\bar{g} = 0$. The $\Theta(x^p)$ can therefore be seen as the inherent cost of the recursion itself, which depends only on p, which in turn is determined uniquely by the a_i and b_i. The term with the integral on the right, on the other hand, also depends on the recombination costs $\bar{g}(x)$, and it is big whenever $\bar{g}(x)$ is big.

It is also clear that the values of the base cases are completely irrelevant (as long as they are non-negative).

6 The Master Theorem

If we look at a restricted class of functions $\bar{g}$, we can make the last two observations of the previous section a bit more precise: If $\bar{g}(x)$ is of the form x^q for some $q \in \mathbb{R}$, the conditions on $\bar{g}$ (non-negativity, polynomial growth, boundedness, integrability) are all satisfied. If the other conditions for the Akra–Bazzi theorem are satisfied, we have:

$$f \in \Theta\left(x^p\right) + \Theta\left(x^p \int_t^x u^{q-p-1}\mathrm{d}u\right) = \begin{cases} \Theta\left(x^p\right) + \Theta\left(x^q\right) & = \Theta\left(x^p\right) & \text{for } q < p \\ \Theta\left(x^p\right) + \Theta\left(x^p \ln x\right) & = \Theta\left(x^p \ln x\right) & \text{for } q = p \\ \Theta\left(x^p\right) + \Theta\left(x^q\right) & = \Theta\left(x^q\right) & \text{for } q > p \end{cases}$$

The three cases differ in how high the inherent costs of the recursion are compared to the recombination costs. In the first case, the recombination costs are smaller than the recursion costs, which means that most of the work is done at the bottom of the recursion tree, since there are many leaves, but recombining them is cheap (‘bottom-heavy recursion’). In the third case, the recombination costs dominate and most of the work will be done recombining the results near the top of the recursion tree (‘top-heavy recursion’). In the second case, the recursion costs and the recombination costs have a similar rate of growth (‘balanced recursion’). This case can be generalised further by considering $\bar{g}(x) = x^p \ln^q x$.

[2] His conditions are slightly more restrictive; among other things, he requires them to hold for all $x \geq 1$.

This leads to our generalised Master Theorem:

Corollary 1 (Master Theorem)

Bottom-heavy recursion.
If $g \in O(x^q)$ for some $q < p$, then $f \in O(x^p)$. If, additionally, $f(x)$ is positive for all sufficiently large x, we even have $f \in \Theta(x^p)$.[3]

Balanced recursion.
If $g \in \Theta(x^p \ln^q x)$ for some q, then

$$f \in \begin{cases} \Theta(x^p) & \text{if } q < \text{-1} \\ \Theta(x^p \ln\ln x) & \text{if } q = \text{-1} \\ \Theta(x^p \ln^{q+1} x) & \text{if } q > \text{-1} \end{cases}$$

Top-heavy recursion.
If $g \in \Theta(x^q)$ for some $q > p$, then $f \in \Theta(x^q) = \Theta(g)$.

7 Proving the Akra–Bazzi theorem and the Master theorem

We shall now describe the proof of our version of the Akra–Bazzi theorem. Some parts of the proof are very technical; in these parts of the proof, we shall attempt to provide the reader with a good high-level understanding of what must be proven and how we proved it without mentioning too many details. For more details, we refer the reader to the formal proof development in the *Archive of Formal Proofs* [9] or, where applicable, Leighton's proof [14]. In any case, we recommend that readers familiarise themselves with Leighton's proof before attempting to understand ours.

7.1 Formal setting

First of all, we will explain how the conditions mentioned in Section 4 are stated formally in Isabelle/HOL: We use a *locale* [4] called *akra_bazzi_function*. A locale is a named context that contains fixed variables, assumptions, and definitions. Such a locale can be *instantiated* by providing values for the fixed variables and proving that its assumptions hold for these values. Instantiating a locale gives the user access to all the facts that were proven in this locale, specialised to the specific values that it was instantiated with. Figure 1 shows the definition of the locale *akra_bazzi_function*, modulo some insignificant notational adjustments.

The only difference to the conditions stated in Section 4 is that the recursive calls are of the shape $f(ts_i(x))$ instead of $f\ (b_i \cdot x + h_i\ x)$. The reason for this is that the latter would require expressing a call like $f(\lfloor \frac{1}{2}x \rfloor)$ in the rather awkward form $f(\frac{1}{2}x + (\lfloor \frac{1}{2}x \rfloor - \frac{1}{2}x))$, whereas the former is more direct.

[3] Note that due to the other constraints on f and g, the condition that $f(x)$ is positive for all sufficiently large x must hold if either $g(x)$ is positive for all sufficiently large x or $f(x)$ is positive in all the base cases, i. e. for $x \in [x_0; x_1)$.

locale akra_bazzi_function =
  fixes $x_0\ x_1\ k$ :: nat **and** $as\ bs$ :: real list **and** ts :: (nat $\Rightarrow$ nat) list **and**
    f :: nat $\Rightarrow$ real **and** g :: nat $\Rightarrow$ real
  assumes $k \neq 0$ **and** length$(as) = k$ **and** length$(bs) = k$ **and** length$(ts) = k$
  and $\forall a \in as.\ a \geq 0$ **and** $\forall b \in bs.\ b \in (0;1)$ **and** $\exists i \in [1;k].\ as_i > 0$
  and $\forall i \in [1;k].$ akra_bazzi_term(x_0, x_1, bs_i, ts_i)
  and $\forall x \in [x_0;x_1).\ f(x) \geq 0$
  and $\forall x \geq x_1.\ f(x) = g(x) + \sum_{i=1}^{k} as_i \cdot f(ts_i(x))$
  and $\forall x \geq x_1.\ g(x) \geq 0$

Figure 1 The locale *akra_bazzi_function* that formally captures the conditions imposed upon the recursively-defined function f

The conditions on the recursive calls are replaced by the condition that all the ts_i be Akra–Bazzi terms, where

definition akra_bazzi_term(x_0, x_1, b, t) =
$$(\exists \varepsilon\ h.\ \varepsilon > 0 \land h \in O(\lambda x.\ x/\ln^{1+\varepsilon} x) \land$$
$$(\forall x \geq x_1.\ t(x) \geq x_0 \land t(x) < x \land b \cdot x + h(x) = t(x)))\ .$$

One can then easily prove introduction rules to discharge this condition for specific forms of recursive calls, e. g.

lemma akra_bazzi_term_ceiling:
  assumes $b > 0$ **and** $b < 1$ **and** $x_0 \leq b \cdot x_1$ **and** $(1-b) \cdot x_1 \geq 1$
  shows akra_bazzi_term$(x_0, x_1, b, \lambda x.\ \lceil b \cdot x \rceil)$

Provided that such rules exist for every recursive call occurring in the recursive equation of f, this condition and most of the other locale assumptions contain only the constants *as*, *bs*, *k*, *x0*, and *x1* and can therefore be solved by simple evaluation for a concrete function f with concrete values for *as*, *bs*, etc. The remaining conditions are:

1. $\forall x \in [x_0;x_1).\ f(x) \geq 0$
2. $\forall x \geq x_1.\ g(x) \geq 0$
3. $\forall x \geq x_1.\ f(x) = g(x) + \sum_{i=1}^{k} as_i \cdot f\ (ts_i(x))$

These conditions must be shown by the user, but they are typically direct consequences from the definitions of f and g.

7.2 Asymptotic estimates

We now move on to the actual proofs. First of all, we need to prove a number of asymptotic inequalities, i. e. inequalities that hold whenever x is large enough. Leighton mentions the first four of these on page 5, but does not provide any proof (he mentions that they can be proven using 'standard Taylor series expansions and asymptotic analysis'). Apart from these, we found that we need four more inequalities that ensure that conditions like $b_i x + h_i(x) < x$ and $1 - \ln(b_i x + h_i(x))^{-\varepsilon/2} > 0$ will hold for all relevant inputs x. Without these conditions, several terms that occur in the proof of the Akra–Bazzi theorem would not even be well-defined.

Lemma 1 (Asymptotic inequalities) *For any $H, \varepsilon \in \mathbb{R}_{>0}$, $b \in (0;1)$, and $p \in \mathbb{R}$, there exists some $x_0 \in \mathbb{R}$ such that the following inequalities hold for all $x \geq x_0$:*

$$\left(1 \pm \frac{H}{b\ln^{1+\varepsilon} x)}\right)^p \left[1 + \left(\ln^{-\varepsilon/2}\left(bx + \frac{Hx}{\ln^{1+\varepsilon} x}\right)\right)\right] \geq 1 + \ln^{-\varepsilon/2} x \tag{1}$$

$$\left(1 \pm \frac{H}{b\ln^{1+\varepsilon} x}\right)^p \left[1 - \left(\ln^{-\varepsilon/2}\left(bx + \frac{Hx}{\ln^{1+\varepsilon} x}\right)\right)\right] \leq 1 - \ln^{-\varepsilon/2} x \tag{2}$$

$$\frac{1}{2}\left(1 + \ln^{-\varepsilon/2} x\right) \leq 1 \tag{3}$$

$$2\left(1 - \ln^{-\varepsilon/2} x\right) \geq 1 \tag{4}$$

$$\left[\ln\left(bx - \frac{Hx}{\ln^{1+\varepsilon} x}\right)\right]^{-\varepsilon/2} < 1 \tag{5}$$

$$\frac{H}{\ln^{1+\varepsilon} x} < \frac{b}{2} \tag{6}$$

$$\frac{H}{\ln^{1+\varepsilon} x} < \frac{1-b}{2} \tag{7}$$

$$x\left(1 - b - \frac{H}{\ln^{1+\varepsilon} x}\right) > 1 \tag{8}$$

(The $\pm$ means that the inequality must hold both for a $+$ and for a $-$ in its place.)

Proof All but the first two of these inequalities are trivial and can be proven by comparing the limits of the left-hand side and the right-hand side; the first two, however, require non-trivial asymptotic analysis using Taylor series expansions. Since these two inequalities are a crucial ingredient in the proof of this generalised Akra–Bazzi theorem, we will briefly sketch the proof of (1). (The proof of (2) is mostly analogous.)

The key ingredient in proving the inequality is the Taylor series expansion

$$(1 \pm t(x))^y = 1 \pm yt(x) + O\left(t(x)^2\right) = 1 + O(t(x)) \qquad \text{if } \lim_{x\to\infty} t(x) = 0$$

In the following, we will indicate such an expansion with the symbol $\overset{\text{Taylor}}{=}$ and a curly bracket that denotes which term is taken to be $t(x)$ in the expansion.

First of all, we estimate the first factor on the left-hand side with[4]

$$\left(1 \pm \overbrace{\frac{H}{b\ln^{1+\varepsilon} x}}^{t(x)}\right)^p \overset{\text{Taylor}}{=} 1 + O\left(\ln^{-1-\varepsilon} x\right) = 1 + o\left(\ln^{-1-\varepsilon/2} x\right)$$

[4] The notation here becomes a bit informal. Terms like $f(x) + O(g(x))$ stand for the set $\{f(x) + h(x) \mid h(x) \in O(g(x))\}$ and all equality symbols are then essentially set inclusions, i. e. $f(x) + O(\ldots) = g(x) + O(\ldots)$ means that the left-hand side is a subset of the right-hand side.

Moreover, we have in the second factor:

$$\begin{aligned}
\ln^{-\varepsilon/2}\left(bx+\frac{Hx}{\ln^{1+\varepsilon}x}\right) &= \left[\ln\left(bx\left(1+\frac{H}{b\ln^{1+\varepsilon}x}\right)\right)\right]^{-\varepsilon/2} = \\
&= \left[\ln bx+\ln\left(1+\frac{H}{b\ln^{1+\varepsilon}x}\right)\right]^{-\varepsilon/2} = \\
&= \left(\ln^{-\varepsilon/2}bx\right)\left[1+\underbrace{\frac{1}{\ln bx}\ln\left(1+\frac{H}{b\ln^{1+\varepsilon}x}\right)}_{t(x)}\right]^{-\varepsilon/2} \overset{\text{Taylor}}{=} \\
&= \left(\ln^{-\varepsilon/2}bx\right)\left[1+O\left(\frac{1}{\ln bx}\ln\left(1+\frac{H}{b\ln^{1+\varepsilon}x}\right)\right)\right] = \\
&= \left(\ln^{-\varepsilon/2}bx\right)+\left(\ln^{-1-\varepsilon/2}bx\right)O\left(\ln\left(1+\frac{H}{b\ln^{1+\varepsilon}x}\right)\right) = \\
&= \left(\ln^{-\varepsilon/2}bx\right)+\left(\ln^{-1-\varepsilon/2}bx\right)o(1) = \\
&= \left(\ln^{-\varepsilon/2}bx\right)+o\left(\ln^{-1-\varepsilon/2}x\right)
\end{aligned}$$

Combining these two asymptotic estimates, we have:

$$\begin{aligned}
&\left(1\pm\frac{H}{b\ln^{1+\varepsilon}x}\right)^p\left[1+\left(\ln^{-\varepsilon/2}\left(bx+\frac{Hx}{\ln^{1+\varepsilon}x}\right)\right)\right] = \\
&\quad = \left[1+o\left(\ln^{-1-\varepsilon/2}x\right)\right]\left[1+\left(\ln^{-\varepsilon/2}bx\right)+o\left(\ln^{-1-\varepsilon/2}x\right)\right] = \\
&\quad = 1+(\ln bx)^{-\varepsilon/2}+o\left(\ln^{-1-\varepsilon/2}x\right) = \\
&\quad = 1+(\ln b+\ln x)^{-\varepsilon/2}+o\left(\ln^{-1-\varepsilon/2}x\right) = \\
&\quad = 1+\left(\ln^{-\varepsilon/2}x\right)\left(1+\underbrace{\frac{\ln b}{\ln x}}_{t(x)}\right)^{-\varepsilon/2}+o\left(\ln^{-1-\varepsilon/2}x\right) \overset{\text{Taylor}}{=} \\
&\quad = 1+\left(\ln^{-\varepsilon/2}x\right)\left(1-\frac{\varepsilon\ln b}{2\ln x}+O\left(\ln^{-2}x\right)\right)+o\left(\ln^{-1-\varepsilon/2}x\right) = \\
&\quad = 1+\ln^{-\varepsilon/2}x+\left[\frac{-\varepsilon\ln b}{2}\ln^{-1-\varepsilon/2}x+o\left(\ln^{-1-\varepsilon/2}x\right)\right] \geq
\end{aligned}$$

Since $b\in(0;1)$, we have $\ln b<0$. Therefore, the term in brackets will be positive for sufficiently large x and we have:

$$\geq 1+\ln^{-\varepsilon/2}x$$

□

7.3 The continuous Akra–Bazzi theorem

For the next part, we closely follow Leighton's proof (pp. 6–8). Here we prove the Akra–Bazzi theorem for *continuous* recurrences, i. e. a function $f : \mathbb{R} \to \mathbb{R}$ that fulfils all the conditions stated before, but not just on $\mathbb{N}$, but on $\mathbb{R}$. We therefore look at the following setting, which is essentially the real-valued analogue to the setting described in Section 4:

Consider $f : \mathbb{R} \to \mathbb{R}$ with the following properties:

$$f(x) \geq 0 \qquad \text{for all } x \in [x_0;x_1]$$

$$f(x) = g(x) + \sum_{i=1}^{k} a_i \cdot f(b_i \cdot x + h_i(x)) \qquad \text{for all } x > x_1$$

for a natural number $k \in \mathbb{N} \setminus \{0\}$, a function $g : \mathbb{R} \to \mathbb{R}$, natural numbers $x_0, x_1 \in \mathbb{R}$, real coefficients $a_i \in \mathbb{R}$, $b_i \in \mathbb{R}$, functions $h_i : \mathbb{N} \to \mathbb{R}$, and $p \in \mathbb{R}$ such that:

- $g(x) \geq 0$ for all $x \geq x_0$
- $a_i \geq 0$ for all $i \in [1;k]$ and $a_i > 0$ for at least one $i \in [1;k]$
- $b_i \in (0;1)$ for all $i \in [1;k]$
- for every $i \in [1;k]$, there exists an $\varepsilon_i > 0$ such that $h_i \in O(x/\ln^{1+\varepsilon_i} x)$
- $\sum_{i=1}^{k} a_i \cdot b_i^p = 1$
- $g(u)u^{-p-1}$ is integrable on $[x_0;x]$ for any $x \geq x_0$

We can assume w.l.o.g. that all the h_i fulfil $h_i \in O(x/\ln^{1+\varepsilon} x)$ for the same ε by choosing the minimum of all the ε_i values. It is then clear that there exists a constant H such that, for sufficiently large x, we have $|h_i(x)| \leq Hx\ln^{-1-\varepsilon} x$ for all $i \in [1;k]$.

We also assume that x_0 and x_1 are chosen large enough such that the following inequalities hold:

- $1 \leq x_0 \leq \frac{1}{2} b_i x_1$ for all $i \in [1;k]$
- $|h_i(x)| \leq Hx\ln x^{-1-\varepsilon}$ for all $i \in [1;k]$ and all $x \geq x_1$
- the inequalities (1) to (8) from Lemma 1 for any $b \in \{b_1 \ldots b_k\}$ and all $x \geq x_0$
- there exists some real number C such that $Cx \leq b_i x - Hx\ln^{-1-\varepsilon} x$ for any $i \in [1;k]$ and all $x \geq x_1$

7.3.1 The lower bound.

We will now show how to obtain an asymptotic lower bound on $f(x)$. For this, we further have to assume the existence of positive and finite bounds

$$F := \inf_{x \in [x_0;x_1]} f(x) \qquad \text{and} \qquad G := \sup_{x \in [x_0;x_1]} g(x)$$

and the growth condition $\forall x \geq x_1.\ \forall u \in [Cx;x].\ c_2 g(x) \geq g(u)$ for some $c_2 > 0$. We can then show the following two lemmas:

Lemma 2 *There exists a $c_4 > 0$ such that, for any $i \in [1;k]$ and all $x \geq x_1$:*

$$x^p \int_{b_i x + h_i(x)}^{x} \frac{g(u)}{u^{p+1}}\, du \leq c_4 g(x)$$

Proof Technical and uninteresting; refer to Leighton's Lemma 1 or our formal proof development for details.

Lemma 3 *There exists a $c_5 > 0$ with $c_5 \le \frac{1}{2c_4}$ such that, for any $x \in [x_0;x_1]$:*

$$2c_5 x^p \left(1 + \int_{x_0}^{x} \frac{g(u)}{u^{p+1}}\, du\right) \le f(x)$$

Proof We have $x^p \le \max(x_0^p, x_1^p)$ and $g(u)u^{-p-1} \le G \cdot \max(x_0^{-p-1}, x_1^{-p-1})$; it is easy to use this to derive some bound c such that

$$\frac{2}{F} x^p \left(1 + \int_{x_0}^{x} \frac{g(u)}{u^{p+1}}\, \mathrm{d}u\right) \le c$$

and therefore

$$\frac{2}{c} x^p \left(1 + \int_{x_0}^{x} \frac{g(u)}{u^{p+1}}\, \mathrm{d}u\right) \le F \le f(x)\ .$$

Setting $c_5 := \min(c^{-1}, (2c_4)^{-1})$ yields the desired bound. □

We can then show the following lower bound for f

Lemma 4

$$c_5 x^p \left(1 + \ln^{-\varepsilon/2} x\right) \left(1 + \int_{x_0}^{x} \frac{g(u)}{u^{p+1}}\, du\right) \le f(x) \qquad \textit{for all } x \ge x_0$$

Proof The proof is by induction over x with the base case $x \in [x_0;x_1]$ and the inductive step $x > x_1$ while assuming that the induction hypothesis holds for all $b_i x + h_i(x)$ for any $i \in [1;k]$. This induction scheme is well-founded, since the bounds on the h_i and inequality (8) imply $\lceil b_i x + h_i(x) \rceil < \lceil x \rceil$ for any $x \ge x_1$, so the measure $\mu : \mathbb{R} \to \mathbb{N}, x \mapsto \lceil x \rceil$ decreases.

Base case. We have $x \in [x_0;x_1]$ and therefore:

$$c_5 x^p \underbrace{\left(1 + \ln^{-\varepsilon/2} x\right)}_{\substack{\le 2 \\ \text{by Lemma 1.(3)}}} \left(1 + \int_{x_0}^{x} \frac{g(u)}{u^{p+1}}\, \mathrm{d}u\right) \le 2c_5 x^p \left(1 + \int_{x_0}^{x} \frac{g(u)}{u^{p+1}}\, \mathrm{d}u\right) \overset{\text{Lemma 3}}{\le} f(x)$$

Induction step We have $x > x_1$ and we assume the following induction hypothesis for any $i \in [1;k]$:

$$c_5 (b_i x + h_i(x))^p \left(1 + \ln^{-\varepsilon/2}(b_i x + h_i(x))\right) \left(1 + \int_{x_0}^{b_i x + h_i(x)} \frac{g(u)}{u^{p+1}}\, \mathrm{d}u\right) \le$$
$$\le f(b_i x + h_i(x)) \qquad \text{(IH)}$$

We can then show:

$$\begin{aligned}
&c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)\overset{\text{Lemma 3}}{\leq}\\
&\leq c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)+g(x)-2c_5c_4g(x)\overset{\text{Lemma 1.(3)}}{\leq}\\
&\leq c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)+g(x)-c_5c_4\left(1+\ln^{-\varepsilon/2}x\right)g(x)=\\
&=g(x)+c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u-\frac{c_4}{x^p}g(x)\right)\overset{\text{Lemma 2}}{\leq}\\
&=g(x)+c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u-\int_{b_ix+h_i(x)}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)=\\
&=g(x)+c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{b_ix+h_i(x)}\frac{g(u)}{u^{p+1}}\mathrm{d}u\right)=\\
&=g(x)+\left(\sum_{i=1}^{k}a_ib_i^p\right)c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{b_ix+h_i(x)}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)=\\
&=g(x)+\sum_{i=1}^{k}a_ic_5b_i^px^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{b_ix+h_i(x)}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)
\end{aligned}$$

Let us now focus on the $b_i^px^p\left(1+\ln^{-\varepsilon/2}x\right)$ term. Let $s:=-1$ if $p\geq 0$ and $s:=1$ otherwise.[5]By applying first Lemma 1.(1) and then $|h_i(x)|\leq Hx\ln x^{-1-\varepsilon}$, we have:

$$\begin{aligned}
b_i^px^p\left(1+\ln^{-\varepsilon/2}x\right)&\overset{(1)}{\leq} b_i^px^p\left(1+\frac{sH}{b_i\ln^{1+\varepsilon}x}\right)^p\left(1+\ln^{-\varepsilon/2}\left(b_ix+\frac{Hx}{\ln^{1+\varepsilon}x}\right)\right)=\\
&=\left(b_ix+\frac{sHx}{\ln^{1+\varepsilon}x}\right)^p\left(1+\ln^{-\varepsilon/2}\left(b_ix+\frac{Hx}{\ln^{1+\varepsilon}x}\right)\right)\leq\\
&\leq (b_ix+h_i(x))^p\left(1+\ln^{-\varepsilon/2}(b_ix+h(x))\right)
\end{aligned}$$

Plugging this result into the inequality chain we interrupted before, we have:

$$\begin{aligned}
&g(x)+\sum_{i=1}^{k}a_ic_5b_i^px^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{b_ix+h_i(x)}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)\leq\\
&\leq g(x)+\sum_{i=1}^{k}a_ic_5\,(b_ix+h_i(x))^p\left[1+\ln^{-\varepsilon/2}(b_ix+h(x))\right]\left(1+\int_{x_0}^{b_ix+h_i(x)}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)\overset{\text{(IH)}}{\leq}\\
&\leq g(x)+\sum_{i=1}^{k}a_if(b_ix+h_i(x))=f(x)
\end{aligned}$$

This concludes the induction. We have thus shown that for all $x\geq x_0$:

$$c_5x^p\left(1+\int_{x_0}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)\leq c_5x^p\left(1+\ln^{-\varepsilon/2}x\right)\left(1+\int_{x_0}^{x}\frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)\leq f(x)$$

□

[5] Note the implicit case distinction for $p\geq 0$ and $p<0$: for $p\geq 0$, we need to show $b_ix-Hx\ln^{-1-\varepsilon}x\leq b_ix+h_i(x)$, whereas for $p<0$ we need to show $b_ix+Hx\ln^{-1-\varepsilon}x\geq b_ix+h_i(x)$ since the negative exponent flips the inequality. This case distinction is not mentioned by Leighton.

7.3.2 The upper bound

The proof of the upper bound is analogous with the following exceptions:

- The assumptions we need are $\sup_{x\in[x_0;x_1]} f(x) < \infty$ and $\forall x \geq x_1.\ \forall u \in [Cx;x].\ c_1 g(x) \leq g(u)$ for some $c_1 > 0$.
- The inequality we need to show by induction is

$$f(x) \leq c_6 x^p \left(1 - \ln^{-\varepsilon/2} x\right) \left(1 + \int_{x_0}^{x} \frac{g(u)}{u^{p+1}}\,\mathrm{d}u\right)$$

7.4 Lifting to the discrete case

The remaining work in the proof of the Akra–Bazzi theorem is now to lift this result for continuous recurrences to discrete recurrences. We will again illustrate this for the lower bound. We consider the setting given in Theorem 1.

First of all, we find values $G > 0$ and $\hat{x}_0$, $\hat{x}_1$ such that:

- $x_1 \leq \hat{x}_0 \leq \hat{x}_1$
- $f(x) > 0$ for all $x \geq \hat{x}_0$
- $g(x) \geq G\bar{g}(x)$ for all $x \geq \hat{x}_0$ (possible because $g \in \Omega(\bar{g})$)
- $\bar{g}(x) \geq 0$ for all $x \geq \hat{x}_0$
- $b_i x + h_i(x) \geq \hat{x}_0$ for all $x \geq \hat{x}_1$ and $i \in [1;k]$

Since g has the asymptotic lower bound $\bar{g}$ and we want to find an asymptotic lower bound on f, it is natural that the function obtained by copying the definition of f, but with the costs $\bar{g}$ instead of g, should be an asymptotic lower bound on f. We therefore define the function $\hat{f} : \mathbb{N} \to \mathbb{R}$ as follows:

$$\hat{f}(x) = \begin{cases} \max(0, f(x)/G) & \text{if } x < \hat{x}_1 \\ \bar{g}(x) + \sum_{i=1}^{k} a_i \hat{f}(b_i x + h_i(x)) & \text{otherwise} \end{cases}$$

It is clear that $\hat{f}$ is non-negative everywhere, positive for large enough x, and $G\hat{f}(x) \leq f(x)$ for all $x \geq x_0$. Therefore, any asymptotic lower bound on $\hat{f}$ will also be a lower bound on f.

Next, we find some $\bar{x}_0 > \hat{x}_1$ such that

- the asymptotic inequalities (1) to (8) hold for all $x \geq x_0$, $c = H$, $b = b_i$ for $i \in [1;k]$
- $|h_i(x)| \leq Hx\ln^{-1-\varepsilon} x$ for all $x \geq \bar{x}_0$ and $i \in [1;k]$
- $Cx \leq b_i x - Hx\ln^{-1-\varepsilon} x$ for all $x \geq \bar{x}_0$ and $i \in [1;k]$
- $\bar{g}$ is bounded from above on all intervals that lie above $\bar{x}_0$
- $\bar{g}(u) \leq c_2 \bar{g}(x)$ for all $x \geq \bar{x}_0$ and $u \in [Cx;x]$
- $\hat{f}(\lfloor x \rfloor) > 0$ and $\bar{g}(x) \geq 0$ for all $x \geq \bar{x}_0$
- $\bar{g}(u)u^{-p-1}$ is integrable on all intervals that lie above $\bar{x}_0$

and we find a $\bar{x}_1$ such that $\bar{x}_1 \geq \frac{2}{b_i}\bar{x}_0$ for all $i \in [1;k]$.

We can then extend $\hat{f}$ to a function $\bar{f} : \mathbb{R} \to \mathbb{R}$ using the following definition:

$$\bar{f}(x) = \begin{cases} \hat{f}(\lfloor x \rfloor) & \text{if } x \leq \bar{x}_1 \\ g(x) + \sum_{i=1}^{k} a_i \bar{f}(b_i x + h_i(x)) & \text{otherwise} \end{cases}$$

Clearly, $\bar{f}(x) = \hat{f}(x)$ for all natural numbers x.

We apply the continuous lower bound theorem derived in the previous section to $\bar{f}(x)$ with the 'base cases' between $\bar{x}_0$ and $\bar{x}_1$. This gives us the lower bound

$$c_5 x^p \left(1 + \int_{\bar{x}_0}^{x} \frac{\bar{g}(u)}{u^{p+1}}\,\mathrm{d}u\right) \leq \bar{f}(n)$$

For all natural numbers $n \geq \bar{x}_0$, we then have:

$$Gc_5 n^p \left(1 + \int_{\bar{x}_0}^{n} \frac{\bar{g}(u)}{u^{p+1}}\,\mathrm{d}u\right) \leq G\bar{f}(n) = G\hat{f}(n) \leq f(n)$$

and therefore

$$f \in \Omega\left(n^p \left(1 + \int_{\bar{x}_0}^{n} \frac{\bar{g}(u)}{u^{p+1}}\,\mathrm{d}u\right)\right) .$$

Since we could have chosen $\bar{x}_0$ larger as well if we had wanted to, we have essentially shown that

$$f \in \Omega\left(n^p \left(1 + \int_{t}^{n} \frac{\bar{g}(u)}{u^{p+1}}\,\mathrm{d}u\right)\right)$$

holds for any sufficiently large t.

The formal proof of all of this consists mostly of finding large enough values for $\bar{x}_0$ and $\bar{x}_1$ and proving that all required properties hold. For instance, the condition that $\inf_{x\in[\bar{x}_0;\bar{x}_1]} \bar{f}(x) > 0$ is fulfilled because, by definition, $\bar{f}(x) = \hat{f}(\lfloor x \rfloor)$ only takes a finite number of values on the interval $[\bar{x}_0;\bar{x}_1]$, all of which are positive.

This entire process of linking the assumptions of the discrete settings to the assumptions of the continuous setting is very technical and intricate – it takes up almost a quarter of the entire proof of the Akra–Bazzi theorem – but mathematically uninteresting, which is why we left out a lot of detail in this paper proof.

7.5 The Master Theorem

From the proof of the Akra–Bazzi theorem in the locale context *akra_bazzi_function*, we can now show the Master Theorem, also inside this context. The proofs are straightforward applications of the Akra–Bazzi theorem. The user can then interpret the *akra_bazzi_function* locale for her function f and use the case of the Master Theorem appropriate for her function.

In the Isabelle formalisation, the Master Theorem is split into five cases (cf. Table 2), with the first case having a weak form (O) and a strong form (Θ).

8 Automation

The formalisation also contains three proof methods that add a certain degree of automation to the usage of the Master Theorem. We will describe them in the following sections.

8.1 Akra–Bazzi terms

As mentioned previously in Section 7.1, the condition that recursive calls must be Akra–Bazzi terms can be discharged by introduction rules that reduce the condition to simple statements on constants. We provide a theorem collection called `akra_bazzi_term_intro` to which the user can add custom introduction rules for Akra–Bazzi terms. The Akra–Bazzi proof methods then automatically use these rules to discharge conditions of this kind.

Case name	Assumptions			Conclusion
Case 1 (O)	$g \in O(x^q)$	$q < p$		$f \in O(x^p)$
Case 1	$g \in O(x^q)$	$q < p$	$f(x) > 0$ (for suff. large x)	$f \in \Theta(x^p)$
Case 2.1	$g \in \Theta(x^p \ln^q x)$	$q < -1$		$f \in \Theta(x^p)$
Case 2.2	$g \in \Theta(x^p / \ln x)$			$f \in \Theta(x^p \ln \ln x)$
Case 2.3	$g \in \Theta(x^p \ln^{q-1} x)$	$q > 0$		$f \in \Theta(x^p \ln^q x)$
Case 3	$g \in \Theta(x^q)$	$q > p$		$f \in \Theta(x^q)$

Table 2 The five cases of the Master Theorem as formalised in Isabelle/HOL

8.2 akra_bazzi_termination

It is possible to define recursive functions with complex recursion schemes in Isabelle/HOL. [13]. For every function definition, the user must show that the function is indeed *well-defined*: the definition must be complete, different equations must not overlap with one another, and the function must terminate, i. e. there must not be any infinite chains of recursive calls. The first two conditions can virtually always be proven automatically; the last condition, termination, is usually the most difficult to prove. Isabelle/HOL can prove it automatically in many cases, but Akra–Bazzi style recursion schemes are usually not among them: for example, attempting to define the cost function for Merge Sort recursively by its recurrence relation $f(n) = f(\lfloor \frac{n}{2} \rfloor) + f(\lceil \frac{n}{2} \rceil) + n$ for $n \geq 2$ will fail since Isabelle's termination prover is unable to prove that n becomes smaller in every recursion step.

To aid the user in proving termination for such recursion schemes, we developed the proof method `akra_bazzi_termination`, which uses the `akra_bazzi_term_intro` rules mentioned before to reduce the proof obligation of termination to simpler proof obligations that contain only constants, which can then be solved automatically using Isabelle's simplifier. A typical function definition of an Akra–Bazzi function then looks like this:

function merge_sort_cost :: nat $\Rightarrow$ real **where**
  merge_sort_cost$(0) = 0$
| merge_sort_cost$(1) = 1$
| $n \geq 2 \Longrightarrow$ merge_sort_cost$(n) =$
    merge_sort_cost$(\lfloor n/2 \rfloor) +$ merge_sort_cost$(\lceil n/2 \rceil) + n$

The termination of this function can be proven using the `akra_bazzi_termination` method. In this case, it produces ten proof obligations of the form $0 < \frac{1}{2}$, $\frac{1}{2} < 1$, $0 \leq \frac{1}{2} \cdot 2$, etc. These can be solved automatically with the simplifier.

It should be noted that `akra_bazzi_termination` works in more complicated situations as well, e. g. when the function has several arguments (curried or tupled), which may even change during the recursive call.

In order to achieve this, `akra_bazzi_termination` performs some analysis of the function's type to find a parameter that is a natural number and for which the recursive calls are Akra–Bazzi terms. It then starts a termination proof using this parameter as a termination measure and applies the corresponding introduction rules for Akra–Bazzi terms, which leaves only the simple proof obligations we saw earlier.

Note also that `akra_bazzi_termination` is completely independent from the Akra–Bazzi theorem itself; its only connection to Akra–Bazzi is the fact that it helps automate termination proofs of Akra–Bazzi-style recurrences.

8.3 master_theorem

The main proof method is `master_theorem`, which can be invoked on goals of the form $f \in O(\_)$ or $f \in \Theta(\_)$. It takes as an argument the applicable case of the Master Theorem (e. g. 1 or 2.2) and applies it to the goal. The recursive equation of f and the values x_0 and x_1 are optional parameters which the method attempts to guess if absent. The appropriate values for k, as, bs, ts, and g are always inferred automatically from the recursive equation, provided the required `akra_bazzi_term_intro` rules exist.

To provide some more detail as to where all these values come from:

- The function f can obviously be determined from the goal itself.
- If not provided explicitly, the method will try to obtain the recursive equation for f from its definition.
- p, the characteristic number of the recurrence, never needs to be specified explicitly. If it appears in the goal, the proof method infers it from the goal; if it does not appear in the goal, its value is not required explicitly – if there is e. g. a proof obligation that some number q is larger than p, the user is presented with the equivalent proof obligation that $\sum_{i=1}^{k} a_i \cdot b_i^q < 1$.
- x_0 is set to 0 if not given explicitly, which is always a correct choice except in the unusual case that f is negative for some inputs.
- x_1 is determined from the precondition of the recursive equation (e. g. 2 for the precondition $n > 1$).
- k, as and ts are determined by partitioning the right-hand side of the recursive equation into summands and bringing each summand into the form $a_i \cdot f(ts_i\ n)$ if possible.
- g is the sum of all summands that cannot be brought into this form.
- for each $t \in ts$, the corresponding $b \in bs$ is determined by finding a rule from the theorem collection `akra_bazzi_term_intro` that matches t and extracting b from the unifier.

We will now illustrate the practical application of the proof method with two examples.

Example: Merge Sort
As an example, we consider the `merge_sort_cost` function from Section 8.2. Case 2.3 of the Master Theorem tells us that the growth of this function is $\Theta(n \ln n)$. In Isabelle, we write:

lemma merge_sort_cost $\in \Theta(\lambda n.\ n \cdot \ln\ n)$
apply (master_theorem 2.3)

This leaves us with the following proof obligations:

1. $\forall n.\ 0 \le n \implies n < 2 \implies 0 \le \text{merge\_sort\_cost}\ n$
2. $\forall n.\ 2 \le n \implies \text{merge\_sort\_cost}(n) =$
   $n\ +\ \text{merge\_sort\_cost}(\lfloor n/2 \rfloor)\ +\ \text{merge\_sort\_cost}(\lceil n/2 \rceil)$
3. $\forall n.\ 2 \le n \implies 0 \le n$
4. $(\lambda n.\ n) \in \Theta(\lambda n.\ n)$

The first goal, non-negativity of merge_sort_cost, can be proven easily by case distinction using the function definition. All the remaining goals can be discharged automatically by the simplifier.

Example: Boncelet coding
Another interesting example is given by Drmota and Szpankowski [8]: the average phrase length $d(n)$ in Boncelet coding [6]. This is not related to Divide & Conquer algorithms at all, but d does fulfil the Akra–Bazzi-type recurrence

$$d(n) = 1 + p \cdot d(\lfloor p \cdot n + \delta \rfloor) + q \cdot d(\lfloor q \cdot n - \delta \rfloor)$$

where $p \in (0;1)$ is a probability, $q = 1 - p$, and $\delta > 0$, $\delta < 1$, $2p + \delta < 2$. Since our Master Theorem applies to this recurrence, we can use the `master_theorem` tactic:

lemma boncelet_phrase_length:
fixes $p\ \delta$ :: real
fixes d :: nat $\Rightarrow$ real
defines $q \equiv 1 - p$
assumes $p \in (0;1)$ **and** $\delta \in (0;1)$ **and** $2 \cdot p + \delta < 2$ **and** $\forall n.\ d(n) \ge 0$
assumes $\forall n{\ge}2.\ d(n) = 1 + p \cdot d(\lfloor p \cdot n + \delta \rfloor) + q \cdot d(\lfloor q \cdot n + \delta \rfloor)$
shows $d \in \Theta(\lambda x.\ \ln x)$
using assms **by** (master_theorem recursion: d_rec, simp_all)

This example is particularly interesting because here we do not have a concrete function that we defined ourselves, but prove a theorem for an entire class of functions satisfying a certain recurrence. Since d is not a function-package function, we need to give `master_theorem` the recurrence rule to use, but then, the lemma can be proven automatically again. Note also that we did not specify which case to use here; Isabelle implicitly backtracks over all possible cases and succeeds in case 2.3. It should, however, be mentioned that the Θ-estimate is not very useful in this case: as Drmota and Szpankowski elaborate, an analysis of the redundancy of the Boncelet scheme requires more precise asymptotics, like the ones they provide.

Further examples of complexity proofs for recurrences derived from the computational costs of Divide & Conquer algorithms such as Karatsuba's multiplication algorithm for natural numbers, Strassen's algorithm for matrix multiplication, or the deterministic Median-of-Medians selection algorithm can be found in the examples file of our entry in the *Archive of Formal Proofs* [9].

All examples can be proven mostly automatically with the `master_theorem` method and Isabelle's simplifier. (in no small part thanks to the decision procedure we presented in section 3.2.2) There are only three kinds of subgoals that are sometimes left behind:

– Positivity/non-negativity for sufficiently large inputs; this can typically be shown by straightforward induction and simplification.

- Showing that p satisfies $\sum_{i=1}^{k} as_i \cdot bs_i^p = 1$; proving this is usually automatic, but can be non-trivial in some cases. One such example is that from our introduction: $f(n) = f(\lfloor n/4 \rfloor) + f(\lfloor n/2 \rfloor) + 1$ with $p = \log_2 \varphi$, where φ is the golden ratio.
- Inequalities like $q < p$ for complicated constants p, q (e. g. involving logarithms and square roots); these can usually be proven automatically by approximation [12].

8.4 akra_bazzi_approximation

In some of the cases of the Master Theorem, the goal contains the parameter p. As shown before, this p always exists and is unique. It can, however, not always be expressed in a closed form. One example for such a situation is the function $f(x) = f(\lfloor x/3 \rfloor) + f(\lfloor 3x/4 \rfloor)$, for which p seems to have no closed form, but can be approximated to 1.152.
Our Master Theorem in Isabelle/HOL yields

$$\textbf{lemma}\ \ f \in \Theta\left(\lambda n.\ n^{\text{akra\_bazzi\_exponent}([1,1],[1/3,3/4])}\right)$$

where *akra_bazzi_exponent* is a function that, given lists *as* and *bs* that fulfil the usual conditions, returns the unique p such that $\sum_{i=1}^{k} as_i \cdot bs_i^p = 1$.

Of course, one would now like to obtain and verify at least an approximate value for this exponent. An approximate value for p can be found e. g. using an external computer algebra system or by applying Newton's method as described in Sec. 8.4. Once such an approximation has been found, it can then be verified with our proof method `akra_bazzi_approximate`:

$$\begin{array}{l}\textbf{lemma}\ \text{akra\_bazzi\_exponent}([1,1],[1/3,3/4]) \in [1.1519623; 1.1519624] \\ \quad \textbf{by}\ (\text{akra\_bazzi\_approximate}\ 29)\end{array}$$

This proof method internally uses the `approximation` tactic [12], which uses Taylor series expansions, interval arithmetic, and reflection (i. e. Isabelle's code generator [11]) to certify bounds on the values of certain transcendental functions. The number 29 in the invocation here indicates the precision (measured in binary digits) of the computation. If it is too low, the proof attempt might fail even though the statement is true; if it is too high, the proof will take more time. The user of the proof method should therefore find a precision value that is as low as possible while still being high enough for the proof to succeed.

Like `akra_bazzi_termination`, the method `akra_bazzi_approximation` is completely orthogonal to the Akra–Bazzi theorem; neither depends on the other. In particular, no part of the proof of the Akra–Bazzi theorem relies on numerical approximation.

Approximating p. As a side note: p can be approximated quite easily. Recall that p was defined as the unique real number such that $t(p) = 1$ where

$$t : \mathbb{R} \to \mathbb{R}_{>0},\ x \mapsto \sum_{i=1}^{k} a_i \cdot b_i^x\ .$$

Note that $t(x) - 1$ is convex and $t'(x)$ has no zeros. This means that Newton's method can be used to approximate p very efficiently: Due to the convexity of $t(x) - 1$, a Newton step at any lower bound (i. e. intersecting the tangent of t at the lower bound with the x axis) will yield a better lower bound, and intersecting the secant of a lower bound and an upper bound on p with the x axis will yield a better upper bound. This can be used to obtain good approximation intervals for p very quickly.

As initial lower and upper bounds, one can e. g. use the estimates

$$-\frac{\ln a_k}{\ln b_k} \leq p \leq -\frac{\ln\left(n \cdot \max_{i\in[1;n]} a_i\right)}{\ln\left(\max_{i\in[1;n]} b_i\right)} \qquad \text{where } k \in [1;n] \text{ arbitrary }.$$

We have not verified this approximation algorithm in Isabelle, since the approximation can easily be done externally in an unverified way and then certified using e. g. the `akra_bazzi_approximation` method.

9 Comparison with similar theorems

9.1 Leighton's Akra–Bazzi theorem

Our version of the Akra–Bazzi theorem differs from Leighton's [14] in a few minor respects:

- We require only one a_i to be positive and the rest to be non-negative. In Leighton's version, all of them must be positive.
- Every property that we require to hold between x_0 and x_1, Leighton requires to hold between 1 and x_0. (which corresponds to our x_1)
- We have two separate functions g and $\bar{g}$, where the latter is an integrable asymptotic bound for the former. Leighton essentially requires g and $\bar{g}$ to be the same.
- We have not only a Θ version of the Akra–Bazzi theorem, but also O and Ω versions, which may be useful when the behaviour of g is not fully known.

9.2 Master Theorem

Due to its being derived from the Akra–Bazzi theorem, our version of the Master Theorem is much more general than the versions of the Master Theorem that are typically presented in the literature (e. g. *Introduction to Algorithms* [7]). Our version of the Master Theorem imposes far fewer restrictions on the shape of the recursive call, allowing multiple terms with floors, ceilings, and other deviations.

Apart from the more general recursion scheme that our version of the Master Theorem allows, there are two more differences to the version of the Master Theorem given by Cormen *et al.* [7]:

The second case of our version of the Master Theorem is more general, as it allows arbitrary real numbers q whereas the version by Cormen *et al.* require $q \geq 0$.[6]

The third case is more restrictive than that by Cormen *et al.*: we demand $g \in \Theta(x^q)$, whereas Cormen *et al.* only demand $g \in \Omega(x^q)$ and the existence of some $c < 1$ such that for all sufficiently large x, the regularity condition $a \cdot g(x/b) \leq c \cdot g(x)$ holds. For a more complex recursion scheme, as allowed by our Master Theorem, this regularity condition would be so complicated that it would be very inconvenient:

$$\sum_{i=1}^{k} a_i \cdot g(b_i \cdot x + h_i(x)) \leq c \cdot g(x)$$

[6] The Master Theorem presented in the book actually demands $q = 0$, but Exercise 4.2-2 is a generalisation to $q \geq 0$.

Given such a regularity condition, the proof that f is then in $\Theta(g)$ is relatively simple and does not require the Akra–Bazzi theorem at all, which is why we did not include this in our proof.

It should also be noted that while most informal proofs of the Master Theorem mention that rounding before the recursive call does not change the result, this is seldom proven concisely. Cormen *et al.* give a partial proof that their Master Theorem also holds for $f(x) = a \cdot f(\lfloor x/b \rfloor) + g(x)$ and $f(x) = a \cdot f(\lceil x/b \rceil) + g(x)$, but they do not address the case $f(x) = a_1 \cdot f(\lfloor x/b \rfloor) + a_2 \cdot f(\lceil x/b \rceil) + g(x)$. This is unfortunate, because even simple Divide & Conquer algorithms such as Merge Sort have cost recurrences of this kind. Ad-hoc arguments using monotonicity can be made for concrete examples such as Merge Sort, but it is convenient to handle all of these deviations with a unified theorem like the Akra–Bazzi theorem.

10 Conclusion

We formally verified a very general version of the Akra–Bazzi method [1] with the theorem prover Isabelle/HOL. This enables users of Isabelle to obtain verified asymptotic bounds for many typical 'Divide & Conquer' recurrence relations. In the process of our formalisation, we found a missing case in Leighton's original proof [14]. We also clarified some important parts of the proof that Leighton does not address (such as the asymptotic inequalities and lifting the estimate for continuous recurrences to discrete ones) and slightly generalised the theorem.

Based upon our formal proof of the Akra–Bazzi method, we also proved a generalisation of the well-known Master Theorem whose generality is somewhere between the 'classic' Master Theorem and the Akra–Bazzi method, but with no additional cost compared to the 'classic' Master Theorem. In particular, we thus accounted rigorously for any rounding in the recursive calls, which is often neglected in informal proofs of the Master Theorem.

Additionally, we developed some automated proof methods that facilitate defining such recursive functions and working with our Master Theorem. We evaluated this machinery on some standard textbook examples of Divide & Conquer algorithms (Merge Sort, Karatsuba multiplication, deterministic Median-of-Medians selection) and found that the complexity proofs were almost completely automatic.

Acknowledgements Tobias Nipkow and Johannes Hölzl commented on an early draft of this work. Louay Bazzi made a very helpful suggestion that allowed us to fix the missing case in Leighton's proof. We also thank the anonymous reviewers for their helpful suggestions and insightful questions.

References

1. Akra, M., Bazzi, L.: On the solution of linear recurrence equations. Computational Optimization and Applications **10**(2), 195–210 (1998). DOI 10.1023/A:1018373005182. URL `http://dx.doi.org/10.1023/A%3A1018373005182`
2. Avigad, J., Donnelly, K.: Formalizing *O* notation in Isabelle/HOL. In: D. Basin, M. Rusinowitch (eds.) Automated Reasoning, *Lecture Notes in Computer Science*, vol. 3097, pp. 357–371. Springer Berlin Heidelberg (2004). DOI 10.1007/978-3-540-25984-8_27. URL `http://dx.doi.org/10.1007/978-3-540-25984-8_27`
3. Avigad, J., Hölzl, J., Serafin, L.: A formally verified proof of the central limit theorem. CoRR **abs/1405.7012** (2014). Presented at the Isabelle Workshop 2014
4. Ballarin, C.: Locales: A module system for mathematical theories. Journal of Automated Reasoning **52**(2), 123–153 (2014). DOI 10.1007/s10817-013-9284-7. URL `http://dx.doi.org/10.1007/s10817-013-9284-7`
5. Bazzi, L., Mitter, S.K.: The solution of linear probabilistic recurrence relations. Algorithmica **36**(1), 41–57 (2003). DOI 10.1007/s00453-002-1003-4. URL `http://dx.doi.org/10.1007/s00453-002-1003-4`
6. Boncelet Jr, C.G.: Block arithmetic coding for source compression. Information Theory, IEEE Transactions on **39**(5), 1546–1554 (1993). DOI 10.1109/18.259639. URL `http://dx.doi.org/10.1109/18.259639`
7. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn., 4th printing. The MIT Press (2009)
8. Drmota, M., Szpankowski, W.: A Master theorem for discrete divide and conquer recurrences. J. ACM **60**(3), 16:1–16:49 (2013). DOI 10.1145/2487241.2487242. URL `http://doi.acm.org/10.1145/2487241.2487242`
9. Eberl, M.: The Akra–Bazzi theorem and the Master theorem. Archive of Formal Proofs (2015). `http://www.isa-afp.org/entries/Akra_Bazzi.shtml`, Formal proof development
10. Eberl, M.: Landau symbols. Archive of Formal Proofs (2015). `http://www.isa-afp.org/entries/Landau_Symbols.shtml`, Formal proof development
11. Haftmann, F., Nipkow, T.: Code generation via higher-order rewrite systems. In: M. Blume, N. Kobayashi, G. Vidal (eds.) Functional and Logic Programming, *Lecture Notes in Computer Science*, vol. 6009, pp. 103–117. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-12251-4_9. URL `http://dx.doi.org/10.1007/978-3-642-12251-4_9`
12. Hölzl, J.: Proving inequalities over reals with computation in Isabelle/HOL. In: G.D. Reis, L. Théry (eds.) Proceedings of the ACM SIGSAM 2009 International Workshop on Programming Languages for Mechanized Mathematics Systems (PLMMS'09), pp. 38–45. Munich (2009)
13. Krauss, A.: Automating recursive definitions and termination proofs in Higher-Order Logic. Ph.D. thesis, Technische Universität München, Institut für Informatik (2009). URL `http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20090722-681651-1-1`
14. Leighton, T.: Notes on better Master theorems for divide-and-conquer recurrences (1996). URL `http://courses.csail.mit.edu/6.046/spring04/handouts/akrabazzi.pdf`

A The Class of Admissible Integrals

We will now provide a proof of our claim from Section 3.3 that any admissible integral must coincide with the Riemann integral for all piecewise-continuous functions. First of all, we shall state again what admissibility means formally: Consider an integral notion $\mathcal{I}$. Formally, $\mathcal{I}$ consists of

- A functional $\mathcal{I} : \mathbb{R}^{\mathbb{R}} \times \mathbb{R}^2 \to \mathbb{R}$ that maps a function f and interval bounds a and b to the $\mathcal{I}$-integral of f from a to b, denoted by
$$\underset{\mathcal{I}}{\int_a^b} f(x)\,\mathrm{d}x\,.$$
- A set $\mathcal{I} \subseteq \mathbb{R}^{\mathbb{R}} \times \mathbb{R}^2$, the set of $\mathcal{I}$-integrable functions. We say that f is $\mathcal{I}$-integrable on $[a;b]$ if $(f,a,b) \in \mathcal{I}$.

For the remainder of this section, we will always indicate for every use of the integral sign which notion of integration is meant by writing the name of the integral underneath the integral as we did above.

Now assume that $\mathcal{I}$ fulfils the following four properties (which are the same that were stated in Section 3.3, but more formally):

$$\forall a,b,c \in \mathbb{R}. \qquad a \le b \,\wedge\, c \ge 0 \longrightarrow ((\lambda x.\ c),a,b) \in \mathcal{I} \,\wedge\, \underset{\mathcal{I}}{\int_a^b} c\,\mathrm{d}x = c \cdot (a-b) \tag{9}$$

$$\forall a,b,a',b' \in \mathbb{R}.\ \forall f \in \mathbb{R}^{\mathbb{R}}. \qquad a \le a' \le b' \le b \,\wedge\, (f,a,b) \in \mathcal{I} \longrightarrow (f,a',b') \in \mathcal{I} \tag{10}$$

$$\forall a,b \in \mathbb{R}.\ \forall f,g \in \mathbb{R}^{\mathbb{R}}. \qquad (f,a,b) \in \mathcal{I} \,\wedge\, (g,a,b) \in \mathcal{I} \,\wedge\, (\forall x \in [a;b].\ f(x) \le g(x)) \longrightarrow \underset{\mathcal{I}}{\int_a^b} f(x)\,\mathrm{d}x \le \underset{\mathcal{I}}{\int_a^b} g(x)\,\mathrm{d}x \tag{11}$$

$$\forall a,b,c \in \mathbb{R}.\ \forall f \in \mathbb{R}^{\mathbb{R}}. \qquad a \le b \le c \,\wedge\, (f,a,c) \in \mathcal{I} \longrightarrow \underset{\mathcal{I}}{\int_a^c} f(x)\,\mathrm{d}x = \underset{\mathcal{I}}{\int_a^b} f(x)\,\mathrm{d}x + \underset{\mathcal{I}}{\int_b^c} f(x)\,\mathrm{d}x \tag{12}$$

To show that the $\mathcal{I}$-integral of a non-negative function f over $[a;b]$ coincides with the Riemann integral if f is both $\mathcal{I}$-integrable and piecewise-continuous on $[a;b]$, we recall that the Riemann integral is equivalent to the Darboux integral. The Darboux integral is defined as the supremum L_f of the lower Darboux sums $L_{f,P}$ and the infimum U_f of the upper Darboux sums $U_{f,P}$ over all subdivisions $P = (c_0 \ldots c_n)$ of the interval $[a;b]$ whenever L_f and U_f are equal. Formally:

$$\underset{\text{Darboux}}{\int_a^b} f(x)\,\mathrm{d}x := L_f = U_f \qquad (\text{if } L_f = U_f)$$

where

$$L_f = \sup_P L_{f,P} = \sup_P \sum_{i=0}^{n-1} (c_{i+1} - c_i) \inf_{x \in [c_i;c_{i+1}]} f(x) \quad \text{and} \quad U_f = \inf_P U_{f,P} = \inf_P \sum_{i=0}^{n-1} (c_{i+1} - c_i) \sup_{x \in [c_i;c_{i+1}]} f(x)$$

Suppose f is non-negative, continuous, and $\mathcal{I}$-integrable on $[a;b]$. For any subdivision $P = (c_0 \ldots c_n)$ of the interval $[a;b]$, we can use (10) and (12) to split up the $\mathcal{I}$-integral over $[a;b]$:

$$\underset{\mathcal{I}}{\int_a^b} f(x)\,\mathrm{d}x = \sum_{i=0}^{n-1} \underset{\mathcal{I}}{\int_{c_i}^{c_{i+1}}} f(x)\,\mathrm{d}x$$

Using this together with the monotonicity and constant-interval property of $\mathcal{I}$, we have:

$$L_{f,P} \overset{\text{def}}{=} \sum_{i=0}^{n-1} (c_{i+1} - c_i) \inf_{x \in [c_i;c_{i+1}]} f(x) \overset{(9)}{=} \sum_{i=0}^{n-1} \underset{\mathcal{I}}{\int_{c_i}^{c_{i+1}}} \inf_{x \in [c_i;c_{i+1}]} f(x)\,\mathrm{d}x \overset{(11)}{\le} \underbrace{\sum_{i=0}^{n-1} \underset{\mathcal{I}}{\int_{c_i}^{c_{i+1}}} f(x)\,\mathrm{d}x}_{=\underset{\mathcal{I}}{\int_a^b} f(x)\,\mathrm{d}x} \overset{(11)}{\le}$$
$$\le \sum_{i=0}^{n-1} \underset{\mathcal{I}}{\int_{c_i}^{c_{i+1}}} \sup_{x \in [c_i;c_{i+1}]} f(x) \overset{(9)}{=} \sum_{i=0}^{n-1} (c_{i+1} - c_i) \sup_{x \in [c_i;c_{i+1}]} f(x) \overset{\text{def}}{=} U_{f,P}$$

Therefore, the $\mathcal{I}$-integral lies between all lower and upper Darboux sums. Since f is continuous, f is also Darboux-integrable, and therefore the supremum of the lower Darboux sums and the infimum of the upper Darboux sums are the same. Since the $\mathcal{I}$-integral lies inbetween, we have:

$$\underset{\text{Darboux}}{\int_a^b} f(x)\,\mathrm{d}x \overset{\text{def}}{=} L_f = U_f = \sum_{i=0}^{n-1} \underset{\mathcal{I}}{\int_{c_i}^{c_{i+1}}} f(x)\,\mathrm{d}x = \underset{\mathcal{I}}{\int_a^b} f(x)\,\mathrm{d}x\,.$$

We have therefore shown that $\mathcal{I}$ coincides with the Riemann integral on all continuous non-negative functions. Since we can split the $\mathcal{I}$-integral of a piecewise-continuous function into a sum of $\mathcal{I}$-integrals of continuous functions, this extends to all piecewise-continuous non-negative functions.

However, this result does not extend to more general notions of integrals and integrability: For example, let us consider the following integral $\mathcal{I}$: A function is $\mathcal{I}$-integrable if it is a constant function or if it is $[\mathbb{Q}] = (\lambda x.\ \text{if } x \in \mathbb{Q} \text{ then } 1 \text{ else } 0)$, the indicator function of the rational numbers. The value of the integral is defined as

$$\underset{\mathcal{I}}{\int_a^b} c\,\mathrm{d}x := c \cdot (b-a) \qquad \text{and} \qquad \underset{\mathcal{I}}{\int_a^b} [\mathbb{Q}]\,\mathrm{d}x := b-a$$

Then $\mathcal{I}$ fulfils all four properties, but unlike the Lebesgue/Bochner/Henstock–Kurzweil integral, the $\mathcal{I}$-integral of $[\mathbb{Q}]$ is non-zero on all non-empty intervals.