Context Dependent Interpretations

master thesis in computer science

by

Andreas Schnabl

submitted to the Faculty of Mathematics, Computer Science and Physics of the University of Innsbruck

in partial fulfillment of the requirements for the degree of Diplom-Ingenieur

supervisor: Dr. Georg Moser, Institute of Computer Science

Innsbruck, 4 October 2007

Master Thesis

Context Dependent Interpretations

Andreas Schnabl
andreas.schnabl@student.uibk.ac.at

4 October 2007

Supervisor: Dr. Georg Moser

Abstract

Context-dependent interpretations are a termination proof method developed by Hofbauer in 2001. They extend the interpretations into $\mathcal{F}$-algebras by introducing an additional parameter to the interpretation functions. The additional parameter is changed by the context of the evaluated subterm, thus giving rise to the name "context-dependent interpretations". They were designed to give good upper bounds on the derivation height of terms with respect to rewrite systems. In this thesis, the algorithm of Contejean, Marché, Tomás, and Urbain for automatically finding polynomials interpretations to prove termination of rewrite systems is adapted to context-dependent interpretations. We will describe our implementation of this adaptation. Furthermore, we will present a subclass of context-dependent interpretations which induces a quadratic upper bound on the derivational complexity of the considered rewrite system. Finding context-dependent interpretations of this subclass is also part of the implementation, for which we will present some experimental results.

Contents

Acknowledgements

I would like to thank my supervisor Georg Moser for his guidance for the whole duration of this master thesis. His large amounts of feedback and constructive criticism were invaluable for bringing this thesis into its current shape. He also helped a lot at staying on the right track with regard to the structure of the project and pointed out very useful references.

Moreover, I am grateful to Martin Korp, Christian Sternagel, and Harald Zankl for helping me with all my questions about the libraries of TTT and TTT2. I am also grateful towards Simon Bailey, who gave me a lot of technical help with the systems I used to test the performance of the implemented tools. Thanks to Martin Avanzini for providing me with an implementation of LMPO.

I would also like to thank my family for their support throughout my time at university. Their help helped me greatly at staying focused on my studies.

1 Introduction

Derivational complexity analysis is a research field within term rewriting, which is a model of computation. Derivational complexity analysis deals with the maximum length of rewrite sequences admitted by terminating rewrite systems or classes of rewrite systems. The derivational complexity of a rewrite system is a function which maps the size of a term to the maximum number of rewrite steps that can be done starting from any term of that size. For classes of rewrite systems, characterizing the functions that may bound the derivational complexity (function) for rewrite systems of that class is a way to measure an aspect of its strength. One result in this area is the following theorem by Hofbauer [8]:

Theorem 1.1 (Hofbauer 1992)**.** *Termination via the multiset path order implies the existence of a primitive recursive bound on the derivational complexity of the rewrite system.*

A similar result is the following theorem by Weiermann [27]:

Theorem 1.2 (Weiermann 1995)**.** *Termination via the lexicographic path order implies the existence of a multiple recursive bound on the derivational complexity of the rewrite system.*

Primitive recursion and multiple recursion in these theorems mean the same as defined below in Section 2.1. Another result from this area is the following theorem by Hofbauer and Lautemann about polynomial interpretations [10]:

Theorem 1.3 (Hofbauer, Lautemann 1989)**.** *Suppose $\mathcal{R}$ is a rewrite system terminating by polynomial interpretations. Then the derivational complexity function $\mathrm{dc}_{\mathcal{R}}(n)$ of is bounded by $2^{2^{c \cdot n}}$ for some $c \in \mathbb{R}^+$.*

Last, we want to mention the following result by Lepper [11]:

Theorem 1.4 (Lepper 2001)**.** *Suppose $\mathcal{R}$ is a rewrite system terminating by the Knuth-Bendix order (KBO). Then the derivational complexity function $\mathrm{dc}_{\mathcal{R}}(n)$ of is bounded by $\mathrm{Ack}(2^{\mathcal{O}(n)})$.*

With some restrictions, this result has been extended to infinite rewrite systems in 2006 by Moser [23]. All bounds mentioned in these theorems are essentially optimal.

Another application of derivational complexity analysis (and term rewriting in general) are functional programming languages. Functional programs are conceivable as constructor based orthogonal term rewriting systems that take a term as input and return its normal form as output. For example, see [7] for some recent techniques by Giesl, Swiderski, Schneider-Kamp and Thiemann for transforming Haskell programs into rewrite systems. In this case, we are interested in the number of steps it takes to compute the normal form. While the upper bounds in the results cited above are nice for theoretical considerations, they are too huge to be practical for analyzing functional programs. We are interested in logarithmic, linear, polynomial or similar upper bounds on the derivational complexity of rewrite systems, instead.

Methods have been established in order to capture the rewrite systems that represent functions which can be computed by Turing machines in a number of steps that is a polynomial in the size of the input. In such rewrite systems, the application of the function is represented by a term $f(t_1, \ldots, t_n)$, where f is a defined function symbol and the terms $t_1, \ldots, t_n$ only consist of constructor symbols. These terms are particularly interesting for rewrite systems which encode functions in their defined symbols. As shown by Bonfante, Cichon, Marion, and Touzet [3, 4], a subclass of polynomial interpretations can show that such a function encoded by a rewrite system is computable in a polynomial number of steps by a Turing machine. Also, the termination proof methods LMPO by Marion [18] and POP by Arai and Moser [1] can verify that functions encoded by rewrite systems. Like the subclass of polynomial interpretations described in [3, 4], POP is also able to induce a polynomial upper bound on the derivation height of function symbols. However, the property that a rewrite system represents functions computable in a polynomial number of steps by Turing machines and the property that we have a polynomial upper bound on the derivation height of special terms are different from the property that a rewrite system has a polynomial derivational complexity. Also, the polynomial bound holds only for constructor terms $f(t_1, \ldots, t_n)$. Concerning polynomial derivational complexity, we only know of a result from string rewriting, which was presented by Waldmann in a talk in 2006 [26]. In this talk, it was shown that matrix interpretations induce a polynomial upper bound on the derivational complexity if the interpretation has a special shape.

In this thesis, we deal with *context-dependent interpretations*. They were introduced by Hofbauer in 2001 [9]. For interpretations over the domain $\mathbb{N}$ (and in particular, polynomial interpretations), the interpretation of a term is also an upper bound on its derivation height. Context-dependent interpretations are an effort to improve this bound, and at the same to generalize interpretations into the natural numbers. Even though Hofbauer stated that he believes that it might be possible to find context-dependent interpretations automatically, we do not know of any other implementation doing that. We will describe two approaches, which we have implemented, for automatically generating context-dependent interpretations. The first approach is a heuristic given in Hofbauer's paper which transforms given polynomial interpretations over the natural numbers into context-dependent interpretations. The second approach is based on an algorithm to find polynomial interpretations for proving termination automatically, which has been developed by Contejean, Marché, Tomás, and Urbain in 2005 [5]. We have adapted this algorithm such that it works for context-dependent interpretations. Furthermore, we present a subclass of context-dependent interpretations which is well-suited for automatic search by the second method and induces a quadratic (and therefore polynomial) upper bound on the derivational complexity of the considered rewrite system. Experimental results show that our implementation of this approach performs quite well on the Termination Problems Database (TPDB) [17], which is the standard test database for termination proof methods. They also show that from a quantitative point of view, this method compares quiet well to LMPO.

This thesis is organized as follows: In Chapter 2, we recall some basic notions which we will use later on. In particular, we recall polynomial interpretations and present some interesting subclasses of polynomial interpretations. In Chapter 3, we introduce context-dependent interpretations. We also recall some results about context-dependent interpretation which were given by Hofbauer in [9]. Chapter 4 describes the two algorithms to find context-dependent interpretations automatically. In Chapter 5, we present the subclass of context-dependent interpretations which induces a quadratic upper bound on derivational complexity. Furthermore, we show that the restricted class of context-dependent interpretations that we are dealing with is still strong enough to handle rewrite systems which are terminating, but not simply terminating. In Chapter 6, we discuss implementations of the approaches described in Chapter 4. Chapter 7 points out some work that is related to our thesis. We describe LMPO and work by Lucas about proving termination by polynomial interpretations into the real numbers. Finally, Chapter 8 summarizes this thesis and states possible future work about context-dependent interpretations.

2 Preliminaries

2.1 Basic Notions

In this section, we define some basic notions that we use in this thesis. We start with some basic notions from outside the field of term rewriting.

Given two functions $f(x)$ and $g(x)$, we say that $f(x) = \mathcal{O}(g(x))$ if there exist some constants c, y_0 such that for all $y \geq y_0$, we have $f(y) \leq c \cdot g(y)$. As an example, for some $a, b \in \mathbb{R}_0^+$, consider the functions

$$f(x) = a(1+b)(x^2+x) \qquad g(x) = x^2 \quad .$$

Let $c = 2a(b+1)$ and $y_0 = 1$. Then for all $y \geq 1$, we have

$$f(y) = a(1+b)(y^2+y) \leq a(1+b)(y^2+y^2) = 2a(1+b)y^2 = c \cdot g(y)$$

and therefore, $f(x) = \mathcal{O}(g(x))$. Given a series of real numbers $(x_1, x_2, \ldots)$, the *limes inferior* (or *infimum* of the series, denoted by $\inf(x_1, x_2, \ldots)$, is the greatest real number x such that for all $i \in \mathbb{N}$, we have $x \leq x_i$. Similarly, the *limes superior* (or *supremum*) of the series, denoted by $\sup(x_1, x_2, \ldots)$, is the smallest real number x such that for all $i \in \mathbb{N}$, we have $x \geq x_i$. As an example, we have $\inf_{i \in \mathbb{N}} \frac{1}{i+1} = 0$. We say that a function f is in *FP* if there exist a Turing machine M, a polynomial P, and a natural number n_0 and a bijective size-preserving encoding function e such that on input $e(x_1, \ldots, x_n)$, M computes $e(f(x_1, \ldots, x_n))$ in at most $P(n)$ steps whenever $n \geq n_0$, where n is the size of the input x. The set of primitive recursive functions is the smallest set of functions over $\mathbb{N}$ such that

- the *zero function* z with $z(x) = 0$ is primitive recursive,
- the *successor function* s with $s(x) = x + 1$ is primitive recursive,
- the *projection functions* π_i^j for all $j \in \mathbb{N}$ and $i \in \{1, \ldots, j\}$, where $\pi_i^j(x_1, \ldots, x_i, \ldots, x_j) = x_i$
- whenever functions f of arity m and $g_1, \ldots, g_m$ of arity n are primitive recursive, then $h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ is primitive recursive, as well (*closure under composition*),
- and whenever functions g of arity n and h of arity $n+2$ are primitive recursive, then the function f of arity $n+1$ with $f(0, x_1, \ldots, x_n) = g(x_1, \ldots, x_n)$ and $f(k+1, x_1, \ldots, x_n) = h(f(k, x_1, \ldots, x_n), k, x_1, \ldots, x_n)$ is primitive recursive, as well (*closure under primitive recursion*).

An example of a function that grows faster than any primitive recursive function is the *binary Ackermann function*. The binary Ackermann function Ack is defined recursively as follows:

$$\begin{aligned}
\mathrm{Ack}(0, m) &= m + 1 \\
\mathrm{Ack}(n + 1, 0) &= \mathrm{Ack}(n, 1) \\
\mathrm{Ack}(n + 1, m + 1) &= \mathrm{Ack}(n, \mathrm{Ack}(n + 1, m))
\end{aligned}$$

Definition 2.1. For $k > 2$, the k-ary Ackermann function is defined recursively as follows:

$$\begin{aligned}
\mathrm{Ack}(\bar{0}, m) &= m + 1 \\
\mathrm{Ack}(\bar{l}, n + 1, 0) &= \mathrm{Ack}(\bar{l}, n, 1) \\
\mathrm{Ack}(\bar{l}, n + 1, m + 1) &= \mathrm{Ack}(\bar{l}, n, \mathrm{Ack}(\bar{l}, n + 1, m)) \\
\mathrm{Ack}(\bar{l}, n + 1, 0, \bar{0}, m) &= \mathrm{Ack}(\bar{l}, n, m, \bar{0}, m)
\end{aligned}$$

In this definition, $\bar{0}$ is used to denote a sequence of a fixed number of zeroes, and $\bar{l}$ to denote a sequence of a fixed number of variables $l_1, \ldots, l_i$.

Definition 2.2. The set of *multiply recursive* functions is the smallest set of functions over $\mathbb{N}$ which contains the zero function, the successor function, the projection functions, and the binary and all k-ary Ackermann functions and is closed under composition and primitive recursion.

A *multiset* M is a collection of elements from a domain A where each element may occur arbitrarily often. It is characterized by a function $M : A \to \mathbb{N}$, where $M(a) = n$ if the element a occurs in M exactly n times. The *empty multiset*, denoted by $\emptyset$ is the multiset such that for all $a \in A$, we have $\emptyset(a) = 0$. Given multisets $M_1, \ldots, M_n, M, N$ over a domain A, the following operations are defined on them:

- $M \subseteq N$ if for all $a \in A$, we have $M(a) \leq N(a)$
- $X = M \uplus N$ if for all $a \in A$, we have $X(a) = M(a) + N(a)$
- $X = M - N$ if for all $a \in A$, we have $X(a) = \max\{0, M - N\}$
- $X = \max_{\mathrm{mul}}\{M_1, \ldots, M_n\}$ if we have $X(a) = \max\{M_1(a), \ldots, M_n(a)\}$ for all $a \in A$

Given an irreflexive order $>$ on a set A, the *multiset order* $>_{\mathrm{mul}}$ on multisets over A is defined as follows: we have $M >_{\mathrm{mul}} N$ if there exist two multisets X and Y over A such that $N = (M - X) \uplus Y$, $X \neq \emptyset$, and $\forall y \in Y : \exists x \in X : x > y$.

2.2 Term Rewriting

The remainder of this chapter will cover some basics of term rewriting and termination of term rewrite systems. We will only cover the concepts which are relevant to this thesis. For a general introduction to term rewriting, see [2, 22, 25], for instance.

A *term rewrite system* (TRS) $\mathcal{R}$ consists of a *signature* $\mathcal{F}$, a countably infinite set of variables $\mathcal{V}$ that is disjoint from $\mathcal{F}$, and a finite set of rewrite rules $l \to r$, where l and r are terms such that $l \notin \mathcal{V}$ and all variables which occur in r also occur in l. The signature $\mathcal{F}$ defines a set of function symbols, and assigns to each function symbol f a natural number n. This number n denotes the number of arguments of f, and we say that f has *arity* n. We call function symbols with arity 0 *constants*. For the rest of this thesis, we will only consider TRSs with a finite signature and a finite amount of rules. We will also assume that every signature contains at least one constant function symbol.

Definition 2.3. Given a set of functions symbols $\mathcal{F}$ such that each function symbol has a fixed arity, and a set of variables $\mathcal{V}$, the set of terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ over $\mathcal{F}$ and $\mathcal{V}$ is the smallest set such that

1. every variable from $\mathcal{V}$ is in $\mathcal{T}(\mathcal{F}, \mathcal{V})$,
2. every constant function symbol from $\mathcal{F}$ is in $\mathcal{T}(\mathcal{F}, \mathcal{V})$,
3. and whenever we have that terms $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and f is a function symbol of arity $n > 0$, then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$.

The set of terms $\mathcal{T}(\mathcal{F})$ without any variables is called the set of *ground terms* over $\mathcal{F}$. It is defined exactly as $\mathcal{T}(\mathcal{F}, \mathcal{V})$, but the first clause of the definition is dropped.

Definition 2.4. The *root symbol* of a term t, denoted by $\mathrm{root}(t)$ is defined as follows:

$$\begin{aligned} \mathrm{root}(x) &= x && \text{if } x \in \mathcal{V} \\ \mathrm{root}(f(t_1, \ldots, t_n)) &= f && \text{if } f \in \mathcal{F} \text{ and } \mathrm{arity}(f) = n \end{aligned}$$

The *size* of a term t, denoted by $|t|$, is defined as follows:

$$\begin{aligned} |x| &= 1 && \text{if } x \in \mathcal{V} \\ |f(t_1, \ldots, t_n)| &= (\sum_{i=1}^{n} |t_i|) + 1 && \text{if } f \in \mathcal{F} \text{ and } \mathrm{arity}(f) = n \end{aligned}$$

Definition 2.5. A *substitution* is a mapping $\sigma : \mathrm{Dom}(\sigma) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$, and a *ground substitution* is a mapping $\sigma : \mathrm{Dom}(\sigma) \to \mathcal{T}(\mathcal{F})$, where $\mathrm{Dom}(\sigma)$ denotes a finite subset of $\mathcal{V}$. Application of a (ground) substitution σ to a term t (this is denoted by $t\sigma$) replaces all occurrences of variables x where $x \in \mathrm{Dom}(\sigma)$ by $\sigma(x)$.

Definition 2.6. A *context* C is a term from $\mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{V})$ such that $\Box$ occurs exactly once. The symbol $\Box$ denotes a fresh function symbol of arity 0. Application of a context to a term t (this is denoted by $C[t]$) replaces the unique occurrence of $\Box$ in C by t.

With the rewrite rules of a TRS $\mathcal{R}$, we associate the relation $\to_{\mathcal{R}}$, which is defined as follows:

Definition 2.7. Given a TRS $\mathcal{R}$, its *rewrite relation* is the smallest relation over $\mathcal{T}(\mathcal{F}, \mathcal{V})$ that contains all rewrite rules and is closed under contexts and substitutions. If $\mathcal{R}$ is clear from the context, we also write $\to$ instead of $\to_{\mathcal{R}}$. The transitive closure of this relation is denoted by $\to^+$. The reflexive and transitive closure of $\to$ is denoted by $\to^*$. For $n \in \mathbb{N}$, we write $\to^n$ to denote the reflexive closure of the empty relation if $n = 0$, and $\to \cdot \to^{n-1}$ otherwise.

Definition 2.8. A TRS $\mathcal{R}$ *terminates* (we can also say, it is *strongly normalizing*, or $\to_{\mathcal{R}}$ is *Noetherian*) if there exists no infinite chain of terms $t_0, t_1, \ldots$ such that for each $i \in \mathbb{N}$, we have $t_i \to_{\mathcal{R}} t_{i+1}$.

Definition 2.9. A TRS $\mathcal{R}$ is *confluent* if for all terms a, b, and c such that $a \to^* b$ and $a \to^* c$, there exists a term t such that $b \to^* d$ and $c \to^* d$.

Definition 2.10. For a given signature $\mathcal{F}$, the *embedding TRS* for $\mathcal{F}$ (denoted by $\mathcal{E}\mathrm{mb}(\mathcal{F})$) is the rewrite system consisting of the rewrite rules

$$f(x_1, \ldots, x_i, \ldots, x_n) \to x_i$$

for all function symbols f and all $i \in \{1, \ldots, n\}$ with $n = \mathrm{arity}(f)$. The transitive closure of the rewrite relation of an embedding TRS is denoted by $\rhd$, and the reflexive and transitive closure is denoted by $\trianglerighteq$.

If for a TRS $\mathcal{R}$ over a signature $\mathcal{F}$, we have the property that $\mathcal{R} \cup \mathcal{E}\mathrm{mb}(\mathcal{F})$ is terminating, we say that $\mathcal{R}$ is *simply terminating*.

One of the most well-known syntactic methods of proving termination is the *multiset path order* (MPO).

Theorem 2.11. *Given a rewrite system $\mathcal{R}$ and an irreflexive order $>$ on $\mathcal{F}$ ($>$ is called the* precedence*), the multiset path order on $\mathcal{T}(\mathcal{F}, \mathcal{V})$, denoted by $>^{\mathrm{mpo}}$, is defined as follows: Given two terms $t = f(t_1, \ldots, t_n)$ and s, we have $t >^{\mathrm{mpo}} s$ if*

1. *$t_i >^{\mathrm{mpo}} s$ or $t_i = s$ for some $i \in \{1, \ldots, n\}$, or*
2. *$s = g(s_1, \ldots, s_m)$, $f > g$, and $t >^{\mathrm{mpo}} s_i$ for all $i \in \{1, \ldots, m\}$, or*
3. *$s = f(s_1, \ldots, s_n)$ and $(t_1, \ldots, t_n) >^{\mathrm{mpo}}_{\mathrm{mul}} (s_1, \ldots, s_n)$*

If for every rewrite rule $l \to r \in \mathcal{R}$, we have $l >^{\mathrm{mpo}} r$, then $\mathcal{R}$ is terminating. Moreover, $\mathcal{R}$ is simply terminating.

Definition 2.12. Given a TRS R over a signature $\mathcal{F}$, we say that a function symbol $f \in \mathcal{F}$ is a *defined symbol* if there exists a rewrite rule $l \to r \in \mathcal{R}$ such that $f = \mathrm{root}(l)$. Otherwise, we call f a *constructor symbol*. The set of defined symbols is denoted by $\mathcal{F}_{\mathcal{D}}$, and the set of constructor symbols is denoted by $\mathcal{F}_{\mathcal{C}}$. We call a term of the shape $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ a *constructor term* if f is a defined symbol, and the terms $t_1, \ldots, t_n$ do not contain any defined symbols. The TRS $\mathcal{R}$ is a *constructor system* if for every rewrite rule $l \to r \in \mathcal{R}$, l is a constructor term.

Definition 2.13. A confluent constructor system $\mathcal{R}$ *computes* a function $f : (\mathcal{T}(\mathcal{F}_\mathcal{C}))^n \to \mathcal{T}(\mathcal{F}_\mathcal{C})$ if whenever we have $f : x_1, \dots, x_n \mapsto y$, then we also have $f(x_1, \dots, x_n) \to^* y$.

Definition 2.14. For a terminating TRS $\mathcal{R}$, the *derivation height* of a ground term t with respect to $\mathcal{R}$ is defined as

$$\mathrm{dh}_\mathcal{R}(t) = \max\{n \mid \exists s : t \to_\mathcal{R}^n s\} \ .$$

As an example, consider the TRS $\mathcal{R}$ consisting of the two rewrite rules

$$x{+}\mathsf{0} \to x$$
$$x{+}\mathsf{S}(y) \to \mathsf{S}(x{+}y)$$

over the signature containing the constant function symbol $\mathsf{0}$, the unary function symbol S, and the binary function $+$. Consider the ground term $S(0)+S(S(0))$. Then the only possible rewrite sequence is

$$\mathsf{S}(\mathsf{0}){+}\mathsf{S}(\mathsf{S}(\mathsf{0})) \to \mathsf{S}(\mathsf{S}(\mathsf{0}){+}\mathsf{S}(\mathsf{0})) \to \mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{0}){+}\mathsf{0})) \to \mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{0}))) \ .$$

Therefore, we have $\mathrm{dh}_\mathcal{R}(\mathsf{S}(\mathsf{0}){+}\mathsf{S}(\mathsf{S}(\mathsf{0}))) = 3$. Also note that the fact that we only consider finite rewrite systems is essential for the well-definedness of $\mathrm{dh}_\mathcal{R}$. Consider the infinite, but terminating rewrite system $\mathcal{R}$ with the rewrite rule

$$\mathsf{b}(x) \to x$$

and a rewrite rule

$$\mathsf{a} \to \mathsf{b}^n(0)$$

for all $n \in \mathbb{N}$. Then there exists no natural number which bounds the derivation height of the term a.

Definition 2.15. The *derivational complexity* of a TRS $\mathcal{R}$ is the following function $\mathrm{dc}_\mathcal{R} : \mathbb{N} \to \mathbb{N}$:

$$\mathrm{dc}_\mathcal{R}(n) = \max\{\mathrm{dh}_\mathcal{R}(t) \mid |t| = n\}$$

2.3 Polynomial Interpretations

The basic semantic method of proving termination of a TRS is finding an *interpretation into a well-founded monotone algebra.*

Definition 2.16. An $\mathcal{F}$*-algebra* for some signature $\mathcal{F}$ is defined to be a pair $\mathcal{A} = (A, [\cdot]_\mathcal{A})$. We call the set A the *carrier* of the algebra. The other element of the pair, $[\cdot]_\mathcal{A}$, consists of interpretation functions $f_\mathcal{A} : A^n \to A$ for all function symbols $f \in \mathcal{F}$, where n is the arity of f.

Definition 2.17. An *assignment* is a mapping $\alpha : \mathcal{V} \to A$, where A is the carrier of an $\mathcal{F}$-algebra. We denote the application of an interpretation $[\cdot]_\mathcal{A}$ and an assignment α to a term t by $[\alpha]_\mathcal{A}(t)$. It is evaluated as follows:

$$\begin{aligned} [\alpha]_\mathcal{A}(x) &= \alpha(x) && \text{if } x \in \mathcal{V} \\ [\alpha]_\mathcal{A}(f(t_1, \dots, t_n)) &= f_\mathcal{A}([\alpha]_\mathcal{A}(t_1), \dots, [\alpha]_\mathcal{A}(t_n)) && \text{if } f \in \mathcal{F} \text{ and } \mathrm{arity}(f) = n \end{aligned}$$

Definition 2.18. An interpretation function $f_{\mathcal{A}} : A^n \to A$ is *monotone* with respect to a (quasi-)order $>$, if for all $a_1, \ldots, a_n, b \in A$ with $a_i > b$ for some $i \in \{1, \ldots, n\}$, we have

$$f_{\mathcal{A}}(a_1, \ldots, a_i, \ldots, a_n) > f_{\mathcal{A}}(a_1, \ldots, b, \ldots, a_n) \ .$$

If $>$ is not irreflexive, then we also call this property *weak monotonicity.*

A *well-founded monotone $\mathcal{F}$-algebra* is an $\mathcal{F}$-algebra $(A, [\cdot]_{\mathcal{A}})$ equipped with a proper order $>$ such that $>$ is well-founded and for every function symbol $f \in \mathcal{F}$, the interpretation function $f_{\mathcal{A}}$ is monotone with respect to $>$.

Definition 2.19. A well-founded monotone algebra $(A, [\cdot]_{\mathcal{A}}, >)$ is *compatible* with a TRS $\mathcal{R}$ if for every rewrite rule $l \to r \in \mathcal{R}$ and every assignment α, $[\alpha]_{\mathcal{A}}(l) > [\alpha]_{\mathcal{A}}(r)$.

The following theorem relates termination of TRSs and well-founded monotone algebras. A proof of it can be found in [22].

Theorem 2.20. *A rewrite system $\mathcal{R}$ is terminating if and only if there exists a well-founded monotone algebra $\mathcal{A} = (A, [\cdot]_{\mathcal{A}}, >)$ such that $\mathcal{R}$ is compatible with $\mathcal{A}$.*

One of the well-known examples of interpretations into well-founded monotone $\mathcal{F}$-algebras are *polynomial interpretations*:

Definition 2.21. A polynomial interpretation is an interpretation into a well-founded monotone algebra $(A, [\cdot]_{\mathcal{A}}, >)$ such that $A \subseteq \mathbb{N}$, $>$ is the standard order on the natural numbers, and for every function symbol f, the interpretation function is a polynomial.

Lemma 2.22. *Suppose that we have a polynomial interpretation into a well-founded monotone algebra $\mathcal{A}$. Let $\mathcal{R}$ be a TRS. If $\mathcal{A}$ is compatible with $\mathcal{R}$, then $\mathcal{R}$ is terminating, and the following bound on the derivation height holds for all terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and all assignments α:*

$$\mathrm{dh}_{\mathcal{R}}(t) \leq [\alpha]_{\mathcal{A}}(t)$$

Proof. Since every polynomial interpretation is an interpretation into a well-founded monotone algebra, termination of $\mathcal{R}$ follows directly from Theorem 2.20. Because the carrier of the interpretation is a subset of $\mathbb{N}$ and $>$ is the standard order on $\mathbb{N}$, we have that $a > b \iff a - b \geq 1$. Therefore, for all terms t and s such that $t \to^{[\alpha]_{\mathcal{A}}(t)} s$, we have $[\alpha]_{\mathcal{A}}(s) = 0$. Thus, s must be a normal form, and indeed we have $\mathrm{dh}_{\mathcal{R}}(t) \leq [\alpha]_{\mathcal{A}}(t)$. □

2.4 Subclasses of Polynomial Interpretations

In this section, we are looking at two simple subclasses of polynomial interpretations. One of them is very useful for automatically proving termination of rewrite systems, while the other infers a very low bound on the derivational complexity of a rewrite system.

Definition 2.23. A *polynomial interpretation with linear polynomials* is a polynomial interpretation such that for each function symbol f of arity n, the interpretation function $f_{\mathcal{A}}$ has the form

$$f_{\mathcal{A}}(x_1, \dots, x_n) = a_{f,0} + \sum_{i=1}^{n} a_{f,i} x_i \ ,$$

where $a_{f,j} \in \mathbb{N}$ for $j \in \{0, \dots, n\}$.

Linear polynomials are the most commonly used polynomials for automatically generated termination proofs with polynomial interpretations. Other classes of polynomials which may be used for proving termination by polynomial interpretations are the *simple*, *simple-mixed*, and *quadratic* polynomials from Steinbach's classification [24].

Definition 2.24. A *polynomial interpretation with simple polynomials* is a polynomial interpretation such that for each function symbol f of arity n, the interpretation function $f_{\mathcal{A}}$ has the form

$$f_{\mathcal{A}}(x_1, \dots, x_n) = \sum_{i_j \in \{0,1\}} a_{f,i_1,\dots,i_n} x_1^{i_1} \cdot \dots \cdot x_n^{i_n} \ ,$$

with $a_{f,i_1,\dots,i_n} \in \mathbb{N}$.

Definition 2.25. A *polynomial interpretation with simple-mixed polynomials* is a polynomial interpretation such that for each function symbol f of arity n, the interpretation function $f_{\mathcal{A}}$ has the form

$$f_{\mathcal{A}}(x_1, \dots, x_n) = \sum_{i_j \in \{0,1\}} a_{f,i_1,\dots,i_n} x_1^{i_1} \cdot \dots \cdot x_n^{i_n} + \sum_{i=1}^{n} b_i x_i^2 \ ,$$

with $a_{f,i_1,\dots,i_n} \in \mathbb{N}$ and $b_i \in \mathbb{N}$.

Definition 2.26. A *polynomial interpretation with quadratic polynomials* is a polynomial interpretation such that for each function symbol f of arity n, the interpretation function $f_{\mathcal{A}}$ has the form

$$f_{\mathcal{A}}(x_1, \dots, x_n) = \sum_{i_j \in \{0,1,2\}} a_{f,i_1,\dots,i_n} x_1^{i_1} \cdot \dots \cdot x_n^{i_n} \ ,$$

with $a_{f,i_1,\dots,i_n} \in \mathbb{N}$.

A simple subclass of linear polynomials is the class of *additive polynomials*. As we will see, a termination proof by polynomial interpretation with additive polynomial induces a linear upper bound on the derivational complexity of a rewrite system.

Definition 2.27. A *polynomial interpretation with additive polynomials* is a polynomial interpretation such that for each function symbol f of arity n, the interpretation function $f_{\mathcal{A}}$ has the form

$$f_{\mathcal{A}}(x_1, \dots, x_n) = a_f + \sum_{i=1}^{n} x_i \ ,$$

where $a_f \in \mathbb{N}$.

Lemma 2.28. *Suppose that we have a polynomial interpretation with additive polynomials into a well-founded monotone algebra $\mathcal{A}$, and that $\mathcal{A}$ is compatible with $\mathcal{R}$. Then for each ground term t and every assignment α, we have*

$$[\alpha]_{\mathcal{A}}(t) \leq c|t| \ ,$$

where c is the smallest natural number such that for every function symbol f, we have $c_{\mathcal{A}} \geq a_f$. The constant a_f has the same meaning as in Definition 2.27.

Proof. We prove this by induction on the structure of t. Since t is a ground term, it has the structure $f(t_1, \ldots, t_n)$ for some ground terms $t_1, \ldots, t_n$ and a function symbol f of arity n. Then, by unfolding the definition of $[\alpha]_{\mathcal{A}}$, we get

$$\begin{aligned}
[\alpha]_{\mathcal{A}}(t) &= f_{\mathcal{A}}([\alpha]_{\mathcal{A}}(t_1), \ldots, [\alpha]_{\mathcal{A}}(t_n)) \\
&= a_f + \sum_{i=1}^{n} [\alpha]_{\mathcal{A}}(t_i) \\
&\leq a_f + \sum_{i=1}^{n} c|t_i| \ .
\end{aligned}$$

The last line here follows from the induction hypothesis. By definition of c, we know that $a_f \leq c$. Furthermore, by definition of $|\cdot|$, we have $\sum_{i=1}^{n} |t_i| = |t| - 1$. Therefore,

$$\begin{aligned}
a_f + \sum_{i=1}^{n} c|t_i| &\leq c + (\sum_{i=1}^{n} |t_i|)c \\
&= c + (|t| - 1)c \\
&= c|t| \ .
\end{aligned}$$

This concludes

$$[\alpha]_{\mathcal{A}}(t) \leq c|t| \ ,$$

which is what we wanted to show. □

Lemma 2.29. *Let $\mathcal{R}$ be a TRS, and suppose that we have a polynomial interpretation with additive polynomials into a well-founded monotone algebra $\mathcal{A}$, and that $\mathcal{A}$ is compatible with $\mathcal{R}$. Then the following holds:*

$$\mathrm{dc}_{\mathcal{R}}(m) \in \mathcal{O}(m)$$

Proof. By Lemma 2.22, $\mathcal{R}$ is terminating, and we have for all ground terms t and all assignments α

$$\mathrm{dh}_{\mathcal{R}}(t) \leq [\alpha]_{\mathcal{A}}(t) \ .$$

Together with Lemma 2.28, we get

$$\mathrm{dh}_{\mathcal{R}}(t) \leq [\alpha]_{\mathcal{A}}(t) \leq c|t| \in \mathcal{O}(|t|) \ .$$

This concludes

$$\mathrm{dc}_{\mathcal{R}}(m) \in \mathcal{O}(m) \ ,$$

which is what we wanted to show. □

Table 2.1: A simple rewrite system.

$\mathsf{a}(\mathsf{b}(x)) \to \mathsf{b}(\mathsf{a}(x))$

As an example, consider the single-rule TRS $\mathcal{R}$ from Table 2.1 over the signature containing the unary function symbols a and b, and the constant function symbol c. The following polynomial interpretation with linear polynomials shows termination of $\mathcal{R}$:

$$\mathsf{a}_{\mathcal{A}}(x) = 2x \qquad \mathsf{b}_{\mathcal{A}}(x) = x + 1 \qquad \mathsf{c}_{\mathcal{A}} = 0$$

However, for this example, there exists no polynomial interpretation with additive polynomials into a well-founded monotone algebra $\mathcal{A}$ such that $\mathcal{A}$ is compatible with $\mathcal{R}$. If such an interpretation existed, then the interpretation functions for a and b would have the shape

$$\mathsf{a}_{\mathcal{A}}(x) = d + x \quad , \qquad \mathsf{b}_{\mathcal{A}}(x) = e + x \ .$$

Then we would have

$$[\alpha]_{\mathcal{A}}(\mathsf{a}(\mathsf{b}(x))) = d + e + \alpha(x) \ , \qquad [\alpha]_{\mathcal{A}}(\mathsf{b}(\mathsf{a}(x))) = e + d + \alpha(x) \ ,$$

hence

$$[\alpha]_{\mathcal{A}}(\mathsf{a}(\mathsf{b}(x))) = [\alpha]_{\mathcal{A}}(\mathsf{b}(\mathsf{a}(x))) \ ,$$

and therefore

$$[\alpha]_{\mathcal{A}}(\mathsf{a}(\mathsf{b}(x))) \not> [\alpha]_{\mathcal{A}}(\mathsf{b}(\mathsf{a}(x))) \ .$$

Thus, there is no way to establish compatibility of $\mathcal{A}$ with $\mathcal{R}$.

As shown by Bonfante, Cichon, Marion, and Touzet in [3, 4], additive polynomials are also closely related to functions computable in FP:

Definition 2.30. Let $\mathcal{R}$ be a TRS. Suppose that we have a polynomial interpretation such that for each defined symbol f, the interpretation function $f_{\mathcal{A}}$ is a polynomial, and for each constructor symbol g, the interpretation function $g_{\mathcal{A}}$ has the shape

$$g_{\mathcal{A}}(x_1, \ldots, x_n) = a_g + \sum_{i=1}^{n} x_i \ ,$$

where $a_g \in \mathbb{N}$. Then we call the interpretation a $\Pi(0)$-*interpretation*. If all interpretation functions for the defined function symbols are linear/simple/simple-mixed/quadratic polynomials, then we say that the interpretation is a $\Pi(0)$-*interpretation with linear/simple/simple-mixed/quadratic polynomials*, respectively. If $\mathcal{R}$ is a confluent constructor system with a $\Pi(0)$-interpretation, then $\mathcal{R}$ is a $\Pi(0)$-*rewrite system*.

In terms of derivation height, we have the following bound on an extension of the class of $\Pi(0)$-rewrite systems:

Lemma 2.31. *Let $\mathcal{R}$ be a TRS. Suppose that we have a polynomial interpretation such that for each defined symbol f, the interpretation function $f_\mathcal{A}$ is a polynomial, and for each constructor symbol g, the interpretation function $g_\mathcal{A}$ has the shape*

$$g_\mathcal{A}(x_1, \ldots, x_n) = a_g + \sum_{i=1}^{n} x_i \enspace ,$$

where $a_g \in \mathbb{N}$. Then there exists a polynomial P in one variable such that for every ground constructor term t, we have

$$\mathrm{dh}_\mathcal{R}(t) \leq P(|t|) \enspace .$$

Proof. It follows directly from Lemma 2.28 that there exists a constant c such that for each term $t \in \mathcal{T}(\mathcal{F}_\mathcal{C})$, we have $[\alpha]_\mathcal{A}(t) \leq c|t|$. For each defined function symbol f with arity n, the interpretation function $f(x_1, \ldots, x_n)$ is a polynomial P_f in n variables. Therefore, for each ground constructor term of the shape $f(t_1, \ldots, f_n)$, we have the bound $[\alpha]_\mathcal{A}(f(t_1, \ldots, t_n)) \leq P_f(c|t_1|, \ldots, c|t_n|)$. Now define $P'_f(x) = P(x, \ldots, x)$ for all function symbols $f \in \mathcal{F}_\mathcal{D}$ and $P(x) = \sum_{f \in \mathcal{F}_\mathcal{D}} P_f(cx)$. Then for each ground constructor term $t = f(t_1, \ldots, t_n)$, $P'_f(|t|)$ and $P(|t|)$ are obviously polynomials in $|t|$. Furthermore, we have

$$P_f(c|t_1|, \ldots, c|t_n|) \leq P_f(c|t|, \ldots, c|t|) = P'_f(c|t|) \leq P(|t|) \enspace ,$$

and thus $\mathrm{dh}_\mathcal{R}(t) \leq [\alpha]_\mathcal{A}(t) \leq P(|t|)$. □

Definition 2.32. A function $f : A^n \to A$ is $\Pi(0)$-*computable* if there exists a $\Pi(0)$-rewrite system which computes f such that $\mathcal{T}(\mathcal{F}_\mathcal{C}) = A$.

Theorem 2.33 (Bonfante, Cichon, Marion, Touzet 1999)**.** *The set of* $\Pi(0)$-*computable functions is exactly the set of functions computable in FP.*

3 Context-Dependent Interpretations

3.1 Introduction

Context-dependent interpretations were introduced by Hofbauer in 2001 [9]. Like with interpretations into well-founded monotone algebras, we define an interpretation function for each function symbol. However, these interpretation functions take a real parameter, which we will usually denote by Δ, as an additional argument. Additionally, some functions are defined which describe how this parameter Δ changes when we move through the term. Before introducing context-dependent interpretations formally, we reiterate the introductory example of [9], following Hofbauer's presentation, to give a motivation why context-dependent interpretations should be used.

Consider the TRS $\mathcal{R}$ from Table 2.1 over the signature containing the unary function symbols a and b, and the constant function symbol c. One way to prove termination of $\mathcal{R}$ is to construct a polynomial interpretation into a well-founded monotone algebra $\mathcal{B}$. As we have seen in of Section 2.4, a suitable interpretation is

$$\mathsf{a}_{\mathcal{B}}(x) = 2x \qquad \mathsf{b}_{\mathcal{B}}(x) = x + 1 \qquad \mathsf{c}_{\mathcal{B}} = 0 \ .$$

Consider the family of terms $\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))$. Lemma 2.22 allows us to use the bound $\mathrm{dh}_{\mathcal{R}}(\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))) \leq [\alpha]_{\mathcal{B}}(\mathsf{a}^n(\mathsf{b}^m(\mathsf{c})))$ on their derivation height. We have $[\alpha]_{\mathcal{B}}(\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))) = 2^n \cdot m$. However, we can see that for each a in these terms, the rewrite rule can be applied at most m times. Afterward, no more bs remain to the right of this a. Therefore, the maximum number of rewrite steps that can be done starting from $\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))$ is $n \cdot m$. So in this example, Lemma 2.22 heavily overestimates the derivation height of the given terms.

For context-dependent interpretations, an additional parameter which is usually denoted by Δ is introduced into the interpretation functions. In this example, the following interpretation functions ranging over $\mathbb{R}_0^+$ with the parameter Δ ranging over $\mathbb{R}^+$ are suitable:

$$\mathsf{a}_{\mathcal{A}}(\Delta, x) = (1 + \Delta)x \qquad \mathsf{b}_{\mathcal{A}}(\Delta, x) = x + 1 \qquad \mathsf{c}_{\mathcal{A}}(\Delta) = 0$$

Note that these interpretation functions are very similar to the ones that were used in the polynomial interpretation. With these interpretation functions, we can evaluate ground terms of the structure $f(t_1, \dots, t_n)$ as follows:

$$\alpha]_{\mathcal{A}}(\Delta, f(t_1, \dots, t_n)) = f_{\mathcal{A}}(\Delta, [\alpha]_{\mathcal{A}}(f^1_{\mathcal{A}}(\Delta), t_1), \dots, [\alpha]_{\mathcal{A}}(f^n_{\mathcal{A}}(\Delta), t_n))$$

In order to apply this definition, we need to additionally define the functions $f^i_{\mathcal{A}}$ for all function symbols f and $i \in \{1, \dots, n\}$ with $n = \mathrm{arity(f)}$. The intuition behind these functions is that they dictate how the parameter Δ changes as we proceed into the term, adding the "context-dependence" to the interpretation. For the example, the following functions can be used:

$$\mathsf{a}^1_{\mathcal{A}}(\Delta) = \frac{\Delta}{1+\Delta} \qquad \mathsf{b}^1_{\mathcal{A}}(\Delta) = \Delta$$

Because the context-dependency has been added to the interpretation and the carrier is now the set of nonnegative real numbers (which is not well-founded with respect to the standard order on the real numbers), the usual termination criterion cannot be used. Instead, it is required that the interpretation is Δ-*monotone* and Δ-*compatible* with $\mathcal{R}$. Δ-monotonicity requires for every function symbol f, each $i \in \{1, \ldots, n\}$ with $n = \text{arity}(f)$, all $\Delta \in \mathbb{R}^+$, and all $a_1, \ldots, a_n, b \in \mathbb{R}^+_0$, it holds that $b - a_i \geq f^i_{\mathcal{A}}(\Delta)$ implies

$$f_{\mathcal{A}}(\Delta, a_1, \ldots, b, \ldots, a_n) - f_{\mathcal{A}}(\Delta, a_1, \ldots, a_i, \ldots, a_n) \geq \Delta \ .$$

If for all $f \in \mathcal{F}$, the interpretation function $f_{\mathcal{A}}$ is weakly monotone in all arguments with respect to the standard order on the real numbers, then it is sufficient to check just

$$f_{\mathcal{A}}(\Delta, a_1, \ldots, a_i + f^i_{\mathcal{A}}(\Delta), \ldots, a_n) - f_{\mathcal{A}}(\Delta, a_1, \ldots, a_i, \ldots, a_n) \geq \Delta \ .$$

Δ-compatibility requires the same as standard compatibility, i.e. that all rewrite rules are oriented from left to right by the interpretation. However, the difference between the left hand side and the right hand side must be at least Δ, i.e. for all $\Delta \in \mathbb{R}^+$ and all rewrite rules $l \to r \in \mathcal{R}$, the following must hold:

$$[\alpha]_{\mathcal{A}}(\Delta, l) - [\alpha]_{\mathcal{A}}(\Delta, r) \geq \Delta$$

For the example in this section, checking Δ-monotonicity amounts to checking whether the inequalities

$$\mathsf{a}_{\mathcal{A}}(\Delta, x + \mathsf{a}^1_{\mathcal{A}}(\Delta)) - \mathsf{a}_{\mathcal{A}}(\Delta, x) \geq \Delta$$
$$\mathsf{b}_{\mathcal{A}}(\Delta, x + \mathsf{b}^1_{\mathcal{A}}(\Delta)) - \mathsf{b}_{\mathcal{A}}(\Delta, x) \geq \Delta$$

hold for all $x \in \mathbb{R}^+_0$ and $\Delta \in \mathbb{R}^+$. Indeed, we have

$$\mathsf{a}_{\mathcal{A}}(\Delta, x + \mathsf{a}^1_{\mathcal{A}}(\Delta)) - \mathsf{a}_{\mathcal{A}}(\Delta, x) = (1+\Delta)(x + \frac{\Delta}{1+\Delta}) - (1+\Delta)x = \Delta$$

and

$$\mathsf{b}_{\mathcal{A}}(\Delta, x + \mathsf{b}^1_{\mathcal{A}}(\Delta)) - \mathsf{b}_{\mathcal{A}}(\Delta, x) = x + \Delta + 1 - (x+1) = \Delta \ .$$

Therefore, Δ-monotonicity holds for this interpretation. For Δ-compatibility, the following inequality has to be checked:

$$[\alpha]_{\mathcal{A}}(\Delta, \mathsf{a}(\mathsf{b}(x))) - [\alpha]_{\mathcal{A}}(\Delta, \mathsf{b}(\mathsf{a}(x))) \geq \Delta$$

Indeed, we have

$$\begin{aligned}
&[\alpha]_{\mathcal{A}}(\Delta, \mathsf{a}(\mathsf{b}(x))) - [\alpha]_{\mathcal{A}}(\Delta, \mathsf{b}(\mathsf{a}(x))) \\
&\quad = \mathsf{a}_{\mathcal{A}}(\Delta, [\alpha]_{\mathcal{A}}(\mathsf{a}^1_{\mathcal{A}}(\Delta), \mathsf{b}(x))) - \mathsf{b}_{\mathcal{A}}(\Delta, [\alpha]_{\mathcal{A}}(\mathsf{b}^1_{\mathcal{A}}(\Delta), \mathsf{a}(x))) \\
&\quad = (1+\Delta)\mathsf{b}_{\mathcal{A}}(\frac{\Delta}{1+\Delta}, x) - (\mathsf{a}_{\mathcal{A}}(\Delta, x) + 1) \\
&\quad = (1+\Delta)(\alpha(\frac{\Delta}{1+\Delta}, x) + 1) - ((1+\Delta)\alpha(\frac{\Delta}{1+\Delta}, x) + 1) \\
&\quad = \Delta \ .
\end{aligned}$$

Therefore, by Theorem 3.6 below, $\mathcal{R}$ is terminating. Furthermore, by Theorem 3.8 below, the following upper bound on the derivation height holds for every ground term t:

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in D} \frac{[\alpha]_{\mathcal{A}}(\Delta, t)}{\Delta}$$

For the example family of terms $\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))$, we have

$$[\alpha]_{\mathcal{A}}(\Delta, \mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))) = (1 + \Delta n)m \ ,$$

which can be proved by induction on n. For the base case, we obviously have $[\alpha]_{\mathcal{A}}(\Delta, \mathsf{b}^m(\mathsf{c})) = m$. For the induction step,

$$\begin{aligned}
[\alpha]_{\mathcal{A}}(\Delta, \mathsf{a}(\mathsf{a}^n(\mathsf{b}^m(\mathsf{c})))) &= (1+\Delta)[\alpha]_{\mathcal{A}}(\frac{\Delta}{1+\Delta}, \mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))) \\
&= (1+\Delta)(1 + \frac{\Delta}{1+\Delta} n)m \\
&= (1 + \Delta + \Delta n)m \\
&= (1 + \Delta(n+1))m \ ,
\end{aligned}$$

where the second line follows from the induction hypothesis. Finally, Theorem 3.8 yields the following bound on the derivation height:

$$\mathrm{dh}_{\mathcal{R}}(\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))) \leq \inf_{\Delta \in \mathbb{R}^+} \frac{(1+\Delta n)m}{\Delta} = n \cdot m \ ,$$

which is optimal for this example. This shows that context-dependent interpretations are capable of inducing stronger upper bounds on the derivation height of terms than polynomial interpretations.

3.2 Context-Dependent Interpretations

Having reviewed Hofbauer's motivating example for context-dependent interpretations, we will now introduce them more formally.

Definition 3.1. A *context-dependent algebra* is a triple $\mathcal{A} = \{A, D, [\cdot]_{\mathcal{A}}\}$. Again, the set A is the carrier of the algebra. The last element of the triple, $[\cdot]_{\mathcal{A}}$, contains interpretation functions $f_{\mathcal{A}} : D \times A^n \to A$ for every function symbol $f \in \mathcal{F}$, where n is the arity of f. Additionally, for each $i \in \{1, \ldots, n\}$, we have a function $f^i_{\mathcal{A}} : D \to D$.

Definition 3.2. A Δ-*assignment* is a mapping $\alpha : D \times \mathcal{V} \to A$, where the triple $\{A, D, [\cdot]_{\mathcal{A}}\}$ is a context-dependent algebra. We denote the application of a context-dependent interpretation $[\cdot]_{\mathcal{A}}$ and a Δ-assignment α to a term t by $[\alpha]_{\mathcal{A}}(t)$. It is evaluated as follows:

$$\begin{aligned}[\alpha]_{\mathcal{A}}(\Delta, x) &= \alpha(\Delta, x)\\ [\alpha]_{\mathcal{A}}(\Delta, f(t_1, \dots, t_n)) &= f_{\mathcal{A}}(\Delta, [\alpha]_{\mathcal{A}}(f^1_{\mathcal{A}}(\Delta), t_1), \dots, [\alpha]_{\mathcal{A}}(f^n_{\mathcal{A}}(\Delta), t_n)),\end{aligned}$$

where $x \in \mathcal{V}$, $f \in \mathcal{F}$, and $\text{arity}(f) = n$. Because of the special role of the additional parameter, we will write $\alpha[\Delta](x)$, $[\alpha]_{\mathcal{A}}[\Delta](t)$, and $f_{\mathcal{A}}[\Delta](t_1, \dots, t_n)$ instead of $\alpha(\Delta, x)$ $[\alpha]_{\mathcal{A}}(\Delta, t)$, and $f_{\mathcal{A}}(\Delta, t_1, \dots, t_n)$, respectively for the rest of this thesis.

Definition 3.3. Let $(A, D, [\cdot]_{\mathcal{A}})$ be a context-dependent algebra, and $\{>_\Delta \mid \Delta \in D\}$ a set of proper orders. We say that $[\cdot]_{\mathcal{A}}$ is Δ-monotone with respect to the set of orders $>_\Delta$ if for all $\Delta \in D$, $a_1, \dots, a_n, b \in A$ with $a_i >_{f^i_{\mathcal{A}}(\Delta)} b$ for some $i \in \{1, \dots, n\}$, we have

$$f_{\mathcal{A}}[\Delta](a_1, \dots, a_i, \dots, a_n) >_\Delta f_{\mathcal{A}}[\Delta](a_1, \dots, b, \dots, a_n).$$

Furthermore, we say that $[\cdot]_{\mathcal{A}}$ is *monotone* with respect to a (quasi-)order $>$, if for all $\Delta \in D$, $a_1, \dots, a_n, b \in A$ with $a_i > b$ for some $i \in \{1, \dots, n\}$, we have

$$f_{\mathcal{A}}[\Delta](a_1, \dots, a_i, \dots, a_n) > f_{\mathcal{A}}[\Delta](a_1, \dots, b, \dots, a_n).$$

If $>$ is not irreflexive, then we also call this property *weak monotonicity*.

We say that a *well-founded Δ-monotone algebra* is a context-dependent algebra $(A, D, [\cdot]_{\mathcal{A}})$ equipped with proper orders $>_\Delta$ for all $\Delta \in D$ such that $[\cdot]_{\mathcal{A}}$ is Δ-monotone with respect to these orders, and all $>_\Delta$ are well-founded.

Definition 3.4. A well-founded Δ-monotone algebra $(A, D, [\cdot]_{\mathcal{A}}, \{>_\Delta \mid \Delta \in D\})$ is Δ-*compatible* with a TRS $\mathcal{R}$ if for every rewrite rule $l \to r \in \mathcal{R}$, every $\Delta \in D$, and every Δ-assignment α, we have

$$[\alpha]_{\mathcal{A}}[\Delta](l) >_\Delta [\alpha]_{\mathcal{A}}[\Delta](r).$$

Lemma 3.5. *Let $\mathcal{R}$ be a TRS and $\mathcal{A}$ a well-founded Δ-monotone algebra that is Δ-compatible with $\mathcal{R}$. Then for all terms s and t, and all $\Delta \in D$, we have*

$$s \to t \Longrightarrow s >_\Delta t.$$

Proof. This lemma can be proved by induction on s. If the rewrite step takes place at the root position of s, then the lemma follows immediately from Δ-monotonicity. If the rewrite step takes place below the root position, then s must have the structure $f(s_1, \dots, s_i, \dots, s_n)$, and the rewrite step takes place in s_i for some $i \in \{1, \dots, n\}$. Then we have $s_i \to t_i$ and $t = f(s_1, \dots, t_i, \dots, s_n)$. From the induction hypothesis, we get

$$[\alpha]_{\mathcal{A}}[f^i_{\mathcal{A}}(\Delta)](s_i) >_{f^i_{\mathcal{A}}(\Delta)} [\alpha]_{\mathcal{A}}[f^i_{\mathcal{A}}(\Delta)](t_i).$$

Together with Δ-monotonicity, this yields

$$f_{\mathcal{A}}[\Delta](\ldots,[\alpha]_{\mathcal{A}}[f^i_{\mathcal{A}}(\Delta)](s_i),\ldots) >_\Delta f_{\mathcal{A}}[\Delta](\ldots,[\alpha]_{\mathcal{A}}[f^i_{\mathcal{A}}(\Delta)](t_i),\ldots)$$

Thus,

$$[\alpha]_{\mathcal{A}}(\ldots,s_i,\ldots) >_\Delta [\alpha]_{\mathcal{A}}(\ldots,t_i,\ldots),$$

which is what we wanted to show. □

Theorem 3.6. *A rewrite system $\mathcal{R}$ is terminating if and only if there exists a well-founded Δ-monotone algebra $\mathcal{A} = (A, D, [\cdot]_{\mathcal{A}}, \{>_\Delta \mid \Delta \in D\})$ that is Δ-compatible with $\mathcal{R}$.*

Proof. The "if" part follows immediately from the well-foundedness of the orders $>_\Delta$ and Lemma 3.5. For the "only if", we construct a well-founded monotone algebra $\mathcal{A} = (A, D, [\cdot]_{\mathcal{A}}, \{>_\Delta \mid \Delta \in D\})$ that is Δ-compatible with $\mathcal{R}$. By Theorem 2.20, there is a well-founded monotone algebra $\mathcal{B} = (B, [\cdot]_{\mathcal{B}}, >)$ that is compatible with $\mathcal{R}$. Then let $A = B$, $D = \{0\}$, $>_0 = >$, and $[\cdot]_{\mathcal{A}}$ be constructed as follows; for each function symbol f of arity n, let $f_{\mathcal{A}}[\Delta](x_1,\ldots,x_n) = f_{\mathcal{B}}(x_1,\ldots,x_n)$, and for every $i \in \{1,\ldots,n\}$, let $f_{\mathcal{A}}(\Delta) = \Delta$. Since $>$ is well-founded, $>_0$ is obviously well-founded, as well. Monotonicity of $\mathcal{B}$ implies for all $a_i > b$:

$$f_{\mathcal{B}}(a_1,\ldots,a_i,\ldots,a_n) > f_{\mathcal{B}}(a_1,\ldots,b,\ldots,a_n)$$
$$f_{\mathcal{A}}[\Delta](a_1,\ldots,a_i,\ldots,a_n) >_0 f_{\mathcal{A}}[\Delta](a_1,\ldots,b,\ldots,a_n)$$

As $f^i_{\mathcal{A}}(\Delta) = \Delta$ and $>_0 = >$, this concludes Δ-monotonicity of $\mathcal{A}$. Hence, $\mathcal{A}$ is a well-founded Δ-monotone algebra.

For Δ-compatibility, we first prove the following claim: For every assignment α, there is a Δ-assignment α_0 such that for every term t, we have $[\alpha]_{\mathcal{B}}(t) = [\alpha_0]_{\mathcal{A}}[0](t)$. The Δ-assignment α_0 is created from α by setting $\alpha_0[0](x) = \alpha(x)$. This claim is proved by induction on the structure of t. If t is a variable, then the claim follows from the definition of α_0. If $t = f(t_1,\ldots,t_n)$ then the claim follows from the induction hypotheses and the definitions of $f_{\mathcal{A}}$ and $f_{\mathcal{B}}$, which concludes the inductive proof of the claim.

As can be easily seen, for every Δ-assignment α_0, there is an assignment α such that α_0 was constructed from α as described in the above claim. Therefore, for every Δ-assignment α_0, there is an assignment α such that for every rewrite rule $l \to r \in \mathcal{R}$, $[\alpha_0]_{\mathcal{A}}[0](l) >_0 [\alpha_0]_{\mathcal{A}}[0](r)$ follows immediately from $[\alpha]_{\mathcal{B}}(l) > [\alpha]_{\mathcal{B}}(r)$. Thus we can conclude Δ-compatibility of $\mathcal{A}$ and therefore the theorem. □

We have just introduced context-dependent interpretations along similar lines of thought as Hofbauer did in [9], but on a slightly more abstract level. Hofbauer's context-dependent interpretations are a special case of the interpretations into well-founded Δ-monotone algebras defined below, and we will call them *context-dependent interpretations over the reals* in the following.

Definition 3.7. A *context-dependent interpretation over the reals* is an interpretation into a well-founded Δ-monotone algebra $\mathcal{A} = (A, D, [\cdot]_{\mathcal{A}}, \{>_\Delta | \ \Delta \in D\})$ such that $A \subseteq \mathbb{R}_0^+$, $D \subseteq \mathbb{R}^+$, and for all $\Delta \in D$:

$$a >_\Delta b \iff a - b \geq \Delta$$

As can be easily seen, every order $>_\Delta$ is well-founded on subsets of the nonnegative real numbers. Note that if for every function symbol f, the function $f_\tau[\Delta]$ is weakly monotone with respect to the standard order $\geq$ on $\mathcal{R}_0^+$, checking Δ-monotonicity just amounts to checking whether

$$f_\tau[\Delta](x_1, \dots, x_i + f_\tau^i(\Delta), \dots, x_n) - f_\tau[\Delta](x_1, \dots, x_i, \dots, x_n) \geq \Delta.$$

Theorem 3.8 (Hofbauer 2001)**.** *Let $\mathcal{R}$ be a TRS. Suppose that we have a context-dependent interpretation over the reals into a well-founded Δ-monotone algebra $\mathcal{A}$. If $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$, then $\mathcal{R}$ is terminating and the following bound on the derivation height holds for all terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and all Δ-assignments α:*

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in D} \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$$

Proof. Since every context-dependent interpretation over the reals is an interpretation into a well-founded Δ-monotone algebra, termination of $\mathcal{R}$ follows directly from Theorem 3.6. By Lemma 3.5, each rewrite step subtracts at least Δ from the interpretation of a term. Furthermore, the carrier of the $\mathcal{A}$ must be a subset of $\mathbb{N}$. Therefore, for all $\Delta \in D$ and all terms t and s such that $t \to^n s$ where $n = \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$, we have $[\alpha]_{\mathcal{A}}[\Delta](s) = 0$. Thus, s must be a normal form, and indeed we have $\mathrm{dh}_{\mathcal{R}}(t) \leq \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$. Since this holds for all $\Delta \in D$, we also have $\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in D} \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$, which is what we wanted to show. □

3.3 A special case of context-dependent interpretations

Above we have seen a definition of context-dependent interpretations and the subclass of context-dependent interpretations over the reals. We have also seen that they can be used to prove termination of rewrite systems. Hofbauer has also shown in [9] that context-dependent interpretations over the reals are also a generalization of polynomial interpretations. In this section, we give a presentation of this result of Hofbauer. In polynomial interpretations into some well-founded monotone algebra $\mathcal{A}$, we usually take $\mathbb{N}$ as carrier and the interpretation functions $f_{\mathcal{A}} : \mathbb{N}^n \to \mathbb{N}$ for all function symbols $f \in \mathcal{F}$ are polynomial functions. We can now extend the domain of the interpretation functions from $\mathbb{N}$ to $\mathbb{R}_0^+$. We can assume the original functions to be monotone with respect to the standard order $>$ on $\mathbb{N}$. For natural numbers, the standard order coincides with the order $>_1$, which is defined as $a >_1 b \iff a - b \geq 1$. This property is preserved if we extend the function to the domain $\mathbb{R}_0^+$ by defining an algebra $\mathcal{B}$

with the carrier $\mathbb{R}_0^+$ and the interpretation functions

$$
\begin{aligned}
&f_{\mathcal{B}}(k_1 + x_1, \ldots, k_n + x_n) \\
&\qquad = \sum_{b_i \in \{0,1\}} (f_{\mathcal{A}}(k_1 + b_1, \ldots, k_n + b_n) \prod_{i=1}^{n} ((1 - b_i)(1 - x_i) + b_i x_i)) \ ,
\end{aligned}
$$

where $a_i = k_i + x_i$ with $k_i \in \mathbb{N}$ and $0 \leq x_i < 1$. This function is constructed as follows: each argument in this function is between two natural numbers (k_i and $k_i + 1$, or simply $k_i + b_i$), for which the result can be computed by the original interpretation function. For each argument, this function adds weighted results of $f_{\mathcal{A}}(\ldots, k_i, \ldots)$ and $f_{\mathcal{A}}(\ldots, k_i + 1, \ldots)$. The weight factors are computed in the big product in the above formula. For $b_i = 0$, the weight factor is $1 - x_i$, and for $b_i = 1$, the weight factor is x_i. This means that the closer a_i is towards $k_i + 1$, the more weight $f_{\mathcal{A}}(\ldots, k_i + 1, \ldots)$ gains, and vice versa. Finally, the above formula employs this mechanism for not only one, but all arguments of $f_{\mathcal{B}}$.

To finish the definition of the context-dependent interpretation over the reals, we have to specify how the parameter Δ is handled by the interpretation. For that purpose, we define a context-dependent algebra $\mathcal{C} = (\mathbb{R}_0^+, \mathbb{R}^+, [\cdot]_{\mathcal{C}})$ with the following interpretation functions $f_{\mathcal{C}}$ and $f_{\mathcal{C}}^i$:

$$
\begin{aligned}
f_{\mathcal{C}}[\Delta](a_1, \ldots, a_n) &= \Delta f_{\mathcal{B}}(a_1/\Delta, \ldots, a_n/\Delta) \\
f_{\mathcal{C}}^i(\Delta) &= \Delta
\end{aligned}
$$

Lemma 3.9. *Suppose that we have an interpretation into an algebra $\mathcal{B} = (\mathbb{R}_0^+, [\cdot]_{\mathcal{B}})$ which is monotone and compatible with a TRS $\mathcal{R}$ with respect to the order $>_1$ on the real numbers. Then the interpretation into the context-dependent algebra $\mathcal{C}$ yielded by the above construction fulfills the following properties:*

1. *for all ground terms t, all assignments α, and all Δ-assignments α' such that $\alpha'[\Delta](x) = \Delta\alpha(x)$ for every variable x, the equality $[\alpha']_{\mathcal{C}}[\Delta](t) = \Delta[\alpha]_{\mathcal{B}}(t)$ holds*

2. *$\mathcal{C}$ is Δ-compatible with $\mathcal{R}$ with respect to the set of orders $\{>_\Delta \mid \Delta \in \mathbb{R}^+\}$ from Definition 3.7*

3. *$\mathcal{C}$ is Δ-monotone with respect to the set of orders $\{>_\Delta \mid \Delta \in \mathbb{R}^+\}$ from Definition 3.7*

Proof. Property 1 can be verified by structural induction on t. If t is a variable, then the property holds by the assumptions about α and α'. If t is a constant, then the property holds by the definition of $f_{\mathcal{C}}$, which concludes the base cases. If t has the shape $f(t_1, \ldots, t_n)$ where f is a function symbol with arity $n > 0$, then expanding the definitions of $f_{\mathcal{C}}$ and $f_{\mathcal{C}}^i$ yields:

$$\begin{aligned}[\alpha']_{\mathcal{C}}[\Delta](\ldots,t_i,\ldots) &= f_{\mathcal{C}}[\Delta](\ldots,[\alpha']_{\mathcal{C}}[f_{\mathcal{C}}^i(\Delta)](t_i),\ldots)\\ &= f_{\mathcal{C}}[\Delta](\ldots,[\alpha']_{\mathcal{C}}[\Delta](t_i),\ldots)\\ &= f_{\mathcal{C}}[\Delta](\ldots,\Delta[\alpha]_{\mathcal{B}}(t_i),\ldots)\\ &= \Delta f_{\mathcal{B}}(\ldots,\Delta[\alpha]_{\mathcal{B}}(t_i)/\Delta,\ldots)\\ &= \Delta f_{\mathcal{B}}(\ldots,[\alpha]_{\mathcal{B}}(t_i),\ldots)\\ &= \Delta[\alpha]_{\mathcal{B}}(f(\ldots,t_i,\ldots))\end{aligned}$$

The third line in this proof follows from the induction hypothesis, the rest is straightforward.

For property 2, we know that the interpretation into $\mathcal{B}$ is compatible with $\mathcal{R}$ with respect to the order $>_1$, therefore $[\alpha]_{\mathcal{B}}(l)-[\alpha]_{\mathcal{B}}(r) \geq 1$ for every rewrite rule $l \to r$ in $\mathcal{R}$ and all assignments α. Furthermore, we can conclude $[\alpha']_{\mathcal{C}}[\Delta](l) - [\alpha']_{\mathcal{C}}[\Delta](r) = \Delta[\alpha]_{\mathcal{B}}(l) - \Delta[\alpha]_{\mathcal{B}}(r) \geq \Delta$ from property 1 whenever $\alpha(x) = \alpha'[\Delta](x)/\Delta$ for all variables x. Since $\Delta[\alpha]_{\mathcal{B}}(l) - \Delta[\alpha]_{\mathcal{B}}(r) \geq \Delta$ holds for all assignments α, we also have $[\alpha']_{\mathcal{C}}[\Delta](l) - [\alpha']_{\mathcal{C}}[\Delta](r) \geq \Delta$ for all Δ-assignments α'.

For property 3, we know that the interpretation into $\mathcal{B}$ is monotone with respect to the order $>_1$, so we have $f_{\mathcal{B}}(\ldots,b_i',\ldots) - f_{\mathcal{B}}(\ldots,b_i,\ldots) \geq 1$ whenever $b_i' - b_i \geq 1$. We need to show that $f_{\mathcal{C}}[\Delta](\ldots,a_i',\ldots) - f_{\mathcal{C}}[\Delta](\ldots,a_i,\ldots) \geq \Delta$ whenever $a_i' - a \geq \Delta$, which is equivalent to the inequality $\Delta f_{\mathcal{B}}(\ldots,a_i'/\Delta,\ldots) - \Delta f_{\mathcal{B}}(\ldots,a_i/\Delta,\ldots) \geq \Delta$ or $f_{\mathcal{B}}(\ldots,a_i'/\Delta,\ldots) - f_{\mathcal{B}}(\ldots,a_i/\Delta,\ldots) \geq 1$. We know that $a_i' - a_i \geq \Delta$, which is equivalent to $a_i'/\Delta - a_i/\Delta \geq 1$. Therefore, $f_{\mathcal{B}}(\ldots,a_i'/\Delta,\ldots) - f_{\mathcal{B}}(\ldots,a_i/\Delta,\ldots) \geq 1$ follows from the monotonicity of $\mathcal{B}$ with respect to the order $>_1$, which holds by assumption. □

Lemma 3.9 is a modification of Lemma 2 from [9]. Hofbauer's version of this lemma requires a polynomial interpretation over $\mathbb{N}$. This interpretation is extended to an interpretation over $\mathbb{R}_0^+$ (while preserving 1-monotonicity and also monotonicity) first, then it is transformed into a Δ-monotone (and also monotone) context-dependent interpretation. However, Lemma 3.9 starts with a 1-monotone polynomial interpretation into $\mathbb{R}_0^+$ and only does the second step in the construction. In this way, we construct context-dependent interpretations which are not necessarily weakly monotone, but still imply termination of a rewrite system. We will use this further below in an example to prove termination of a non-simply terminating rewrite system.

3.4 Examples: Non-Simple Termination

In [9], it has already been shown by Hofbauer that context-dependent interpretations are able to handle rewrite systems which are terminating, but not simply terminating. We reiterate his example that showed that Theorem 3.8 can also be applicable to non-simply terminating rewrite systems. Consider the TRS $\mathcal{R}$ with the single rewrite rule

$$\mathsf{a}(\mathsf{a}(x)) \to \mathsf{a}(\mathsf{b}(\mathsf{a}(x))) \ .$$

Consider the interpretation into the following algebra $\mathcal{B} = (\mathbb{R}_0^+, [\cdot]_\mathcal{B})$ equipped with the order $>_1$ from Section 3.3, where $[\cdot]_\mathcal{B}$ is defined by the interpretation functions

$$\mathsf{a}_\mathcal{B}(x) = \lceil x \rceil + \frac{1}{2} \qquad \mathsf{b}_\mathcal{B}(x) = \lceil x - \frac{1}{2} \rceil \ .$$

Now we check that monotonicity and compatibility with $\mathcal{R}$ hold for this interpretation. Since all interpretation functions are weakly monotone, it suffices to check the two inequalities

$$\mathsf{a}_\mathcal{B}(x+1) - \mathsf{a}_\mathcal{B}(x) \geq 1 \qquad \mathsf{b}_\mathcal{B}(x+1) - \mathsf{b}_\mathcal{B}(x) \geq 1$$

in order to verify monotonicity. For the first inequality, we have

$$\begin{aligned} \mathsf{a}_\mathcal{B}(x+1) - \mathsf{a}_\mathcal{B}(x) &= \lceil x+1 \rceil + \frac{1}{2} - (\lceil x \rceil + \frac{1}{2}) \\ &= \lceil x \rceil + 1 + \frac{1}{2} - (\lceil x \rceil + \frac{1}{2}) \\ &= 1 \ . \end{aligned}$$

The second inequality can be checked in a similar fashion. Hence, monotonicity holds with respect to $>_1$. All that is left to show is compatibility with $\mathcal{R}$ with respect to $>_1$. An even stronger property can be shown, namely

$$[\alpha]_\mathcal{B}(\mathsf{a}(\mathsf{a}(x))) - [\alpha]_\mathcal{B}(\mathsf{a}(\mathsf{b}(\mathsf{a}(x)))) = 1$$

for all assignments α. Expanding the left hand side of this equality yields

$$\begin{aligned} &[\alpha]_\mathcal{B}(\mathsf{a}(\mathsf{a}(x))) - [\alpha]_\mathcal{B}(\mathsf{a}(\mathsf{b}(\mathsf{a}(x)))) \\ &\quad = \lceil [\alpha]_\mathcal{B}(\mathsf{a}(x)) \rceil + \frac{1}{2} - (\lceil [\alpha]_\mathcal{B}(\mathsf{b}(\mathsf{a}(x))) \rceil + \frac{1}{2}) \\ &\quad = \lceil [\alpha]_\mathcal{B}(\mathsf{a}(x)) \rceil - \lceil [\alpha]_\mathcal{B}(\mathsf{b}(\mathsf{a}(x))) \rceil \\ &\quad = \lceil \lceil [\alpha]_\mathcal{B}(x) \rceil + \frac{1}{2} \rceil - \lceil \lceil [\alpha]_\mathcal{B}(\mathsf{a}(x)) - \frac{1}{2} \rceil \rceil \\ &\quad = \lceil \lceil [\alpha]_\mathcal{B}(x) \rceil + \frac{1}{2} \rceil - \lceil \lceil \lceil [\alpha]_\mathcal{B}(x) \rceil + \frac{1}{2} - \frac{1}{2} \rceil \rceil \ . \end{aligned}$$

Obviously, $\lceil [\alpha]_\mathcal{B}(x) \rceil$ is a natural number. Therefore, we have $\lceil \lceil [\alpha]_\mathcal{B}(x) \rceil + \frac{1}{2} \rceil = \lceil [\alpha]_\mathcal{B}(x) \rceil + 1$. We apply this and simplify both sides of the "$-$" some more, and get

$$\begin{aligned} &\lceil \lceil [\alpha]_\mathcal{B}(x) \rceil + \frac{1}{2} \rceil - \lceil \lceil \lceil [\alpha]_\mathcal{B}(x) \rceil + \frac{1}{2} - \frac{1}{2} \rceil \rceil \\ &\quad = \lceil [\alpha]_\mathcal{B}(x) \rceil + 1 - \lceil \lceil \lceil [\alpha]_\mathcal{B}(x) \rceil + \frac{1}{2} - \frac{1}{2} \rceil \rceil \\ &\quad = \lceil [\alpha]_\mathcal{B}(x) \rceil + 1 - \lceil \lceil \lceil [\alpha]_\mathcal{B}(x) \rceil \rceil \rceil \\ &\quad = \lceil [\alpha]_\mathcal{B}(x) \rceil + 1 - \lceil [\alpha]_\mathcal{B}(x) \rceil \\ &\quad = 1 \ , \end{aligned}$$

which concludes compatibility with respect to the order $>_1$. Now we can use the construction from Section 3.3 in order to construct a context-dependent algebra $\mathcal{C}$ equipped with the orders $\{>_\Delta \mid \Delta \in \mathbb{R}^+\}$ from Definition 3.7. By Lemma 3.9, $\mathcal{C}$ is Δ-monotone and Δ-compatible with $\mathcal{R}$. We apply Theorem 3.8 to conclude termination of $\mathcal{R}$ and the upper bound

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in D} \frac{[\alpha']_{\mathcal{C}}[\Delta](t)}{\Delta}$$

for all ground terms t. Together with part 1 of Lemma 3.9, this implies that we have the upper bound $\mathrm{dh}_{\mathcal{R}}(t) \leq [\alpha]_{\mathcal{B}}(t)$ on the derivation height for all ground terms t.

Lemma 3.10. *If a term $t \in \mathcal{T}(\mathcal{F})$ is a normal form with respect to $\mathcal{R}$, then $\lfloor [\alpha]_{\mathcal{B}}(t) \rfloor = 0$.*

Proof. First we show by induction on the length of t that any ground term t' which is a normal form with respect to $\mathcal{R}$ and does not have an a as outermost function symbol fulfills the property $[\alpha]_{\mathcal{B}}(t') = 0$. The base case $(t = \mathsf{c})$ holds trivially. For the step case, we have the induction hypothesis that for all subterms t'' of t' whose outermost function symbol is not an a, we have $[\alpha]_{\mathcal{B}}(t'') = 0$, in particular for the outermost such subterm t''. The next function symbol to the left of this t'' is either an a or a b. If it is a b, then we have $\mathsf{b}_{\mathcal{B}}([\alpha]_{\mathcal{B}}(t'')) = \mathsf{b}_{\mathcal{B}}(0) = 0$, which is what we wanted to show. If it is an a, then $\mathsf{a}(t'')$ cannot be equal to t' because of the assumption that a may not be the outermost function symbol of t'. Therefore, there must be another function symbol to the left of $\mathsf{a}(t'')$. That function symbol may not be another a, since the property that t' must be a normal form with respect to $\mathcal{R}$ would be violated in that case. Therefore, the next function symbol must be a b, and we have $t' = \mathsf{b}(\mathsf{a}(t''))$ and $\mathsf{b}_{\mathcal{B}}(\mathsf{a}_{\mathcal{B}}([\alpha]_{\mathcal{B}}(t''))) = \mathsf{b}_{\mathcal{B}}(\frac{1}{2}) = 0$, which is what we wanted to show. This concludes the first claim.

To prove the actual lemma, we have to distinguish two cases. Either the outermost symbol of t is not an a, then the lemma holds because of the above claim. Otherwise, t has the structure $\mathsf{a}(t')$. The outermost function symbol of t' cannot be an a (otherwise, t would not be a normal form with respect to $\mathcal{R}$). Therefore, $[\alpha]_{\mathcal{B}}(t') = 0$, which concludes $\lfloor [\alpha]_{\mathcal{B}}(\mathsf{a}(t')) \rfloor = \lfloor \frac{1}{2} \rfloor = 0$, which is what we wanted to show. $\square$

As shown above, every rewrite step reduces the interpretation of a term in the algebra $\mathcal{B}$ by exactly 1. From Lemma 3.10, we know that the interpretation of a normal form must be smaller than 1. This concludes that $\lfloor [\alpha]_{\mathcal{B}}(t) \rfloor = \mathrm{dh}(t)$, so the interpretation gives an optimal upper bound on the derivation length for all ground terms in this example.

Now we will look at another example which is not simply terminating. Consider the non-simply terminating rewrite system $\mathcal{R}$ with the rewrite rules

$$\begin{aligned} \mathsf{half}(\mathsf{0}) &\to \mathsf{0} & \mathsf{bits}(\mathsf{0}) &\to \mathsf{0} \\ \mathsf{half}(\mathsf{s}(\mathsf{0})) &\to \mathsf{0} & \mathsf{bits}(\mathsf{s}(x)) &\to \mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x)))) \\ \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) &\to \mathsf{s}(\mathsf{half}(x)) \end{aligned}$$

Again, we define an algebra $\mathcal{B}$ with the carrier $\mathbb{R}_0^+$ and the interpretation functions

$$
\begin{aligned}
\mathsf{0}_\mathcal{B} &= 0 \\
\mathsf{s}_\mathcal{B}(z) &= \lfloor z + \frac{1}{3} \rfloor + \frac{5}{3} \\
\mathsf{half}_\mathcal{B}(z) &= \lfloor z \rfloor + \frac{4}{3} \\
\mathsf{bits}_\mathcal{B}(z) &= \begin{cases} 4n & \text{if } \exists n \in \mathbb{N} : n + \frac{1}{3} \le z < n + \frac{2}{3} \\ 4n + 3 & \text{if } \exists n \in \mathbb{N} : n - \frac{1}{3} \le z < n + \frac{1}{3} \end{cases} ,
\end{aligned}
$$

where $z \in \mathbb{R}_0^+$ and $n \in \mathbb{N}$. We can easily see that $\mathsf{0}_\mathcal{B}$, $\mathsf{s}_\mathcal{B}$, and $\mathsf{half}_\mathcal{B}$ are monotone with respect to $>_1$. For $\mathsf{bits}_\mathcal{B}$, it is a little bit more complicated. We have to show that $\mathsf{bits}_\mathcal{B}(z') - \mathsf{bits}_\mathcal{B}(z) \ge 1$ whenever $z' - z \ge 1$. If $n' - \frac{1}{3} \le z' < n' + \frac{1}{3}$ for some n' and $n - \frac{1}{3} \le z < n + \frac{1}{3}$ for some n ($n' - n \ge 1$, otherwise $z' - z \ge 1$ would not hold), we have monotonicity because

$$\mathsf{bits}_\mathcal{B}(z') - \mathsf{bits}_\mathcal{B}(z) = 4n' + 3 - (4n + 3) \ge 4n + 7 - (4n + 3) = 4 \ .$$

Similarly, if $n' + \frac{1}{3} \le z' < n' + \frac{2}{3}$ for some n' and $n + \frac{1}{3} \le z < n + \frac{2}{3}$ for some n ($n' - n \ge 1$, otherwise $z' - z \ge 1$ would not hold). Then we can conclude monotonicity because of

$$\mathsf{bits}_\mathcal{B}(z') - \mathsf{bits}_\mathcal{B}(z) = 4n' - 4n \ge 4n + 4 - 4n = 4 \ .$$

If $n' + \frac{1}{3} \le z' < n' + \frac{2}{3}$ for some n' and $n - \frac{1}{3} \le z < n + \frac{1}{3}$ for some n ($n' - n \ge 1$, otherwise $z' - z \ge 1$ would not hold), then

$$\mathsf{bits}_\mathcal{B}(z') - \mathsf{bits}_\mathcal{B}(z) = 4n' - (4n + 3) \ge 4n + 4 - (4n + 3) = 1 \ .$$

If $n - \frac{1}{3} \le z < n + \frac{1}{3}$ for some n' and $n + \frac{1}{3} \le z < n + \frac{2}{3}$ for some n ($n' - n \ge 2$, otherwise $z' - z \ge 1$ would not hold), then

$$\mathsf{bits}_\mathcal{B}(z') - \mathsf{bits}_\mathcal{B}(z) = 4n' + 3 - 4n \ge 4n + 11 - 4n = 11 \ .$$

In order to see that $\mathcal{B}$ is compatible with $\mathcal{R}$ with respect to the order $>_1$, we have to check that the following inequalities hold for all assignments α:

$$
\begin{aligned}
[\alpha]_\mathcal{B}(\mathsf{half}(\mathsf{0})) - [\alpha]_\mathcal{B}(\mathsf{0}) &\ge 1 \\
[\alpha]_\mathcal{B}(\mathsf{half}(\mathsf{s}(\mathsf{0}))) - [\alpha]_\mathcal{B}(\mathsf{0}) &\ge 1 \\
[\alpha]_\mathcal{B}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) - [\alpha]_\mathcal{B}(\mathsf{s}(\mathsf{half}(x))) &\ge 1 \\
[\alpha]_\mathcal{B}(\mathsf{bits}(\mathsf{0})) - [\alpha]_\mathcal{B}(\mathsf{0}) &\ge 1 \\
[\alpha]_\mathcal{B}(\mathsf{bits}(\mathsf{s}(x))) - [\alpha]_\mathcal{B}(\mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x))))) &\ge 1
\end{aligned}
$$

Expanding the first inequality yields

$$[\alpha]_\mathcal{B}(\mathsf{half}(\mathsf{0})) - [\alpha]_\mathcal{B}(\mathsf{0}) = \frac{4}{3} - 0 = \frac{4}{3} \ .$$

In a similar fashion, we get

$$[\alpha]_{\mathcal{B}}(\mathsf{half}(\mathsf{s}(\mathsf{0}))) - [\alpha]_{\mathcal{B}}(\mathsf{0}) = \frac{7}{3} - 0 = \frac{7}{3}$$

for the second inequality. Now consider the third inequality. Let $n \in \mathbb{N}$ be such that $n - \frac{1}{3} \leq \alpha(x) < n + \frac{2}{3}$. Then we have

$$[\alpha]_{\mathcal{B}}(\mathsf{half}(\mathsf{s}(\mathsf{s}(x)))) - [\alpha]_{\mathcal{B}}(\mathsf{s}(\mathsf{half}(x))) \geq n + \frac{13}{3} - (n + \frac{8}{3}) = \frac{5}{3} \ .$$

The fourth inequality can be treated in a similar way as the first two, resulting in

$$[\alpha]_{\mathcal{B}}(\mathsf{bits}(\mathsf{0})) - [\alpha]_{\mathcal{B}}(\mathsf{0}) = 3 - 0 = 3 \ .$$

For the last inequality, let $n \in \mathbb{N}$ be such that $n - \frac{1}{3} \leq \alpha(x) < n + \frac{2}{3}$ again. This yields

$$[\alpha]_{\mathcal{B}}(\mathsf{bits}(\mathsf{s}(x))) - [\alpha]_{\mathcal{B}}(\mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x))))) = 4n + 11 - (4n + 9 + \frac{2}{3}) = \frac{4}{3} \ .$$

This concludes that $\mathcal{B}$ is compatible with $\mathcal{R}$ with respect to the order $>_1$. As in the previous example, we can now apply Lemma 3.9 to construct a Δ-monotone context-dependent interpretation into the nonnegative real numbers which is Δ-compatible with $\mathcal{R}$. This proves termination of $\mathcal{R}$ and the following upper bounds on the derivation height of all ground terms t:

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in D} \frac{[\alpha']_{\mathcal{C}}[\Delta](t)}{\Delta} \qquad \mathrm{dh}_{\mathcal{R}}(t) \leq [\alpha]_{\mathcal{B}}(t)$$

We want to remark that it is essential in this example that the interpretation functions may not be weakly monotone if we want to construct a context-dependent interpretation by applying Lemma 3.9.

Lemma 3.11. *There exists no interpretation into an algebra $\mathcal{A}$ with the carrier $\mathbb{R}_0^+$ which is weakly monotone with respect to the standard order $\geq$ on the real numbers, monotone with respect to the order $>_1$, and compatible with $\mathcal{R}$ with respect to the order $>_1$.*

Proof. Since $\mathcal{A}$ uses the carrier $\mathbb{R}_0^+$, we know that $\mathsf{0}_{\mathcal{A}} \geq 0$, $\mathsf{s}_{\mathcal{A}}(0) \geq 0$, $\mathsf{half}_{\mathcal{A}}(0) \geq 0$, and $\mathsf{bits}_{\mathcal{A}}(0) \geq 0$. We also know that $\mathcal{A}$ must be monotone with respect to the order $>_1$, therefore $\mathsf{s}_{\mathcal{A}}(n) \geq n$, $\mathsf{half}_{\mathcal{A}}(n) \geq n$, and $\mathsf{bits}_{\mathcal{A}}(n) \geq n$ for all $n \in \mathbb{N}$. Furthermore, $\mathcal{A}$ is weakly monotone with respect to the standard order $\geq$ on the real numbers, therefore we also know that $\mathsf{s}_{\mathcal{A}}(z) \geq n$, $\mathsf{half}_{\mathcal{A}}(z) \geq n$, and $\mathsf{bits}_{\mathcal{A}}(n) \geq n$ for all $z = n + y$ with $n \in \mathbb{N}$ and $0 \leq y < 1$. Because of $y < 1$, we can also write this as $\mathsf{s}_{\mathcal{A}}(z) > z - 1$, $\mathsf{half}_{\mathcal{A}}(z) > z - 1$, and $\mathsf{bits}_{\mathcal{A}}(n) > z - 1$.

Claim: For all terms r, t with $[\alpha]_{\mathcal{A}}(t) = m + a$, $m \in \mathbb{N}$, and $0 \leq a < 1$, if t is a subterm of r, then $[\alpha]_{\mathcal{A}}(r) \geq m$.

We prove this claim by structural induction on r. For the base case, we assume that $r = t$. Then, trivially $[\alpha]_{\mathcal{A}}(r') = m + a \geq m$. For the step case, we assume that $r = \mathsf{s}(r')$, $r = \mathsf{half}(r')$ or $r = \mathsf{bits}(r')$, where t is a subterm of r. By induction hypothesis, $[\alpha]_{\mathcal{A}}(r') \geq m$. If $r = \mathsf{s}(r')$, then together with weak monotonicity and $\mathsf{s}_{\mathcal{A}}(z) \geq n$ for $z = n + y$, $n \in \mathbb{N}$, and $0 \leq y < 1$, this implies that $[\alpha]_{\mathcal{A}}(r) \geq m$. For the cases $r = \mathsf{half}(r')$ and $r = \mathsf{bits}(r')$, $[\alpha]_{\mathcal{A}}(r) \geq m$ follows by analogy, which concludes the induction and thus the claim.

Since $\mathcal{A}$ has to be compatible with $\mathcal{R}$ with respect to the order $>_1$, the rewrite rule $\mathsf{half}(\mathsf{0}) \to \mathsf{0}$ implies that $\mathsf{half}_{\mathcal{A}}(\mathsf{0}_{\mathcal{A}}) \geq \mathsf{0}_{\mathcal{A}} + 1$. We know that $\mathcal{A}$ is weakly monotone with respect to the standard order $\geq$ on the real numbers and monotone with respect to the order $>_1$, therefore $\mathsf{half}_{\mathcal{A}}(\mathsf{0}_{\mathcal{A}} + z) \geq \mathsf{0}_{\mathcal{A}} + n + 1$ whenever $z = n + y$ with $n \in \mathbb{N}$ and $0 \leq y < 1$. Again we have the constraint $y < 1$, so this can be written as $\mathsf{half}_{\mathcal{A}}(z') > z'$ for all $z' \geq \mathsf{0}_{\mathcal{A}}$. Because of the rewrite rule $\mathsf{bits}(\mathsf{0}) \to 0$, we can argue in a similar fashion that $\mathsf{bits}_{\mathcal{A}}(z') > z'$ for all $z' \geq \mathsf{0}_{\mathcal{A}}$.

Let $\mathsf{0}_{\mathcal{A}} = n_0 + x_0$, where $n_0 \in \mathbb{N}$ and $0 \leq x_0 < 1$. Since $[\alpha]_{\mathcal{A}}(\mathsf{half}(\mathsf{0})) \geq n_0 + x_0 + 1 \geq n_0 + 1$, we can conclude from our claim that for all terms t which have $\mathsf{half}(\mathsf{0})$ as a subterm, $[\alpha]_{\mathcal{A}}(t) \geq n_0 + 1 > \mathsf{0}_{\mathcal{A}}$. Because all terms occurring below contain the subterm $x := \mathsf{half}(\mathsf{0})$, we are allowed to use all inequalities we concluded above.

The inequality $\mathsf{half}_{\mathcal{T}}(z) > z$ implies

$$[\alpha]_{\mathcal{A}}(\mathsf{half}(\mathsf{s}(x))) = \mathsf{half}_{\mathcal{A}}([\alpha]_{\mathcal{A}}(\mathsf{s}(x))) > [\alpha]_{\mathcal{A}}(\mathsf{s}(x)) \ .$$

Together with the fact that $\mathcal{A}$ is weakly monotone with respect to the standard order $\geq$ on the real numbers, this implies that

$$[\alpha]_{\mathcal{A}}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x)))) \geq [\alpha]_{\mathcal{A}}(\mathsf{bits}(\mathsf{s}(x))) \ .$$

Applying the inequality $\mathsf{s}_{\mathcal{A}}(z) > z - 1$ yields

$$\mathsf{s}_{\mathcal{A}}([\alpha]_{\mathcal{A}}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x))))) > [\alpha]_{\mathcal{A}}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x)))) - 1 \geq [\alpha]_{\mathcal{A}}(\mathsf{bits}(\mathsf{s}(x))) - 1 \ .$$

This can be written as

$$[\alpha]_{\mathcal{A}}(\mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x))))) > [\alpha]_{\mathcal{A}}(\mathsf{bits}(\mathsf{s}(x))) - 1$$

or

$$[\alpha]_{\mathcal{A}}(\mathsf{bits}(\mathsf{s}(x))) - [\alpha]_{\mathcal{A}}(\mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x))))) < 1 \ ,$$

which is a contradiction to the requirement that $\mathcal{A}$ is compatible with $\mathcal{R}$ with respect to the order $>_1$. □

The upper bound on the derivation length induced by this interpretation looks similar to the previous example. We can see that it will be exponential in the number of occurrences of “bits” this time. However, the typical application of this rewrite system is to start with a constructor term, i.e. a term of the form $\mathsf{bits}(\mathsf{s}^n(\mathsf{0}))$ or $\mathsf{bits}(\mathsf{s}^n(\mathsf{0}))$, so we have better (linear) bounds for these average cases:

$$[\alpha]_{\mathcal{B}}(\mathsf{half}(\mathsf{s}^n(\mathsf{0}))) = 2n + \frac{1}{3}$$
$$[\alpha]_{\mathcal{B}}(\mathsf{bits}(\mathsf{s}^n(\mathsf{0}))) = 8n + 3$$

4 Automating the Search for Context-Dependent Interpretations

4.1 Hofbauer's Heuristic

One approach towards finding context-dependent interpretations automatically is a heuristic that was already mentioned in Hofbauer's paper [9]. In order to explain Hofbauer's heuristic, we will step through the construction of a context-dependent interpretation for an example that was already given in [9]. It is also a problem in the TPDB [17], namely the problem with the id `TRS/SK90-2.50`. Consider the TRS $\mathcal{R}$ consisting of the single rewrite rule

$$\mathsf{a}(\mathsf{b}(x)) \to \mathsf{b}(\mathsf{b}(\mathsf{a}(x)))$$

over the signature containing the unary function symbols a and b, and the constant function symbol c. First, a polynomial interpretation into a well-founded monotone algebra $\mathcal{B}$ is constructed, which proves termination of $\mathcal{R}$. For this example, a suitable interpretation into $\mathbb{N}$ is

$$\mathsf{a}_\mathcal{B}(x) = 3x \qquad \mathsf{b}_\mathcal{B}(x) = x + 1 \qquad \mathsf{c}_\mathcal{B} = 0 \ .$$

The most difficult part in searching a context-dependent interpretation over the reals is to find suitable functions $f_\mathcal{A}$. In Hofbauer's heuristic, this point is addressed by arbitrarily choosing a coefficient $k+1$ in the original interpretation and replacing it by $k + \Delta$. For this example, the best choice is to replace the "3" in $\mathsf{a}_\mathcal{B}$ by "$2 + \Delta$". This yields the following functions $f_\mathcal{A}$:

$$\mathsf{a}_\mathcal{A}[\Delta](x) = (2 + \Delta)x \qquad \mathsf{b}_\mathcal{A}[\Delta](x) = x + 1 \qquad \mathsf{c}_\mathcal{A}[\Delta] = 0$$

The other part of the interpretation we have to find are the functions $f^i_\mathcal{A}$. Once we have the functions $f_\mathcal{A}$, they can be constructed by applying the Δ-monotonicity constraints. For the first (and only) argument position of the function symbol a, we have

$$\mathsf{a}_\mathcal{A}[\Delta](x + \mathsf{a}^1_\mathcal{A}(\Delta)) - \mathsf{a}_\mathcal{A}[\Delta](x) \geq \Delta \ .$$

Expanding this inequality and solving it in $\mathsf{a}^1_\mathcal{A}(\Delta)$ yields

$$\begin{aligned} (2 + \Delta)(x + \mathsf{a}^1_\mathcal{A}(\Delta)) - (2 + \Delta)x &\geq \Delta \\ \mathsf{a}^1_\mathcal{A}(\Delta) &\geq \frac{\Delta}{2 + \Delta} \ . \end{aligned}$$

Similarly, for $\mathsf{b}^1_\mathcal{A}$ we obtain

$$\mathsf{b}^1_\mathcal{A}(\Delta) \geq \Delta \ .$$

In order to get a good bound on the derivation height, we want to keep the interpretation as small as possible. Therefore, we choose

$$\mathsf{a}^1_{\mathcal{A}}(\Delta) = \frac{\Delta}{1+\Delta} \qquad \mathsf{b}^1_{\mathcal{A}}(\Delta) = \Delta \ .$$

This completes the definition of the context-dependent interpretation. It is by construction Δ-monotone. All that is left to do in order apply Theorem 3.8 is to check whether Δ-compatibility holds. Indeed,

$$\begin{aligned}
&[\alpha]_{\mathcal{A}}[\Delta](\mathsf{a}(\mathsf{b}(x))) - [\alpha]_{\mathcal{A}}[\Delta](\mathsf{b}(\mathsf{b}(\mathsf{a}(x))))\\
&\quad = \mathsf{a}_{\mathcal{A}}[\Delta]([\alpha]_{\mathcal{A}}[\mathsf{a}^1_{\mathcal{A}}(\Delta)](\mathsf{b}(x))) - \mathsf{b}_{\mathcal{A}}[\Delta]([\alpha]_{\mathcal{A}}[\mathsf{b}^1_{\mathcal{A}}(\Delta)](\mathsf{b}(\mathsf{a}(x))))\\
&\quad = (2+\Delta)\mathsf{b}_{\mathcal{A}}[\frac{\Delta}{2+\Delta}](x) - (\mathsf{b}_{\mathcal{A}}[\Delta]([\alpha]_{\mathcal{A}}[\Delta]\mathsf{a}(x)) + 1)\\
&\quad = (2+\Delta)(\alpha[\frac{\Delta}{2+\Delta}](x) + 1) - (\mathsf{a}_{\mathcal{A}}[\Delta]([\alpha]_{\mathcal{A}}[\mathsf{a}^1_{\mathcal{A}}](x)) + 2)\\
&\quad = (2+\Delta)(\alpha[\frac{\Delta}{2+\Delta}](x) + 1) - ((2+\Delta)(\alpha[\frac{\Delta}{2+\Delta}](x)) + 2)\\
&\quad = \Delta \ .
\end{aligned}$$

For a term t of the shape $\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))$, the polynomial interpretation yields the bound

$$\mathrm{dh}_{\mathcal{R}}(t) \leq 3^n \cdot m$$

on the derivation height. For the context dependent interpretation, we have

$$[\alpha]_{\mathcal{A}}[\Delta](\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))) = (2^n + (2^n - 1)\Delta)m \ ,$$

which can be proved by induction on n. For the base case, we obviously have $[\alpha]_{\mathcal{A}}[\Delta](\mathsf{b}^m(\mathsf{c})) = m$. For the induction step, we have

$$\begin{aligned}
[\alpha]_{\mathcal{A}}[\Delta](\mathsf{a}(\mathsf{a}^n(\mathsf{b}^m(\mathsf{c})))) &= (2+\Delta)[\alpha]_{\mathcal{A}}[\frac{\Delta}{2+\Delta}](\mathsf{a}^n(\mathsf{b}^m(\mathsf{c})))\\
&= (2+\Delta)(2^n + (2^n-1)\frac{\Delta}{2+\Delta})m\\
&= (2^{n+1} + 2^n\Delta + 2^n\Delta - \Delta)m\\
&= (2^{n+1} + (2^{n+1}-1)\Delta)m \ ,
\end{aligned}$$

where the second line follows from the induction hypothesis. Finally, by Theorem 3.8, we have the following bound on the derivation height:

$$\mathrm{dh}_{\mathcal{R}}(\mathsf{a}^n(\mathsf{b}^m(\mathsf{c}))) \leq \inf_{\Delta\in\mathbb{R}^+} \frac{(2^n + (2^n-1)\Delta)m}{\Delta} = (2^n - 1)m \ ,$$

which is in $\mathcal{O}(2^n)$ and therefore better than the bound from the polynomial interpretation, which is in $\mathcal{O}(3^n)$.

The heuristic we have just seen can be defined by the algorithm in Table 4.1:

Table 4.1: An algorithm to automate the application of Hofbauer's heuristic

1. Find a polynomial interpretation into a well-founded monotone algebra $\mathcal{A}$ such that all interpretation functions $f_{\mathcal{A}}$ are weakly monotone with respect to the standard order $\geq$ on $\mathbb{R}_0^+$ and $\mathcal{A}$ is compatible with $\mathcal{R}$.
2. Choose an arbitrary set of coefficients with positive values from the polynomials in the functions $f_{\mathcal{A}}$. We call this set of coefficients C.
3. For each function symbol f, construct the function $f_{\mathcal{B}}[\Delta](x_1, \ldots, x_n)$ as follows: take the function $f_{\mathcal{A}}(x_1, \ldots, x_n)$, and for each coefficient c such that $c \in C$, change its value to $k - 1 + \Delta$, where k is the original value of c.
4. For each function symbol f and each $i \in \{1, \ldots, n\}$ with $n = \text{arity}(f)$, consider the Δ-monotonicity constraint

   $$f_{\tau}[\Delta](x_1, \ldots, x_i + f_{\tau}^i(\Delta), \ldots, x_n) - f_{\tau}[\Delta](x_1, \ldots, x_i, \ldots, x_n) \geq \Delta \ .$$

   Expand this constraint and solve it in $f_{\tau}^i(\Delta)$. Then for each Δ, define $f_{\tau}^i(\Delta)$ such that this constraint is fulfilled.
5. Check whether Δ-compatibility holds for the constructed interpretation. If it holds, then the interpretation is a result of the algorithm. Otherwise, discard the interpretation.

4.2 Preliminaries for the adapted Algorithm of Contejean et al.

Although Hofbauer's heuristic works very well for some examples, experiments on the TPDB [17] have shown that the number of such examples is rather low. Therefore, we have changed our strategy from modifying an existing polynomial interpretation towards generating context-dependent interpretations directly. For polynomial interpretations, the algorithm of Contejean et al. [5] can be used to find suitable interpretations automatically. As described in this chapter, this algorithm can be adapted for context-dependent interpretations over the reals.

In order to make computations on the polynomials easier, we transform them into a certain normal form. We use the same notation of polynomials as in [5].

Definition 4.1. We define polynomials in normal form as follows:

1. A *literal* is an expression of the form x^n, where x is a variable, and n is a natural number.
2. A *monomial* is an expression of the form $c \cdot \prod_i v_i$, where c is an integer and v_i is a literal for all i. We call c the *coefficient* of the monomial.
3. A (standard) polynomial is in *normal form* if it has the form $\sum_i m_i$ for some monomials m_i.

By applying associativity, commutativity, and distributivity of addition and multiplication, every polynomial can be transformed into this normal form.

For context-dependent interpretations over the reals, we will also consider Δ-*quotients*. This gives rise to the following definition of an *extended polynomials in normal form*.

Definition 4.2. A Δ-*quotient* is an expression of the form

$$\frac{\Delta}{p_1 + p_2\Delta} \ ,$$

where Δ is a parameter ranging over the positive real numbers, and p_1, p_2 are polynomials. All coefficients of p_1 and p_2 must be nonnegative, and either $p_1 > 0$ or $p_2 > 0$ must hold. We say that the Δ-quotient is in normal form if p_1 and p_2 are in normal form.

Definition 4.3. Extended polynomials in normal form are built up as follows:

1. An *extended monomial* is an expression of the form $c \cdot \prod_i v_i$, where c is an integer, and v_i is either a literal or a Δ-quotient for all i. We call c the *coefficient* of the extended monomial. We say that the monomial is in normal form if all v_i which are Δ-quotients are in normal form.
2. An *extended polynomial in normal form* is an expression of the form $\sum_i m_i$ for some extended monomials m_i which are in normal form.

Definition 4.4. A literal *occurs positively (negatively)* in an extended polynomial P in normal form if P contains an extended monomial which contains the literal and whose coefficient is positive (negative).

A useful property of Δ-quotients is the fact that substituting a Δ-quotient d_2 for the Δ in a Δ-quotient d_1 yields another Δ-quotient.

Lemma 4.5. *Let $d_1 = \frac{\Delta}{p_1+p_2\Delta}$ and $d_2 = \frac{\Delta}{q_1+q_2\Delta}$ be Δ-quotients. Then the result of substituting d_2 for Δ in d_1 is equivalent to a Δ-quotient.*

Proof.

$$\begin{aligned} d_1[\Delta := d_2] &= \frac{d_2}{p_1 + p_2 d_2} \\ &= \frac{\frac{\Delta}{q_1+q_2\Delta}}{p_1 + p_2\frac{\Delta}{q_1+q_2\Delta}} \\ &= \frac{\frac{\Delta}{q_1+q_2\Delta}}{p_1\frac{q_1+q_2\Delta}{q_1+q_2\Delta} + p_2\frac{\Delta}{q_1+q_2\Delta}} \\ &= \frac{\Delta}{p_1(q_1 + q_2\Delta) + p_2\Delta} \\ &= \frac{\Delta}{p_1 q_1 + (p_1 q_2 + p_2)\Delta} \end{aligned}$$

All coefficients of p_1, p_2, q_1, and q_2 are nonnegative, therefore the coefficients of $p_1 q_1$ and $p_1 q_2 + p_2$ are nonnegative, as well. Furthermore, we have either $p_1 > 0$ or $p_2 > 0$, and we have either $q_1 > 0$ or $q_2 > 0$. Thus, either $p_1 q_1 > 0$ or $p_1 q_2 + p_2 > 0$ must hold. Since polynomials are closed under addition and multiplication, this shows that the result is indeed equivalent to a Δ-quotient. □

Definition 4.6. A *context-dependent interpretation with Δ-simple polynomials* is defined to be an interpretation into a context-dependent algebra $\mathcal{A} = (\mathbb{R}_0^+, \mathbb{R}^+, [\cdot]_\mathcal{A})$ such that for all function symbols f of arity n, we have

$$f_\mathcal{A}[\Delta](x_1, \dots, x_n) = d_f + e_f\Delta + \sum_{i=1}^{n} a_{f,i}x_i + \sum_{i=1}^{n} b_{f,i}x_i\Delta$$
$$f_\mathcal{A}^i(\Delta) = \frac{\Delta}{a_{f,i} + b_{f,i}\Delta}$$

for some natural numbers $d_f, e_f, a_{f,i}, b_{f,i}$ for all $f \in \mathcal{F}$ and $i \in \{1, \dots, n\}$. If the variables are not instantiated, but kept as variables, then we call such a Δ-simple interpretation for a TRS $\mathcal{R}$ the *parametric context-dependent with Δ-simple polynomials* with respect to $\mathcal{R}$.

We define the set of *coefficient variables* of the parametric context-dependent interpretation with Δ-simple polynomials with respect to a TRS $\mathcal{R}$ as

$$\mathcal{CV}_{\mathcal{DS}}(\mathcal{R}) = \{a_{f,i}, b_{f,i}, e_f, d_f \mid f \in \mathcal{F}, i \in \{1, \dots, n\}\}$$

where $n = \text{arity(f)}$.

Note that the newly introduced parameters in the $f_\mathcal{A}$ functions are all fresh variables, but each $f_\mathcal{A}^i$ function reuses two of the new parameters from the respective $f_\mathcal{A}$ function.

Definition 4.7. A *context-dependent interpretation with restricted Δ-simple polynomials* is defined to be an interpretation into a context-dependent algebra $\mathcal{A} = (\mathbb{R}_0^+, \mathbb{R}^+, [\cdot]_\mathcal{A})$ such that for all function symbols f of arity n, we have

$$f_\mathcal{A}[\Delta](x_1, \dots, x_n) = d_f + e_f\Delta + \sum_{i=1}^{n} a_{f,i}x_i + \sum_{i=1}^{n} b_{f,i}x_i\Delta$$
$$f_\mathcal{A}^i(\Delta) = \frac{\Delta}{a_{f,i} + b_{f,i}\Delta}$$

for some natural numbers $d_f, e_f, a_{f,i}, b_{f,i}$ for all $f \in \mathcal{F}$ and $i \in \{1, \dots, n\}$ with $a_{f,i} \in \{0, 1\}$ for all $i \in \{1, \dots, n\}$. The parametric context-dependent interpretation with restricted Δ-simple polynomials and its set of coefficient variables $\mathcal{CV}_{\mathcal{DSR}}(\mathcal{R})$ are defined in the same way as for context-dependent interpretations with Δ-simple polynomials.

From theses definitions, it follows that every context-dependent interpretation with restricted Δ-simple polynomials is also a context-dependent interpretation with Δ-simple polynomials. Furthermore, we have $\mathcal{CV}_{\mathcal{DS}}(\mathcal{R}) = \mathcal{CV}_{\mathcal{DSR}}(\mathcal{R})$.

Lemma 4.8. *For every context-dependent interpretation with Δ-simple polynomials, Δ-monotonicity holds.*

Proof. For every function symbol $f \in \mathcal{F}$ and every $i \in \{1, \ldots, n\}$ with $n = \text{arity}(f)$, we have

$$\begin{aligned} f_{\mathcal{A}}[\Delta](x_1, \ldots, x_i + f^i_{\mathcal{A}}(\Delta), \ldots, x_n) - f_{\mathcal{A}}[\Delta](x_1, \ldots, x_n) \\ = a_{f,i} \cdot f^i_{\mathcal{A}}(\Delta) + b_{f,i} \cdot \Delta \cdot f^i_{\mathcal{A}}(\Delta) \\ = \frac{\Delta}{a_{f,i} + b_{f,i}\Delta} \cdot (a_{f,i} + b_{f,i}\Delta) \\ = \Delta \ . \end{aligned}$$

Since context-dependent interpretations with Δ-simple polynomials are obviously weakly monotone with respect to the standard order $\geq$ on $\mathbb{R}^+_0$, this is enough to conclude Δ-monotonicity. $\square$

Lemma 4.9. *Every context-dependent interpretation with Δ-simple polynomials into a context-dependent algebra $\mathcal{A} = (\mathbb{R}^+_0, \mathbb{R}^+, [\cdot]_{\mathcal{A}})$ equipped with the orders $\{>_\Delta \mid \Delta \in \mathbb{R}^+\}$ from Definition 3.7 is a context-dependent interpretation over the reals.*

Proof. From Lemma 4.8 and the fact that for all Δ, the order $>_\Delta$ is well-founded on $\mathbb{R}^+_0$, it follows that every context-dependent interpretation with Δ-simple polynomials is an interpretation into a well-founded Δ-monotone algebra. Since $A = \mathbb{R}^+_0$, $D = \mathbb{R}^+$, and for all $\Delta \in \mathbb{R}^+$, we have

$$a >_\Delta b \iff a - b \geq \Delta \ ,$$

this concludes the lemma. $\square$

Lemma 4.10. *In each evaluation step of a context-dependent interpretation with Δ-simple polynomials, the parameter of the context-dependent interpretation is representable as a Δ-quotient if for all functions f^i_τ in the interpretation, we have that the denominator of f^i_τ is positive.*

Proof. We prove this by induction on the position where the evaluation step takes place. If the evaluation step takes place at the root position of the considered term, then the parameter is just

$$\Delta = \frac{\Delta}{1 + 0 \cdot \Delta} \ .$$

If we are looking at an evaluation step below the root of the considered term, then the current parameter has the shape $f^i_\tau(\Delta_1)$ for some f, i, and Δ_1. By induction hypothesis, we know that Δ_1 is equivalent to a Δ-quotient. By Definition 4.6, we have

$$f^i_\tau(\Delta_1) = \frac{\Delta_1}{a_{f,i} + b_{f,i}\Delta_1} \ .$$

We know that $a_{f,i}$ and $b_{f,i}$ are nonnegative polynomials. Furthermore, if the denominator of f^i_τ is positive, then either $a_{f,i} > 0$ or $b_{f,i} > 0$. Therefore, the function f^i_τ has the shape of a Δ-quotient. Now it follows from Lemma 4.5 that $f^i_\tau(\Delta_1)$ must be equivalent to a Δ-quotient, as well. $\square$

4.3 The adapted Algorithm of Contejean et al.

As a first step in the algorithm, *parametric polynomials* are created for all interpretation functions which are needed for the given signature. For context-dependent interpretations, we have found context-dependent interpretations with (restricted) Δ-simple polynomials to be particularly useful. The restricted variant induces a nice upper bound on the derivational complexity of a rewrite system. For the remainder of this chapter, we will abbreviate them by Δ*-simple interpretations* and *restricted* Δ*-simple interpretations*, respectively.

As a running example for this section, we consider the TRS from Table 2.1 again. We have the parametric Δ-simple interpretation into the well-founded Δ-monotone algebra $\mathcal{A} = (\mathbb{R}_0^+, \mathbb{R}^+, [\cdot]_\mathcal{A}, \{>_\Delta \mid \Delta \in \mathbb{R}^+\})$ for $\mathcal{R}$, where $[\cdot]_\mathcal{A}$ is defined by the following functions:

$$\mathsf{a}_\mathcal{A}[\Delta](x) = d_\mathsf{a} + e_\mathsf{a}\Delta + a_{\mathsf{a},1}x + b_{\mathsf{a},1}x\Delta \qquad \mathsf{a}^1_\mathcal{A}(\Delta) = \frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta}$$

$$\mathsf{b}_\mathcal{A}[\Delta](x) = d_\mathsf{b} + e_\mathsf{b}\Delta + a_{\mathsf{b},1}x + b_{\mathsf{b},1}x\Delta \qquad \mathsf{b}^1_\mathcal{A}(\Delta) = \frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}$$

The set of coefficient variables in our example is

$$\mathcal{CV}_{\mathcal{DS}}(\mathcal{R}) = \{d_\mathsf{a}, e_\mathsf{a}, a_{\mathsf{a},1}, b_{\mathsf{a},1}, d_\mathsf{b}, e_\mathsf{b}, a_{\mathsf{b},1}, b_{\mathsf{b},1}\}$$

The parametric Δ-simple interpretation is applied to both sides of each rewrite rule of the TRS $\mathcal{R}$ we consider. For each rewrite rule $l \to r$ in $\mathcal{R}$, we construct a compatibility constraint for that rule in this way:

Definition 4.11. Given a TRS $\mathcal{R}$ and a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$, the *rule constraints* of $\mathcal{R}$ with respect to $\mathcal{A}$ are defined as follows:

$$\mathrm{rc}_\mathcal{A}(\mathcal{R}) = \{[\alpha]_\mathcal{A}[\Delta](l) - [\alpha]_\mathcal{A}[\Delta](r) - \Delta \geq 0 \mid l \to r \in \mathcal{R}\}$$

The following lemma follows immediately from this definition.

Lemma 4.12. *Let $\mathcal{R}$ be a TRS and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. If all constraints in $\mathrm{rc}_\mathcal{A}(\mathcal{R})$ are valid, then $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$.*

In our running example, we have the following rule constraint for the parametric Δ-simple interpretation of $\mathcal{R}$:

$$\begin{aligned}
\mathrm{rc}_\mathcal{A}(\mathcal{R}) &= \{[\alpha]_\mathcal{A}[\Delta](\mathsf{a}(\mathsf{b}(x))) - [\alpha]_\mathcal{A}[\Delta](\mathsf{b}(\mathsf{a}(x))) - \Delta \geq 0\} \\
&= \{d_\mathsf{a} + e_\mathsf{a}\Delta + (a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta)[\alpha]_\mathcal{A}[\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta}](\mathsf{b}(x)) \\
&\qquad - d_\mathsf{b} - e_\mathsf{b}\Delta - (a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta)[\alpha]_\mathcal{A}[\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}](\mathsf{a}(x)) - \Delta \geq 0\}
\end{aligned}$$

$$
\begin{aligned}
&= \{d_{\mathsf{a}} + e_{\mathsf{a}}\Delta + (a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta)(d_{\mathsf{b}} + e_{\mathsf{b}}\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta} \\
&\quad + (a_{\mathsf{b},1} + b_{\mathsf{b},1}\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta})\alpha[\frac{\Delta}{a_{\mathsf{a},1}a_{\mathsf{b},1} + (b_{\mathsf{a},1}a_{\mathsf{b},1} + b_{\mathsf{b},1})\Delta}](x)) \\
&\quad - d_{\mathsf{b}} - e_{\mathsf{b}}\Delta - (a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta)(d_{\mathsf{a}} + e_{\mathsf{a}}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} \\
&\quad + (a_{\mathsf{a},1} + b_{\mathsf{a},1}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta})\alpha[\frac{\Delta}{a_{\mathsf{a},1}a_{\mathsf{b},1} + (a_{\mathsf{a},1}b_{\mathsf{b},1} + b_{\mathsf{a},1})\Delta}](x)) \\
&\quad - \Delta \geq 0\}
\end{aligned}
$$

It follows from the definition of Δ-simple interpretations that all coefficients occurring in the denominator must be nonnegative. Hence, in order to ensure that they are not zero, we only have to ensure that the following constraints hold:

Definition 4.13. Given a TRS $\mathcal{R}$ and a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$, the *nonzero constraints* of $\mathcal{R}$ with respect to $\mathcal{A}$ are defined as follows:

$$\mathrm{nz}_{\mathcal{A}}(\mathcal{R}) = \{a_{f,i} + b_{f,i} - 1 \geq 0 \mid f \in \mathcal{F}, n = \mathrm{arity}(f), i \in \{1, \ldots, n\}\}$$

In the running example, we have two nonzero constraints:

$$\mathrm{nz}_{\mathcal{A}}(\mathcal{R}) = \{a_{\mathsf{a},1} + b_{\mathsf{a},1} - 1 \geq 0,\ a_{\mathsf{b},1} + b_{\mathsf{b},1} - 1 \geq 0\}$$

We have to solve the constraints $\mathrm{rc}_{\mathcal{A}}(\mathcal{R})$ in the variables from $\mathcal{CV}_{\mathcal{DS}}(\mathcal{R})$. Because of the way we defined the functions $f^i_{\mathcal{A}}$, these constraints are in general not standard polynomials. By Lemma 4.10, each parameter in the definition of Δ-simple interpretations is representable as a Δ-quotient. In the interpretation functions $f_{\mathcal{A}}$, the Δ appears only in place of variables. Therefore the constraints are equivalent to extended polynomials in normal form. We transform the extended polynomials in normal form into standard polynomials in normal form.

Lemma 4.14. *Let $\mathcal{R}$ be a TRS and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. Then apply the algorithm in Table 4.2 to all constraints in* $\mathrm{rc}_{\mathcal{A}}(\mathcal{R})$.

If for every Δ-quotient occurring in the constraints, the denominator is positive, then this procedure terminates and the resulting constraints are valid if and only if the constraints in $\mathrm{rc}_{\mathcal{A}}(\mathcal{R})$ *are valid.*

Proof. In every iteration of the procedure, the picked Δ-quotient has by definition the form $\frac{\Delta}{p_1+p_2\Delta}$. By assumption, we have $p_1 + p_2\Delta > 0$. Therefore, in each step, the only action is that a constraint is multiplied with the positive term $p_1 + p_2\Delta$. Hence, for each step, the constraints are valid before the step if and only if they are valid after the step.

All that remains to show now is termination of the procedure. For each monomial m, define dq_m as the multiset of all denominators of Δ-quotients occurring in m. For each constraint c, let the multiset dq_c be

$$\max\nolimits_{\mathrm{mul}}\{\mathrm{dq}_m \mid m \text{ is a monomial occurring in } c\}\ .$$

Table 4.2: An algorithm to eliminate Δ-quotients in the rule constraints

1. As long as there are still Δ-quotients in the constraint, pick an arbitrary one. If no Δ-quotients are left in the constraint, proceed to Step 4.
2. Multiply each monomial on both sides of the constraint with the denominator of the picked Δ-quotient as follows: whenever a monomial contains a Δ-quotient which has the same denominator as the picked Δ-quotient, we use the multiplication to eliminate it. For other monomials, treat all Δ-quotients in that monomial as units and use distributivity on the multiplication in order to bring the constraint back into normal form.
3. Return to Step 1.
4. Return the resulting polynomial, where all Δ-quotients (except for the ones in the parameter) are eliminated, and only a standard polynomial in normal form remains.

Then for every step of the procedure where a Δ-quotient in a constraint c is picked, the number of occurrences of its denominator is reduced by one in every monomial in c where the denominator occurred. Therefore, the number of elements in dq_c is reduced by one. After a finite amount of steps, dq_c will be empty for every constraint c, and thus, no constraints contain any Δ-quotients. At this point, the procedure terminates. $\square$

Definition 4.15. Let $\mathcal{R}$ be a TRS, and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. We call the set of constraints generated from $\mathrm{dc}_{\mathcal{A}}(\mathcal{R})$ by the procedure in Table 4.2 the *rule polynomials* of $\mathcal{R}$ with respect to $\mathcal{A}$. We denote this set by $\mathrm{rp}_{\mathcal{A}}(\mathcal{R})$.

Lemma 4.16. *Let $\mathcal{R}$ be a TRS and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. If all constraints in $\mathrm{rp}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{nz}_{\mathcal{A}}(\mathcal{R})$ are valid, then $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$.*

Proof. Because of Lemma 4.10 and the constraints in $\mathrm{nz}_{\mathcal{A}}(\mathcal{R})$, we know that all denominators of Δ-quotients occurring in $\mathrm{rc}_{\mathcal{A}}(\mathcal{R})$ must be positive. Therefore, we can apply Lemma 4.14 and the validity of the constraints in $\mathrm{rp}_{\mathcal{A}}(\mathcal{R})$ in order to infer validity of the constraints in $\mathrm{rc}_{\mathcal{A}}(\mathcal{R})$. By Lemma 4.12, this implies that $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$. $\square$

In the running example, $\mathrm{rc}_{\mathcal{A}}(\mathcal{R})$ consists of the single constraint

$$\begin{aligned} d_{\mathsf{a}} + e_{\mathsf{a}}\Delta + (a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta)(d_{\mathsf{b}} + e_{\mathsf{b}}\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta} \\ + (a_{\mathsf{b},1} + b_{\mathsf{b},1}\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta})\alpha[\frac{\Delta}{a_{\mathsf{a},1}a_{\mathsf{b},1} + (b_{\mathsf{a},1}a_{\mathsf{b},1} + b_{\mathsf{b},1})\Delta}](x)) \end{aligned}$$

$$
\begin{aligned}
&- d_{\mathsf{b}} - e_{\mathsf{b}}\Delta - (a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta)(d_{\mathsf{a}} + e_{\mathsf{a}}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} \\
&\quad + (a_{\mathsf{a},1} + b_{\mathsf{a},1}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta})\alpha[\frac{\Delta}{a_{\mathsf{a},1}a_{\mathsf{b},1} + (a_{\mathsf{a},1}b_{\mathsf{b},1} + b_{\mathsf{a},1})\Delta}](x)) - \Delta \geq 0 \ .
\end{aligned}
$$

In the following, we will abbreviate the Δ-quotients $\frac{\Delta}{a_{\mathsf{a},1}a_{\mathsf{b},1}+(b_{\mathsf{a},1}a_{\mathsf{b},1}+b_{\mathsf{b},1})\Delta}$ and $\frac{\Delta}{a_{\mathsf{a},1}a_{\mathsf{b},1}+(a_{\mathsf{a},1}b_{\mathsf{b},1}+b_{\mathsf{a},1})\Delta}$ by Δ_1 and Δ_2, respectively. By applying this abbreviation and bringing the extended polynomial on the left hand side into normal form, we get

$$
\begin{aligned}
&d_{\mathsf{a}} + e_{\mathsf{a}}\Delta + a_{\mathsf{a},1}d_{\mathsf{b}} + a_{\mathsf{a},1}e_{\mathsf{b}}\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta} + a_{\mathsf{a},1}a_{\mathsf{b},1}\alpha[\Delta_1](x) \\
&\quad + a_{\mathsf{a},1}b_{\mathsf{b},1}\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta}\alpha[\Delta_1](x) + b_{\mathsf{a},1}d_{\mathsf{b}}\Delta + b_{\mathsf{a},1}e_{\mathsf{b}}\Delta\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta} \\
&\quad + b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta\alpha[\Delta_1](x) + b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta\frac{\Delta}{a_{\mathsf{a},1} + b_{\mathsf{a},1}\Delta}\alpha[\Delta_1](x) \\
&\quad - d_{\mathsf{b}} - e_{\mathsf{b}}\Delta - d_{\mathsf{a}}a_{\mathsf{b},1} - e_{\mathsf{a}}a_{\mathsf{b},1}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} - a_{\mathsf{a},1}a_{\mathsf{b},1}\alpha[\Delta_2](x) \\
&\quad - b_{\mathsf{a},1}a_{\mathsf{b},1}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}\alpha[\Delta_2](x) - d_{\mathsf{a}}b_{\mathsf{b},1}\Delta - e_{\mathsf{a}}b_{\mathsf{b},1}\Delta\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} \\
&\quad - a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta\alpha[\Delta_2](x) - b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}\alpha[\Delta_2](x) - \Delta \geq 0 \ .
\end{aligned}
$$

Now we pick a Δ-quotient in the last constraint, e.g. $\frac{\Delta}{a_{\mathsf{a},1}+b_{\mathsf{a},1}\Delta}$. Multiplying the constraint with the (positive) denominator of the Δ-quotient yields

$$
\begin{aligned}
&d_{\mathsf{a}}a_{\mathsf{a},1} + d_{\mathsf{a}}b_{\mathsf{a},1}\Delta + e_{\mathsf{a}}a_{\mathsf{a},1}\Delta + e_{\mathsf{a}}b_{\mathsf{a},1}\Delta^2 + a_{\mathsf{a},1}^2d_{\mathsf{b}} + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}\Delta + a_{\mathsf{a},1}e_{\mathsf{b}}\Delta \\
&\quad + a_{\mathsf{a},1}^2a_{\mathsf{b},1}\alpha[\Delta_1](x) + a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta\alpha[\Delta_1](x) + a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta\alpha[\Delta_1](x) \\
&\quad + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}\Delta + b_{\mathsf{a},1}^2d_{\mathsf{b}}\Delta^2 + b_{\mathsf{a},1}e_{\mathsf{b}}\Delta^2 + a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta\alpha[\Delta_1](x) \\
&\quad + b_{\mathsf{a},1}^2a_{\mathsf{b},1}\Delta^2\alpha[\Delta_1](x) + b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_1](x) \\
&\quad - a_{\mathsf{a},1}d_{\mathsf{b}} - b_{\mathsf{a},1}d_{\mathsf{b}}\Delta - a_{\mathsf{a},1}e_{\mathsf{b}}\Delta - b_{\mathsf{a},1}e_{\mathsf{b}}\Delta^2 - d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1} - d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta \\
&\quad - e_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} - e_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} \\
&\quad - a_{\mathsf{a},1}^2a_{\mathsf{b},1}\alpha[\Delta_2](x) - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta\alpha[\Delta_2](x) - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}\alpha[\Delta_2](x) \\
&\quad - b_{\mathsf{a},1}^2a_{\mathsf{b},1}\Delta\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}\alpha[\Delta_2](x) - d_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta - d_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2 \\
&\quad - e_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} - e_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta} - a_{\mathsf{a},1}^2b_{\mathsf{b},1}\Delta\alpha[\Delta_2](x) \\
&\quad - a_{\mathsf{a},1}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_2](x) - a_{\mathsf{a},1}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}\alpha[\Delta_2](x) \\
&\quad - b_{\mathsf{a},1}^2b_{\mathsf{b},1}\Delta^2\frac{\Delta}{a_{\mathsf{b},1} + b_{\mathsf{b},1}\Delta}\alpha[\Delta_2](x) - a_{\mathsf{a},1}\Delta - b_{\mathsf{a},1}\Delta^2 \geq 0 \ .
\end{aligned}
$$

There is only one kind of Δ-quotients left in the constraint, namely $\frac{\Delta}{a_{\mathsf{b},1}+b_{\mathsf{b},1}\Delta}$, so we pick it and and do the multiplication:

$$\begin{aligned}
&d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1} + d_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta + d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta + d_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2 + e_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}\Delta \\
&+ e_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2 + e_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta^2 + e_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^3 + a_{\mathsf{a},1}^2 d_{\mathsf{b}}a_{\mathsf{b},1} + a_{\mathsf{a},1}^2 d_{\mathsf{b}}b_{\mathsf{b},1}\Delta \\
&+ a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1}\Delta + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1}\Delta^2 + a_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1}\Delta + a_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1}\Delta^2 \\
&+ a_{\mathsf{a},1}^2 a_{\mathsf{b},1}^2\alpha[\Delta_1](x) + a_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta\alpha[\Delta_1](x) + a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}^2\Delta\alpha[\Delta_1](x) \\
&+ a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_1](x) + a_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta\alpha[\Delta_1](x) + a_{\mathsf{a},1}b_{\mathsf{b},1}^2\Delta^2\alpha[\Delta_1](x) \\
&+ a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1}\Delta + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1}\Delta^2 + b_{\mathsf{a},1}^2 d_{\mathsf{b}}a_{\mathsf{b},1}\Delta^2 + b_{\mathsf{a},1}^2 d_{\mathsf{b}}b_{\mathsf{b},1}\Delta^3 \\
&+ b_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1}\Delta^2 + b_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1}\Delta^3 + a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}^2\Delta\alpha[\Delta_1](x) \\
&+ a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_1](x) + b_{\mathsf{a},1}^2 a_{\mathsf{b},1}^2\Delta^2\alpha[\Delta_1](x) + b_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^3\alpha[\Delta_1](x) \\
&+ b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_1](x) + b_{\mathsf{a},1}b_{\mathsf{b},1}^2\Delta^3\alpha[\Delta_1](x) \\
&- a_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1} - a_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1}\Delta - b_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1}\Delta - b_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1}\Delta^2 - a_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1}\Delta \\
&- a_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1}\Delta^2 - b_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1}\Delta^2 - b_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1}\Delta^3 - d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}^2 - d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta \\
&- d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}^2\Delta - d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^2 - e_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}\Delta - e_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta^2 \\
&- a_{\mathsf{a},1}^2 a_{\mathsf{b},1}^2\alpha[\Delta_2](x) - a_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta\alpha[\Delta_2](x) - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}^2\Delta\alpha[\Delta_2](x) \\
&- a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_2](x) - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta\alpha[\Delta_2](x) - b_{\mathsf{a},1}^2 a_{\mathsf{b},1}\Delta^2\alpha[\Delta_2](x) \\
&- d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta - d_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1}^2\Delta^2 - d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^2 - d_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1}^2\Delta^3 \\
&- e_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2 - e_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^3 - a_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta\alpha[\Delta_2](x) - a_{\mathsf{a},1}^2 b_{\mathsf{b},1}^2\Delta^2\alpha[\Delta_2](x) \\
&- a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_2](x) - a_{\mathsf{a},1}b_{\mathsf{a},1}b_{\mathsf{b},1}^2\Delta^3\alpha[\Delta_2](x) - a_{\mathsf{a},1}b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2\alpha[\Delta_2](x) \\
&- b_{\mathsf{a},1}^2 b_{\mathsf{b},1}\Delta^3\alpha[\Delta_2](x) - a_{\mathsf{a},1}a_{\mathsf{b},1}\Delta - a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^2 - b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta^2 - b_{\mathsf{a},1}b_{\mathsf{b},1}\Delta^3 \geq 0
\end{aligned}$$

This constraint is the only element of the set $\mathrm{rp}_{\mathcal{A}}(\mathcal{R})$. As we can see, Δ_1 and Δ_2 are the only remaining Δ-quotients. They only appear as parameters of α in this constraint.

The next step in the original algorithm of Contejean et al. would be to view these constraints as (in-)equalities for polynomials in Δ and $\alpha(x_i)$ for all $x_i \in \mathcal{V}$. However, for context-dependent interpretations, this approach does not work, because the variables x_i appear in expressions of the form $\alpha[\Delta_j](x_i)$, instead. The fact that Δ_j contains variables from $\mathcal{CV}_{\mathcal{DS}}(\mathcal{R})$ makes matters more complicated. We simplify this by adding the following constraints on all Δ_j:

Definition 4.17. Given a TRS $\mathcal{R}$ and a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$, the *variable equality constraints* of $\mathcal{R}$ with respect to $\mathcal{A}$ are defined as follows:

$$\begin{aligned}
\mathrm{ve}_{\mathcal{A}}(\mathcal{R}) = \{&d_j - d_k = 0 \mid \exists c \in \mathrm{rp}_{\mathcal{A}}(\mathcal{R}) : \exists x \in \mathcal{V} : \\
&(\alpha[\frac{\Delta}{d_j}](x) \text{ occurs positively in } c \wedge \alpha[\frac{\Delta}{d_k}](x) \text{ occurs negatively in } c)\}
\end{aligned}$$

If we add these constraints, we can treat all occurrences of a variable as "equal". Then, we can proceed, as in the original algorithm, by looking at the coefficients of the polynomials in Δ and $\alpha[\_](x_i)$.

Definition 4.18. Let $\mathcal{R}$ be a TRS and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. Then the set of *modified rule polynomials* (denoted by $\mathrm{rp'}_{\mathcal{A}}(\mathcal{R})$) is constructed as follows: take all constraints in $\mathrm{rp}_{\mathcal{A}}(\mathcal{R})$, and whenever a term of the shape $\alpha[\Delta](x)$ occurs both positively and negatively in a constraint, replace these occurrences by $\alpha'(x)$. The function α' is defined by $\alpha'(x) = \alpha[\Delta](x)$ if $\alpha[\Delta](x)$ occurs in $\mathrm{rp}_{\mathcal{A}}(\mathcal{R})$.

In the constraint from $\mathrm{rp}_{\mathcal{A}}(\mathcal{R})$ in the running example, $\alpha[\Delta_1](x)$ is the only positive occurrence of a variable, and $\alpha[\Delta_2](x)$ is the only negative occurrence. Therefore, the set $\mathrm{ve}_{\mathcal{A}}(\mathcal{R})$ consists of the single constraint with the normal form

$$a_{\mathsf{a},1}a_{\mathsf{b},1} + b_{\mathsf{a},1}a_{\mathsf{b},1}\Delta + b_{\mathsf{b},1}\Delta - a_{\mathsf{a},1}a_{\mathsf{b},1} - a_{\mathsf{a},1}b_{\mathsf{b},1}\Delta - b_{\mathsf{a},1}\Delta = 0 \ .$$

We obtain the single constraint in $\mathrm{rp'}_{\mathcal{A}}(\mathcal{R})$ by replacing all occurrences of $\alpha[\Delta_1](x)$ and $\alpha[\Delta_2](x)$ by $\alpha'(x)$.

All constraints in $\mathrm{rp'}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{nz}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{ve}_{\mathcal{A}}(\mathcal{R})$ have either the shape $PO = 0$ or $PO \geq 0$, where PO denotes a polynomial in the variables Δ and all $\alpha'(x_i)$ and $\alpha[\Delta](x_i)$. The variables from $\mathcal{CV}_{\mathcal{DS}}(\mathcal{R})$ form the coefficients of these polynomials. Now, we transform these constraints by testing nonnegativity or equality to zero for each coefficient instead of the whole polynomials.

Definition 4.19. A polynomial is *absolutely nonnegative* if it has nonnegative coefficients only.

As can be easily seen, every absolutely nonnegative polynomial is also nonnegative.

Definition 4.20. Let $\mathcal{R}$ be a TRS and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. Then the *coefficient polynomials* of $\mathcal{R}$ with respect to $\mathcal{A}$ (denoted by $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$) are the constraints demanding that each coefficient in the polynomial constraints in $\mathrm{rp'}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{nz}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{ve}_{\mathcal{A}}(\mathcal{R})$ must be nonnegative (if the original constraint demanded nonnegativity of a polynomial) or equal to zero (if the original constraint set a polynomial equal to zero).

Lemma 4.21. *Let $\mathcal{R}$ be a TRS and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. If all constraints in $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$ are valid, then $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$.*

Proof. If all coefficients of a polynomial are equal to zero, then the whole polynomial is equal to zero. Furthermore, if a polynomial is absolutely nonnegative, then it is also nonnegative. Therefore, by Definition 4.20, validity of the constraints in $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$ implies validity of the constraints in $\mathrm{rp'}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{nz}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{ve}_{\mathcal{A}}(\mathcal{R})$. Because the equalities in $\mathrm{ve}_{\mathcal{A}}(\mathcal{R})$ hold, we know that for each constraint and each variable x, the Δ such that Definition 4.18 demands $\alpha'(x) = \alpha[\Delta](x)$ is unique. Hence, all occurrences of the function α' are well-defined, and by the definition of α' in Definition 4.18, all constraints in $\mathrm{rp}_{\mathcal{A}}(\mathcal{R})$ are valid, as well. Thus, all constraints in $\mathrm{rp}_{\mathcal{A}}(\mathcal{R}) \cup \mathrm{nz}_{\mathcal{A}}(\mathcal{R})$ must be valid. Now we can apply Lemma 4.16 in order to conclude that $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$. ☐

The coefficient polynomials in the running example are constructed as follows: the two constraints in $\mathrm{nz}_{\mathcal{A}}(\mathcal{R})$ do not contain any occurrences of Δ and $\alpha'(x)$, so they are directly taken into $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$. The single big constraint in $\mathrm{rp'}_{\mathcal{A}}(\mathcal{R})$ is split into a number of smaller constraints. For the constant part of the polynomial in Δ and $\alpha'(x)$, we get the constraint

$$d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1} + a_{\mathsf{a},1}^2 d_{\mathsf{b}}a_{\mathsf{b},1} - a_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1} - d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}^2 \geq 0 \enspace .$$

For the coefficients of Δ, we have the constraint

$$\begin{aligned} &d_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1} + d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1} + e_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1} + a_{\mathsf{a},1}^2 d_{\mathsf{b}}b_{\mathsf{b},1} + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1} \\ &\quad + a_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1} + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1} - a_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1} - b_{\mathsf{a},1}d_{\mathsf{b}}a_{\mathsf{b},1} - a_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1} \\ &\qquad - d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} - e_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1} - d_{\mathsf{a}}a_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} - a_{\mathsf{a},1}a_{\mathsf{b},1} \geq 0 \enspace . \end{aligned}$$

For the coefficients of Δ^2, we get

$$\begin{aligned} &d_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1} + e_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1} + e_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1} + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1} + a_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1} \\ &\quad + a_{\mathsf{a},1}b_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1} + b_{\mathsf{a},1}^2 d_{\mathsf{b}}a_{\mathsf{b},1} + b_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1} - b_{\mathsf{a},1}d_{\mathsf{b}}b_{\mathsf{b},1} - a_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1} \\ &\quad - b_{\mathsf{a},1}e_{\mathsf{b}}a_{\mathsf{b},1} - d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}^2 - d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} - e_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1} - d_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1}^2 \\ &\qquad - d_{\mathsf{a}}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} - e_{\mathsf{a}}a_{\mathsf{a},1}b_{\mathsf{b},1} - a_{\mathsf{a},1}b_{\mathsf{b},1} - b_{\mathsf{a},1}a_{\mathsf{b},1} \geq 0 \enspace . \end{aligned}$$

The coefficients of Δ^3 yield the constraint

$$\begin{aligned} &e_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1} + b_{\mathsf{a},1}^2 d_{\mathsf{b}}b_{\mathsf{b},1} + b_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1} \\ &\qquad - b_{\mathsf{a},1}e_{\mathsf{b}}b_{\mathsf{b},1} - d_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1}^2 - e_{\mathsf{a}}b_{\mathsf{a},1}b_{\mathsf{b},1} - b_{\mathsf{a},1}b_{\mathsf{b},1} \geq 0 \enspace . \end{aligned}$$

For $\alpha'(x)$, we get

$$a_{\mathsf{a},1}^2 a_{\mathsf{b},1}^2 - a_{\mathsf{a},1}^2 a_{\mathsf{b},1}^2 \geq 0 \enspace .$$

For $\Delta\alpha'(x)$, we have the constraint

$$\begin{aligned} &a_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1} + a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}^2 + a_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} + a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}^2 \\ &\qquad - a_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1} - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}^2 - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1} - a_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1} \geq 0 \enspace . \end{aligned}$$

The coefficients of $\Delta^2\alpha'(x)$ yield

$$\begin{aligned} &a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} + a_{\mathsf{a},1}b_{\mathsf{b},1}^2 + a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} + b_{\mathsf{a},1}^2 a_{\mathsf{b},1}^2 + b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} \\ &\quad - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} - b_{\mathsf{a},1}^2 a_{\mathsf{b},1} - a_{\mathsf{a},1}^2 b_{\mathsf{b},1}^2 - a_{\mathsf{a},1}b_{\mathsf{a},1}a_{\mathsf{b},1}b_{\mathsf{b},1} - a_{\mathsf{a},1}b_{\mathsf{a},1}b_{\mathsf{b},1} \geq 0 \enspace . \end{aligned}$$

Finally, $\Delta^3\alpha'(x)$ yields the constraint

$$b_{\mathsf{a},1}^2 a_{\mathsf{b},1}b_{\mathsf{b},1} + b_{\mathsf{a},1}b_{\mathsf{b},1}^2 - a_{\mathsf{a},1}b_{\mathsf{a},1}b_{\mathsf{b},1}^2 - b_{\mathsf{a},1}^2 b_{\mathsf{b},1} \geq 0 \enspace .$$

Last, we have to take the single constraint in $\mathrm{ve}_{\mathcal{A}}(\mathcal{R})$ into consideration, which yields the following last two constraints for $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$:

$$\begin{aligned} a_{\mathsf{a},1}a_{\mathsf{b},1} - a_{\mathsf{a},1}a_{\mathsf{b},1} &= 0 \\ b_{\mathsf{a},1}a_{\mathsf{b},1} + b_{\mathsf{b},1} - a_{\mathsf{a},1}b_{\mathsf{b},1} - b_{\mathsf{a},1} &= 0 \end{aligned}$$

The following theorem is a direct consequence of Lemma 4.9, Lemma 4.21, and Theorem 3.8.

Theorem 4.22. *Let $\mathcal{R}$ be a TRS and suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. If all constraints in $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$ are valid, then $\mathcal{R}$ is terminating, and the following bound on the derivation height holds for all terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and all assignments α:*

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in D} \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$$

All that is left to do is to find a satisfying assignment for the constraints $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$. They are Diophantine constraints in the variables in $\mathcal{CV}_{\mathcal{DS}}(\mathcal{R})$. As shown by Matiyasevich in 1970 [20], solvability of Diophantine constraints is undecidable. However, by putting an upper bound on the variables in $\mathcal{CV}_{\mathcal{DS}}(\mathcal{R})$, we can make the problem finite. Once we have imposed this upper bound, the constraints are solved by either giving them to a Diophantine constraint solver (see [5] or [28], for instance), or they can be encoded into a satisfiability problem in propositional logic and solved by a SAT solver (see [21] for an implementation of this approach using MiniSat [6]).

For the running example of this section, both the Diophantine constraint solver and the SAT solver could find a satisfying assignment for the constraints in less than a second. Both implementations return the following assignment:

$$\begin{aligned} d_{\mathsf{a}} &= 0 & d_{\mathsf{b}} &= 1 \\ e_{\mathsf{a}} &= 0 & e_{\mathsf{b}} &= 0 \\ a_{\mathsf{a},1} &= 1 & a_{\mathsf{b},1} &= 1 \\ b_{\mathsf{a},1} &= 1 & b_{\mathsf{b},1} &= 0 \end{aligned}$$

Except for the constant c, which does not occur in the rewrite rules, this assignment yields the same context-dependent interpretation over the reals that was also the conclusion of the example from Section 3.1. This assignment satisfies all constraints in $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$. Therefore, by Theorem 4.22, $\mathcal{R}$ is terminating, and the following bound on the derivation height of all terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ holds:

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in D} \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$$

As shown by the experimental results in Chapter 6, this procedure works already very well on the TPDB [17]. It can solve more problems than the comparable method LMPO. As shown in Chapter 5 below, the restriction of the method we have used to produce these results induces a quadratic upper bound on the derivational complexity of these rewrite systems. In comparison, LMPO can only show a polynomial upper bound on the derivation height of constructor terms.

In order to demonstrate this algorithm, we recite Example 3.1 from [9]. Consider the TRS $\mathcal{R}$ with the single rewrite rule

$$(x \circ y) \circ z \to x \circ (y \circ z)$$

over the signature containing the binary function symbol $\circ$ and the constant function symbol c. Using Hofbauer's heuristic, an interpretation into a well-founded Δ-monotone context-dependent algebra $\mathcal{A}$ can be found manually as follows. We start with a polynomial interpretation into a well-founded monotone algebra $\mathcal{B}$ using the carrier $\mathbb{N}$, the standard order $>$ on $\mathbb{N}$, and the interpretation functions $\circ_{\mathcal{B}}(x, y) = 2x + y + 1$ and $\mathsf{c}_{\mathcal{B}} = 0$. As can be easily seen, this interpretation is monotone and compatible with $\mathcal{R}$. We have several possibilities to apply Hofbauer's heuristic on this interpretation. Among several others, we also get this one:

$$\circ_{\mathcal{A}}[\Delta](x, y) = (1 + \Delta)x + y + 1 \qquad \mathsf{c}_{\mathcal{A}}[\Delta] = 0$$

The constraints for Δ-monotonicity in this example are

$$\begin{aligned} &\circ_{\mathcal{A}}[\Delta(x + \circ^1_{\mathcal{A}}(\Delta), y) - \circ_{\mathcal{A}}(x, y) \geq \Delta \\ &\circ_{\mathcal{A}}[\Delta(x, y + \circ^2_{\mathcal{A}}(\Delta)) - \circ_{\mathcal{A}}(x, y) \geq \Delta. \end{aligned}$$

Solving these constraints yields $\circ^1_{\mathcal{A}}(\Delta) \geq \frac{\Delta}{1+\Delta}$ and $\circ^2_{\mathcal{A}}(\Delta) \geq \Delta$. Therefore, we choose

$$\circ^1_{\mathcal{A}}(\Delta) = \frac{\Delta}{1 + \Delta} \qquad \circ^2_{\mathcal{A}}(\Delta) = \Delta$$

to complete the context-dependent interpretation. This context-dependent interpretation is by construction Δ-monotone. In order to verify Δ-compatibility, we have to check the constraint

$$[\alpha]_{\mathcal{A}}[\Delta]((x \circ y) \circ z) - [\alpha]_{\mathcal{A}}[\Delta](x \circ (y \circ z)) \geq \Delta \ .$$

This constraint can be simplified to

$$\begin{aligned} &[\alpha]_{\mathcal{A}}[\Delta]((x \circ y) \circ z) - [\alpha]_{\mathcal{A}}[\Delta](x \circ (y \circ z)) \\ &= (1 + \Delta)[\alpha]_{\mathcal{A}}[\frac{\Delta}{1 + \Delta}](x \circ y) + [\alpha]_{\mathcal{A}}[\Delta](z) + 1 \\ &\quad - ((1 + \Delta)[\alpha]_{\mathcal{A}}[\frac{\Delta}{1 + \Delta}]x + [\alpha]_{\mathcal{A}}[\Delta](y \circ z) + 1) \\ &= (1 + \Delta)(\frac{1 + 2\Delta}{1 + \Delta}\alpha[\frac{\Delta}{1 + 2\Delta}](x) + \alpha[\frac{\Delta}{1 + \Delta}](y) + 1) + \alpha[\Delta](z) \\ &\quad - ((1 + \Delta)\alpha[\frac{\Delta}{1 + \Delta}](x) + (1 + \Delta)\alpha[\frac{\Delta}{1 + \Delta}](y) + \alpha[\Delta](z) + 1) \\ &= (1 + 2\Delta)\alpha[\frac{\Delta}{1 + 2\Delta}](x) + (1 + \Delta)\alpha[\frac{\Delta}{1 + \Delta}](y) + 1 + \Delta + \alpha[\Delta](z) \\ &\quad - ((1 + \Delta)\alpha[\frac{\Delta}{1 + \Delta}](x) + (1 + \Delta)\alpha[\frac{\Delta}{1 + \Delta}](y) + \alpha[\Delta](z) + 1) \\ &= (1 + 2\Delta)\alpha[\frac{\Delta}{1 + 2\Delta}](x) + \Delta - ((1 + \Delta)\alpha[\frac{\Delta}{1 + \Delta}](x)) \\ &\geq \Delta \end{aligned}$$

or

$$(1 + 2\Delta)\alpha[\frac{\Delta}{1 + 2\Delta}](x) - (1 + \Delta)\alpha[\frac{\Delta}{1 + \Delta}](x) \geq 0 \ .$$

This inequality cannot be proved, since $\alpha[\frac{\Delta}{1+2\Delta}](x)$ and $\alpha[\frac{\Delta}{1+\Delta}](x)$ are independent. However, by Lemma 1 of [9], it suffices if we prove $[\alpha]_{\mathcal{A}}[\Delta](l\sigma) -$

$[\alpha]_{\mathcal{A}}[\Delta](r\sigma)$ for each rewrite rule $l \to r \in \mathcal{R}$ every $\Delta \in \mathbb{R}^+$ and all ground substitutions σ instead of our notion of Δ-compatibility in order to get termination and the upper bound on the derivation height. In our example, the following inequality would then be left to prove:

$$(1 + 2\Delta)[\alpha]_{\mathcal{A}}[\frac{\Delta}{1 + 2\Delta}](x\sigma) - (1 + \Delta)[\alpha]_{\mathcal{A}}[\frac{\Delta}{1 + \Delta}](x\sigma) \geq 0$$

This is proved by induction on the structure of $x\sigma$ in [9]. Termination of $\mathcal{R}$ and the upper bound on the derivation height are proved this way, but we could not yet produce an automatic termination proof by context-dependent interpretations. Our implementation of Hofbauer's heuristic fails because the tools we employ to solve the inequalities (`Mathematica` [19]) fail to do the induction automatically and get stuck at the last inequality. Our implementation of the adapted algorithm of Contejean et al. cannot find this context-dependent interpretation because the constraint $1 + 2\Delta - (1 + \Delta) = 0$ in $\mathrm{ve}_{\mathcal{A}}(\mathcal{R})$ is not fulfilled for all $\Delta \in \mathbb{R}^+$.

5 Quadratic Derivational Complexity

5.1 A criterion for quadratic derivational complexity

Even though context-dependent interpretations over the reals automatically give us a bound for the derivation height of terms, there is still the task to generally compute these bounds for all possible terms. In the following, we show that the class of restricted Δ-simple interpretations induces an upper bound on the derivation height of terms that is quadratic in the number of function symbols in the term.

Definition 5.1. Suppose that we have a Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. Then $k_\mathcal{A}$ is the smallest natural number such that for every every function symbol f of arity n, and for each $i \in \{1, \ldots, n\}$, we have

$$k_\mathcal{A} \geq b_{f,i} \ .$$

Furthermore, $c_\mathcal{A}$ is the smallest natural number such that for every function symbol f, we have

$$c_\mathcal{A} \geq d_f \wedge c_\mathcal{A} \geq e_f \ .$$

Since we are only dealing with finite TRSs, and we only consider function symbols of finite arity, the constants $k_\mathcal{A}$ and $c_\mathcal{A}$ are obviously well-defined.

Lemma 5.2. *Suppose that we have a restricted Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. If $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$, then we have for every ground term t*

$$[\alpha]_\mathcal{A}[\Delta](t) \leq c_\mathcal{A} k_\mathcal{A}(1 + \Delta)(|t|^2 + |t|) \ .$$

Proof. First we show that the following claim holds for every ground term t:

$$[\alpha]_\mathcal{A}[\Delta](t) \leq c_\mathcal{A}(|t| + \Delta \cdot \frac{1}{2} \cdot |t|(k_\mathcal{A}(|t| - 1) + 2)$$

Because $c_\mathcal{A}(|t|+\Delta\cdot\frac{1}{2}\cdot|t|(k_\mathcal{A}(|t|-1)+2) \leq c_\mathcal{A} k_\mathcal{A}(1+\Delta)(|t|^2+|t|)$, the lemma follows directly from the claim. We prove the claim by induction on the structure of t. Since t is a ground term, it has the shape $f(t_1, \ldots, t_n)$ for some ground terms $t_1, \ldots, t_n$ and a function symbol f of arity n. Then, by unfolding the definition of $[\alpha]_{\mathcal{A}}$, we get

$$\begin{aligned}[\alpha]_{\mathcal{A}}[\Delta](t) &= f_{\mathcal{A}}[\Delta]([\alpha]_{\mathcal{A}}[f^1_{\mathcal{A}}(\Delta)](t_1), \ldots, [\alpha]_{\mathcal{A}}[f^n_{\mathcal{A}}(\Delta)](t_n)) \\ &= d_f + e_f\Delta + \sum_{i=1}^{n}((a_{f,i} + b_{f,i}\Delta)[\alpha]_{\mathcal{A}}[f^i_{\mathcal{A}}(\Delta)](t_i)) \\ &= d_f + e_f\Delta \\ &\quad + \sum_{i=1}^{n}((a_{f,i} + b_{f,i}\Delta)[\alpha]_{\mathcal{A}}[\frac{\Delta}{a_{f,i} + b_{f,i}\Delta}](t_i))\end{aligned}$$

If $n = 0$, then f is a constant function symbol, and $|t| = 1$. For that case, this yields

$$\begin{aligned}[\alpha]_{\mathcal{A}}[\Delta](t) &= d_f + e_f\Delta \\ &\leq c_{\mathcal{A}}(1 + \Delta) \ ,\end{aligned}$$

which is what we wanted to show. In the other case, we can apply the induction hypothesis to the term $[\alpha]_{\mathcal{A}}[\frac{\Delta}{a_{f,i}+b_{f,i}\Delta}](t_i)$ for each $i \in \{1, \ldots, n\}$:

$$\begin{aligned}& d_f + e_f\Delta \\ &\quad + \sum_{i=1}^{n}((a_{f,i} + b_{f,i}\Delta)[\alpha]_{\mathcal{A}}[\frac{\Delta}{a_{f,i} + b_{f,i}\Delta}](t_i)) \\ \leq\ & d_f + e_f\Delta \\ &\quad + \sum_{i=1}^{n}((a_{f,i} + b_{f,i}\Delta) \\ &\quad \cdot c_{\mathcal{A}}(|t_i| + \frac{\Delta}{a_{f,i} + b_{f,i}\Delta} \cdot \frac{1}{2} \cdot |t_i|(k_{\mathcal{A}}(|t_i| - 1) + 2))) \ .\end{aligned}$$

We can distribute $(a_{f,i} + b_{f,i}\Delta)$ inside the brackets after $c_{\mathcal{A}}$ in order to simplify this, since it is canceled with the denominator of the division in the second summand:

$$\begin{aligned}& d_f + e_f\Delta \\ &\quad + \sum_{i=1}^{n}((a_{f,i} + b_{f,i}\Delta) \\ &\quad \cdot c_{\mathcal{A}}(|t_i| + \frac{\Delta}{a_{f,i} + b_{f,i}\Delta} \cdot \frac{1}{2} \cdot |t_i|(k_{\mathcal{A}}(|t_i| - 1) + 2))) \\ =\ & d_f + e_f\Delta \\ &\quad + \sum_{i=1}^{n}(c_{\mathcal{A}}((a_{f,i} + b_{f,i}\Delta)|t_i| + \Delta \cdot \frac{1}{2} \cdot |t_i|(k_{\mathcal{A}}(|t_i| - 1) + 2))) \ .\end{aligned}$$

By Definition 5.1, we know that $d_f \leq c_{\mathcal{A}}$, $e_f \leq c_{\mathcal{A}}$, and $b_{f,i} \leq k_{\mathcal{A}}$ for each $i \in \{1, \ldots, n\}$. Furthermore, by Definition 4.7, we know that $a_{f,i} \leq 1$ for each

$i \in \{1, \ldots, n\}$. Applying this yields

$$\begin{aligned}
&d_f + e_f\Delta \\
&\quad + \sum_{i=1}^{n}(c_{\mathcal{A}}((a_{f,i} + b_{f,i}\Delta)|t_i| + \Delta \cdot \frac{1}{2} \cdot |t_i|(k_{\mathcal{A}}(|t_i| - 1) + 2))) \\
&\leq c_{\mathcal{A}} + c_{\mathcal{A}}\Delta + \sum_{i=1}^{n}(c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta)|t_i| + \Delta \cdot \frac{1}{2} \cdot |t_i|(k_{\mathcal{A}}(|t_i| - 1) + 2))) \ .
\end{aligned}$$

By doing two factorizations, we get

$$\begin{aligned}
&c_{\mathcal{A}} + c_{\mathcal{A}}\Delta + \sum_{i=1}^{n}(c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta)|t_i| + \Delta \cdot \frac{1}{2} \cdot |t_i|(k_{\mathcal{A}}(|t_i| - 1) + 2))) \\
&= c_{\mathcal{A}} + c_{\mathcal{A}}\Delta + \sum_{i=1}^{n}(|t_i| \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t_i| - 1) + 2))) \\
&= c_{\mathcal{A}}(1 + \Delta) + \sum_{i=1}^{n}(|t_i| \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t_i| - 1) + 2))) \ .
\end{aligned}$$

Also, by Definition 2.4, we have $|t_i| \leq |t| - 1$ for each $i \in \{1, \ldots, n\}$:

$$\begin{aligned}
&c_{\mathcal{A}}(1 + \Delta) + \sum_{i=1}^{n}(|t_i| \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t_i| - 1) + 2))) \\
&\leq c_{\mathcal{A}}(1 + \Delta) + \sum_{i=1}^{n}(|t_i| \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t| - 2) + 2))) \ .
\end{aligned}$$

Factorizing the term $c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t_i| - 1) + 2)))$ here yields

$$\begin{aligned}
&c_{\mathcal{A}}(1 + \Delta) + \sum_{i=1}^{n}(|t_i| \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t| - 2) + 2))) \\
&\leq c_{\mathcal{A}}(1 + \Delta) + (\sum_{i=1}^{n}|t_i|) \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t| - 2) + 2)) \ .
\end{aligned}$$

From Definition 2.4, we get that $\sum_{i=1}^{n}|t_i| = |t| - 1$. Therefore,

$$\begin{aligned}
&c_{\mathcal{A}}(1 + \Delta) + (\sum_{i=1}^{n}|t_i|) \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t| - 2) + 2)) \\
&\leq c_{\mathcal{A}}(1 + \Delta) + (|t| - 1) \cdot c_{\mathcal{A}}((1 + k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2} \cdot (k_{\mathcal{A}}(|t| - 2) + 2)) \ .
\end{aligned}$$

Now we only need to apply Factorization and Distributivity some more in order to get the result:

$$
\begin{aligned}
& c_{\mathcal{A}}(1+\Delta) + (|t|-1)\cdot c_{\mathcal{A}}((1+k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2}\cdot (k_{\mathcal{A}}(|t|-2)+2)) \\
&= c_{\mathcal{A}}(1+\Delta + (|t|-1)\cdot ((1+k_{\mathcal{A}}\Delta) + \Delta \cdot \frac{1}{2}\cdot (k_{\mathcal{A}}(|t|-2)+2))) \\
&= c_{\mathcal{A}}(1+\Delta + (1+k_{\mathcal{A}}\Delta)(|t|-1) + \Delta \cdot \frac{1}{2}\cdot (|t|-1)(k_{\mathcal{A}}(|t|-2)+2)) \\
&= c_{\mathcal{A}}(|t| + \Delta + k_{\mathcal{A}}\Delta(|t|-1) + \Delta \cdot \frac{1}{2}\cdot (|t|-1)(k_{\mathcal{A}}(|t|-2)+2)) \\
&= c_{\mathcal{A}}(|t| + \Delta(1 + k_{\mathcal{A}}(|t|-1) + \frac{1}{2}\cdot (|t|-1)(k_{\mathcal{A}}(|t|-2)+2))) \\
&= c_{\mathcal{A}}(|t| + \Delta \cdot \frac{1}{2}\cdot (2 + 2k_{\mathcal{A}}(|t|-1) + (|t|-1)(k_{\mathcal{A}}(|t|-2)+2))) \\
&= c_{\mathcal{A}}(|t| + \Delta \cdot \frac{1}{2}\cdot (2 + 2k_{\mathcal{A}}(|t|-1) + (|t|-2)k_{\mathcal{A}}(|t|-1) + 2(|t|-1))) \\
&= c_{\mathcal{A}}(|t| + \Delta \cdot \frac{1}{2}\cdot (2 + |t|\cdot k_{\mathcal{A}}(|t|-1) + 2(|t|-1))) \\
&= c_{\mathcal{A}}(|t| + \Delta \cdot \frac{1}{2}\cdot (2|t| + |t|\cdot k_{\mathcal{A}}(|t|-1))) \\
&= c_{\mathcal{A}}(|t| + \Delta \cdot \frac{1}{2}\cdot |t|(k_{\mathcal{A}}(|t|-1)+2))
\end{aligned}
$$

This concludes

$$[\alpha]_{\mathcal{A}}[\Delta](t) \leq c_{\mathcal{A}}(|t| + \Delta \cdot \frac{1}{2}\cdot |t|(k_{\mathcal{A}}(|t|-1)+2)) \ ,$$

which is what we wanted to show. ☐

Theorem 5.3. *Suppose that we have a restricted Δ-simple interpretation into a well-founded Δ-monotone algebra $\mathcal{A}$. If $\mathcal{A}$ is Δ-compatible with $\mathcal{R}$, then $\mathcal{R}$ is terminating, and we have $\mathrm{dc}_{\mathcal{R}}(m) \in \mathcal{O}(m^2)$.*

Proof. By Theorem 3.8, $\mathcal{R}$ is terminating, and for every term t, we have

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta\in D} \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta} \ .$$

Together with Lemma 5.2, we get

$$
\begin{aligned}
\mathrm{dh}_{\mathcal{R}}(t) &\leq \inf_{\Delta\in D} \frac{c_{\mathcal{A}}k_{\mathcal{A}}(1+\Delta)(|t|^2+|t|)}{\Delta} \\
&= c_{\mathcal{A}}k_{\mathcal{A}}(|t|^2+|t|) \ .
\end{aligned}
$$

Obviously,

$$c_{\mathcal{A}}k_{\mathcal{A}}(|t|^2+|t|) \in \mathcal{O}(|t|^2) \ ,$$

thus we conclude the theorem. ☐

In the running example from Section 4.3, we considered the rewrite system from Table 2.1. In the end, we obtained a Δ-simple interpretation with the interpretation functions

$$\mathsf{a}_{\mathcal{A}}[\Delta](x) = (1+\Delta)x \qquad \mathsf{a}^1_{\mathcal{A}}(\Delta) = \frac{\Delta}{1+\Delta}$$
$$\mathsf{b}_{\mathcal{A}}[\Delta](x) = x+1 \qquad \mathsf{b}^1_{\mathcal{A}}(\Delta) = \Delta \ .$$

This interpretation is actually a restricted Δ-simple interpretation. Therefore, Theorem 5.3 applies to $\mathcal{R}$, which certifies a quadratic upper bound on the derivational complexity of $\mathcal{R}$. Moreover, for every ground term t, Lemma 5.2 gives us the upper bound

$$[\alpha]_{\mathcal{A}}[\Delta](t) \leq (1+\Delta)(|t|^2 + |t|)$$

on the derivation height of $\mathcal{R}$. Applying Theorem 3.8 concludes

$$\mathrm{dh}_{\mathcal{R}}(t) \leq \inf_{\Delta \in \mathbb{R}_0^+} \frac{(1+\Delta)(|t|^2+|t|)}{\Delta} = |t|^2 + |t| \ .$$

Even though this bound is slightly less accurate than the bound that was calculated for this example in Section 3.1, it is still in the same complexity class. Moreover, it is much easier to calculate this bound automatically than the bound from Section 3.1, considering that only a special family of terms was chosen there, and that the bound on $[\alpha]_{\mathcal{A}}[\Delta](t)$ for terms t from this family was calculated by an inductive proof that might not be so easy to automate.

5.2 Quadratic Derivational Complexity and Non-Simple Termination

As we have seen in Section 3.4, context-dependent interpretations can handle rewrite systems which are terminating, but not simply terminating. However, the interpretation function were not really "well-behaved", since they were discontinuous. Still, even restricted Δ-simple interpretations can handle non-simple termination. Consider the one-rule TRS $\mathcal{R}$ from Section 3.4 again:

$$\mathsf{a}(\mathsf{a}(x)) \to \mathsf{a}(\mathsf{b}(\mathsf{a}(x)))$$

By applying the algorithm described in Section 4.3, we obtain the following automatically generated restricted Δ-simple interpretation into a context-dependent algebra $\mathcal{A}$ over the reals:

$$\mathsf{a}_{\mathcal{A}}[\Delta](x) = 2\Delta x + 2 \qquad \mathsf{a}^1_{\mathcal{A}}(\Delta) = \frac{1}{2}$$
$$\mathsf{b}_{\mathcal{A}}[\Delta](x) = \Delta x + \Delta \qquad \mathsf{b}^1_{\mathcal{A}}(\Delta) = 1$$

Checking that this is indeed a Δ-simple interpretation and applying Lemma 4.8 yields that Δ-monotonicity indeed holds for this interpretation. In order to verify Δ-compatibility with $\mathcal{R}$, we have to check that the inequality

$$[\alpha]_{\mathcal{A}}[\Delta](\mathsf{a}(\mathsf{a}(x))) - [\alpha]_{\mathcal{A}}[\Delta](\mathsf{a}(\mathsf{b}(\mathsf{a}(x)))) \geq \Delta$$

holds for all $\Delta \in \mathbb{R}^+$ and all assignments α. Indeed, we have

$$\begin{aligned}
&[\alpha]_\mathcal{A}[\Delta](\mathsf{a}(\mathsf{a}(x))) - [\alpha]_\mathcal{A}[\Delta](\mathsf{a}(\mathsf{b}(\mathsf{a}(x)))) \\
&\quad = 2\Delta[\alpha]_\mathcal{A}[\frac{1}{2}](\mathsf{a}(x)) + 2 - (2\Delta[\alpha]_\mathcal{A}[\frac{1}{2}](\mathsf{b}(\mathsf{a}(x))) + 2) \\
&\quad = 2\Delta(2 \cdot \frac{1}{2} \cdot [\alpha]_\mathcal{A}[\frac{1}{2}](x) + 2) + 2 - (2\Delta(\frac{1}{2} \cdot [\alpha]_\mathcal{A}[1](\mathsf{a}(x)) + \frac{1}{2}) + 2) \\
&\quad = 2\Delta[\alpha]_\mathcal{A}[\frac{1}{2}](x) + 4\Delta + 2 - (2\Delta(\frac{1}{2} \cdot (2[\alpha]_\mathcal{A}[\frac{1}{2}](x) + 2) + \frac{1}{2}) + 2) \\
&\quad = 2\Delta[\alpha]_\mathcal{A}[\frac{1}{2}](x) + 4\Delta + 2 - 2\Delta[\alpha]_\mathcal{A}[\frac{1}{2}](x) + 2\Delta + \Delta + 2 \\
&\quad \geq \Delta \ .
\end{aligned}$$

Now we can apply Theorem 5.3 in order to conclude termination of $\mathcal{R}$ and

$$\mathrm{dc}_\mathcal{R}(m) \in \mathcal{O}(m^2) \ .$$

More exactly, Lemma 5.2 yields the following bound for all ground terms t:

$$[\alpha]_\mathcal{A}[\Delta](t) \leq 4(1 + \Delta)(|t|^2 + |t|)$$

with $c_\mathcal{A} = 2$ and $k_\mathcal{A} = 2$. By applying Theorem 3.8, we can conclude

$$\mathrm{dh}_\mathcal{R}(t) \leq \inf_{\Delta \in \mathbb{R}_0^+} \frac{4(1 + \Delta)(|t|^2 + |t|)}{\Delta} = 4(|t|^2 + |t|) \ .$$

For every two as that occur in a row in the original term, exactly one rewrite step can be done. Therefore, the real derivational complexity of $\mathcal{R}$ is linear. This means that our result is not optimal for this example. Still, this interpretation can certify that the derivational complexity of $\mathcal{R}$ is not exponential, which is already a good thing.

6 Implementation and Experimental Results

The methods described in the previous chapters have been implemented. In this chapter, we discuss these implementations.

6.1 Hofbauer's Heuristic

First, the heuristic of Hofbauer we described in Section 4.1 was implemented. The implementation of this approach has been split into the three programs `cdiprover`, `cdisolver` and `cdibounds`.

Cdiprover : This program takes a TRS as input. The input file has to follow the same syntax as the files in the TPDB. The format specification for this database is available with the termination problems database [17]. First, `cdiprover` tries to find a suitable polynomial interpretation for the given TRS. For this step, the program `dioprover`, which was implemented by Winkler [28], is used. `Dioprover` searches polynomial interpretations using the (non-adapted) algorithm of Contejean et al. [5]. Afterward, we try to apply Hofbauer's heuristic as described in Table 4.1. This generates a set of "stubs" of context-dependent interpretations. Each stub contains the interpretation functions $f_{\mathcal{A}}$ for all function symbols f. The program tries to apply the heuristic at up to two spots in the interpretation at the same time. `Cdiprover` outputs the stubs of the context-dependent interpretations into files ending with `.solution<n>.con`, where `<n>` is a running number. These files can be directly used as input for `cdisolver`. The `.con` files follow the grammar shown in Table 6.1. As we can see, a `spec` consists of four parts. The first part (`<idlist>`) declares a list of variables that is going to be used in the parts below. Otherwise, it would not be possible to differentiate between variables and constant function symbols when parsing a file. The second part (`<listofrules>`) declares the rewrite rules of the considered TRS. The third part (`<inters>`) declares the stub of the context-dependent interpretation. Each element in this "list" corresponds to a function $f_{\mathcal{A}}$ for some function symbol f. The last part (`<monoconstlist>`) declares a list of pairs (a, b), where a is a function symbol, and b is a number in $\{1, \dots, \text{arity}(a)\}$. Every such pair means that the Δ-monotonicity constraint has to be checked for this function symbol and argument position. `Cdiprover` is written in `OCaml`.

Cdisolver : `Cdisolver` takes as input a stub of a context-dependent interpretation. It is possible to use the output files of `cdiprover` directly as input

Table 6.1: The output grammar of `cdiprover`.

`<spec>`	:= (`VAR` `<idlist>`) (`RULES` `<listofrules>`)
	(`INTER` `<inters>`) (`MONOCONSTS` `<monoconstlist>`)
`<idlist>`	:= $\epsilon \mid$ *id* `<idlist>`
`<listofrules>`	:= $\epsilon \mid$ `<rule>` `<listofrules>`
`<rule>`	:= `<term>` `->` `<term>`
`<term>`	:= *id* $\mid$ *id*() $\mid$ *id*(`<termlist>`)
`<termlist>`	:= `<term>` $\mid$ `<term>`, `<termlist>`
`<inters>`	:= `<inters>` *id* = (`<poly>`)
	$\mid$ `<inters>` *id*() = (`<poly>`)
	$\mid$ `<inters>` *id*(`<varlist>`) = (`<poly>`)
`<varlist>`	:= *id* $\mid$ *id*, `<varlist>`
`<poly>`	:= $\epsilon \mid$ `<poly>` + (*int* + *int* ∗ `delta`)`<monovars>`
`<monovars>`	:= $\epsilon \mid$ ∗*id* `<monovars>`
`<monoconstlist>`	:= $\epsilon \mid$ `<monoconst>` `<monoconstlist>`
`<monoconst>`	:= *id* *int* $\mid$ *id*(`<varlist>`) *int*

for `cdisolver`. The expected grammar for the input files of `cdisolver` is shown in Table 6.2. The first four parts have exactly the same meaning as

Table 6.2: The input grammar of `cdisolver`.

`<spec>`	:= (`VAR` `<idlist>`) (`RULES` `<listofrules>`)
	(`INTER` `<inters>`) (`MONOCONSTS` `<monoconstlist>`)
	$\mid$ (`VAR` `<idlist>`) (`RULES` `<listofrules>`)
	(`INTER` `<inters>`) (`MONOCONSTS` `<monoconstlist>`)
	(`MINTERMS` `<mintermlist>`)
`<mintermlist>`	:= $\epsilon \mid$ `<term>` `<mintermlist>`

the output of `cdiprover`. The optional fifth part (`<mintermlist>`) of the input file declares a list of terms. If a context-dependent interpretation which induces termination can be constructed from the stub in the given input file, then the upper bounds on the derivation height are computed for these terms.

After reading the input file, `cdisolver` formulates the constraints for Δ-monotonicity. The solutions to these constraints are used to construct the functions $f^i_{\mathcal{A}}$ for all function symbols f and all $i \in \{1, \ldots, n\}$ with $n = \text{arity}(f)$. Then the constraints for Δ-compatibility are constructed and checked. If they are valid for all assignments α and all $\Delta > 0$, then the complete context-dependent interpretation is given as output. Otherwise, no context-dependent interpretation is returned for the given stub. For the case that the procedure ends with success, we may specify a number of terms for which we want to test the upper bound on the derivation height. If such terms are given, then for each of them, the limit

$$\inf_{\Delta \in D} \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$$

is calculated and returned, where t is the considered term. The core of `cdisolver` is written in C. It is wrapped by an interface in `OCaml`, which is responsible for the input and the output of the program. For solving the constraints for Δ-monotonicity and Δ-compatibility, and for calculating the bounds on the derivation height of terms, `Mathematica` [19] is used. The constraints are built as packets in the core of the program that is written in C. `Mathlink` serves as an interface between C and `Mathematica`, which transfers the packets between these two parts of the solver. Finally, `Mathematica` does the actual solving of the constraints which are encoded in the packets.

Cdibounds : This program is a subset of `cdisolver`. It takes a context-dependent interpretation and a number of terms as input. The expected grammar of the input file is shown in Table 6.3. The first, second, and

Table 6.3: The input grammar of `cdibounds`.

```
      <spec> := (VAR <idlist>) (INTER <inters>)
                (FTAUI <ftauilist>) (MINTERMS <mintermlist>)
      <poly> := ε | <poly> + int <varexps>
   <varexps> := ε | *delta <varexps> | *id <varexps>
                | *(delta/(int + int * delta))<varexps>
 <ftauilist> := ε | <ftaui> <ftauilist>
     <ftaui> := id _tau_ int(delta) = (<poly>)
```

fourth part of the input file (`<idlist>`, `<inters>`, and `<mintermlist>`) have the same meaning as the corresponding parts in the input files for `cdisolver`. The third part (`<ftauilist>`) declares the functions $f^i_{\mathcal{A}}$ for all function symbols f and all $i \in \{1, \dots, \text{arity}(f)\}$. `Cdibounds` assumes that the given context-dependent interpretation is already known to be Δ-monotone and Δ-compatible with the considered TRS. Like in `cdisolver`, `Mathlink` and `Mathematica` are then used to compute the upper bound

$$\inf_{\Delta \in D} \frac{[\alpha]_{\mathcal{A}}[\Delta](t)}{\Delta}$$

for the derivation height of each term t given in the input file.

We have tested the combination of `cdiprover` and `cdisolver` on version 3.2 of the TPDB [15]. The database contains 865 TRSs, of which 686 are known to be terminating. We used two variations of `cdiprover` in this test. One of them generated stubs of context-dependent interpretations where in each stub, Hofbauer's heuristic was applied in exactly one place. The other one allowed up to two applications of the heuristic per stub. Obviously, the first variation is less powerful than the second. However, the second approach is also more computation intensive. For some TRSs in the database, the second approach generated already several hundred stubs. Both variations were tested on a on a Sunfire x4600 (x86_64 architecture) with 8 AMD Opteron 2.6GHz dual core processors and 64GB of RAM. `Cdiprover` was given 10 seconds to generate a polynomial interpretation and the stubs. Additionally, for each stub, `cdisolver` took between 250 and 500 milliseconds to generate a complete context-dependent interpretation and to verify or reject it. The following table shows the results of these tests:

Method	Polynomial interpretations	CDIs
1 application of the heuristic	120	11
Up to 2 applications of the heuristic	120	17

As we can see, there are not many systems in the database which can be proved terminating by context-dependent interpretations found by the heuristic. Also, we have not investigated yet how much improvement for the upper bound on the derivation height of terms we get for these 17 rewrite systems. Therefore, the development of these three tools has been stopped, and the approach described in section 4.3 has been implemented, instead.

6.2 The adapted Algorithm of Contejean et al.

The adapted algorithm of Contejean et al. has been implemented in two similar tools called `cdiprover2` and `cdiprover3`. These two tools take as input a TRS in the format of the TPDB [17]. Both use the same algorithm to generate sets of Diophantine constraints. However, after putting upper bounds on the values of the coefficient variables, the two tools use different approaches to continue. `Cdiprover2` solves the constraints by using the Diophantine constraint solver from [28], while `cdiprover3` transforms the problem into a propositional satisfiability problem and applies the SAT solver MiniSat [6] to solve it.

Both tools have methods to prove that a system is terminating with a polynomial derivational complexity. They currently use two methods to achieve this: polynomial interpretations with additive polynomials (which, by Lemma 2.29, induce a linear upper bound on the derivational complexity), and context-dependent interpretations over the reals with restricted Δ-simple polynomials (which, by Theorem 5.3, induce a quadratic upper bound on the derivational complexity). The interpretations are found by generating the according parametric interpretations, generating constraints which are sufficient to enforce the termination criteria, and solving the constraints. For polynomial interpretations with additive polynomials, this is done according to the original algorithm of

Contejean et al. [5]. For context-dependent interpretations over the reals with Δ-simple polynomials, the constraints in $\mathrm{cp}_{\mathcal{A}}(\mathcal{R})$ are generated by the adapted algorithm of Contejean et al. [5] as described in Section 4.3.

Apart from the methods to prove polynomial derivational complexity of rewrite systems, `cdiprover2` and `cdiprover3` also incorporate other termination proof methods. Since they are extensions of `dioprover` [28], they are naturally able to search for the same subclasses of polynomial interpretations as `dioprover`. These are polynomial interpretations with linear, *simple*, *simple-mixed*, and *quadratic* polynomials. Moreover, the option to search for $\Pi(0)$-interpretations has been added to the tools. Last, it is also possible to search for context-dependent interpretations with (non-restricted) Δ-simple polynomials.

The output of the two tools is a suitable polynomial or context-dependent interpretation if one can be found. Both tools are extensions of the program `dioprover` [28], which implements the original algorithm of Contejean et al. They are written in `OCaml`.

The capabilities of `cdiprover2` and `cdiprover3` at looking for proofs of polynomial derivational complexity of rewrite systems have been tested on the recent version (4.0) of the TPDB [17]. They were run on an i686 with an Intel Pentium 3 GHz dual core processor and 1 GB of memory. As in this year's termination competition [16], a timeout of 120 seconds was used. The results of the tests are listed in the following table:

	`Cdiprover2`			`Cdiprover3`		
	A.	CDI	A. + CDI	A.	CDI	A. + CDI
YES	41	62	72	41	83	84
MAYBE	1309	206	204	1317	330	324
TIMEOUT	8	1090	1082	0	945	950
Avg. YES time	0.512	7.283	3.818	0.021	9.147	5.323

In this table, "A." denotes polynomial interpretations with additive polynomials. "CDI" means context-dependent interpretations over the reals with restricted Δ-simple polynomials. The method "Additive + CDI" first tries to find a polynomial interpretation with additive polynomials. If such an interpretation cannot be found, then a context-dependent interpretation over the reals with Δ-simple polynomials is searched. As would be expected, context-dependent interpretations over the reals are quite a bit more powerful than polynomial interpretations with additive polynomials. However, searching for additive polynomials is much quicker than searching for restricted Δ-simple polynomials. As we can see, adding a search for additive polynomials before looking for restricted Δ-simple polynomials adds a little power to the prover but also reduces the average time needed to find a termination proof considerably. Last, we see that encoding the final constraints into SAT lets us find quite some more solutions in the limited time frame than using the Diophantine constraint solver.

Even though we do not know about any other efforts to implement automatic proofs of polynomial upper bounds on the derivational complexity of rewrite systems, there are methods which can proof another notion of termination in polynomial time. These methods can show that functions encoded by a rewrite system can be computed in FP. Since these are the closest comparisons we can get for the tests mentioned above, we have also tested two such methods, namely $\Pi(0)$-interpretations (as described in Section 2.4, we used simple-mixed polynomials for the defined symbols) generated by cdiprover3 and an implementation of LMPO (as described in Section 7.1 below) by Martin Avanzini. We want to mention that another such method, POP*, which is an extension of POP by Arai and Moser [1], is currently being developed by Moser and Avanzini. The tests were run on the current TPDB [17] with a timeout of 120 seconds for each TRS. The results are shown in the table below:

	A.	CDI	A. + CDI	$\Pi(0)$	LMPO
YES	41	83	84	166	74
MAYBE	1317	330	324	522	1212
TIMEOUT	0	945	950	670	72
Avg. YES time	0.021	9.147	5.323	2.072	0.027

The tests in the first three columns were done with `cdiprover3`. As we can see, the results we get are pretty comparable to LMPO. While $\Pi(0)$-interpretations seem to be rather powerful, it must also be said that they prove that encoded functions are computable in FP, while our method proves polynomial derivational complexity. The only result which we get from $\Pi(0)$-interpretations for derivational complexity is that the derivation height of constructor terms is bounded by a polynomial. The same difference holds for `cdiprover3` and LMPO, but there, `cdiprover3` can be applied to more examples than LMPO.

7 Related Work

7.1 LMPO

LMPO was introduced by Marion in 2003 [18]. It is a restriction of the well-known multiset path order. Like the $\Pi(0)$-interpretations defined in Section 2.4, it has a close relation to functions computable in FP.

Definition 7.1. Given a rewrite system $\mathcal{R}$, a *valency* for a defined function symbol f is a function $\nu(f, \cdot) : \{1, \dots, n\} \to \{0, 1\}$, where $n = \text{arity}(f)$.

Definition 7.2. Let $\mathcal{R}$ be a rewrite system, ν a valency for all $f \in \mathcal{F_D}$ and $\approx$ an equivalence relation on $\mathcal{F_D}$. Then the *permutative congruence for $\approx$ which respects* ν is the smallest equivalence relation $\sim$ on terms such that

- $f(t_1, \dots, t_n) \sim f(s_1, \dots, s_n)$ whenever $f \in \mathcal{F_C}$ and $t_i \sim s_i$ for all $i \in \{1, \dots, n\}$, and
- $f(t_1, \dots, t_n) \sim g(s_1, \dots, s_n)$ whenever $f, g \in \mathcal{F_D}$, $f \approx g$, and there exists a permutation π such that $\nu(g, i) = \nu(f, \pi(i))$ and $s_i \sim t_{\pi(i)}$ for all $i \in \{1, \dots, n\}$.

Definition 7.3. Let $\mathcal{R}$ be a rewrite system, and let $\gtrsim$ be an order on $\mathcal{F_D}$. Let $\approx$ denote the reflexive part of $\gtrsim$, and $>$ the irreflexive part of $\gtrsim$. Then the *light multiset path order* consisting of two orders denoted by $>_1^{\text{lmpo}}$ and $>_0^{\text{lmpo}}$ is defined as follows: Let $\sim$ denote the permutative congruence for $\approx$ which respects ν. Given two terms $t = f(t_1, \dots, t_n)$ and s, we have $t >_1^{\text{lmpo}} s$ if

1. $t_i >_1^{\text{lmpo}} s$ or $t_i \sim s$ for some $i \in \{1, \dots, n\}$, and if $f \in \mathcal{F_D}$, then $\nu(f, i) = 1$; or
2. $s = g(s_1, \dots, s_m)$, $f \in \mathcal{F_D}$, $t >_1^{\text{lmpo}} s_i$ for all $i \in \{1, \dots, m\}$, and either $g \in \mathcal{F_C}$ or $f > g$.

We have $t >_0^{\text{lmpo}} s$ if

1. $t_i >_0^{\text{lmpo}} s$ or $t_i \sim s$ for some $i \in \{1, \dots, n\}$; or
2. $s = g(s_1, \dots, s_m)$, $f \in \mathcal{F_D}$, $g \in \mathcal{F_C}$, and $t >_0^{\text{lmpo}} s_i$ for all $i \in \{1, \dots, m\}$; or
3. $s = g(s_1, \dots, s_m)$, $f, g \in \mathcal{F_D}$, $f > g$, and $t >_{\nu(f,i)}^{\text{lmpo}} s_i$ for all $i \in \{1, \dots, m\}$; or
4. $s = g(s_1, \dots, s_n)$, $f \approx g$, and there exists a permutation π such that
   - $\nu(g, i) = \nu(f, \pi(i))$ for all $i \in \{1, \dots, n\}$,

- $t_{\pi(i)} >_1^{\text{lmpo}} s_i$ for some $i \in \{1 \dots, n\}$ with $\nu(g, i) = 1$, and
- $t_{\pi(i)} >_{\nu(g,i)}^{\text{lmpo}} s_i$ or $t_{\pi(i)} \sim s_i$ for all $i \in \{1, \dots, n\}$.

Theorem 7.4 (Marion 2003)**.** *Let $\mathcal{R}$ be a rewrite system, ν a valency for all defined function symbols, and $\geq$ a precedence over the defined function symbols. If for all rewrite rules $l \to r \in \mathcal{R}$, we have $l >_0^{\text{lmpo}} r$, then $\mathcal{R}$ is terminating.*

Theorem 7.5 (Marion 2003)**.** *A function $f : A^n \to A$ is computable in FP if and only if there exist a confluent constructor system $\mathcal{R}$ which computes f with $\mathcal{T}(\mathcal{F}_\mathcal{C}) = A$, a valency ν for all defined function symbols, and a precedence $\geq$ on the defined function symbols such that for all rewrite rules $l \to r \in \mathcal{R}$, we have $l >_0^{\text{lmpo}} r$.*

For every LMPO, there exists a corresponding multiset path order such that

$$s >_0^{\text{lmpo}} t \Longrightarrow s >^{\text{mpo}} t \ .$$

The precedence for this multiset path order is chosen such that it contains the precedence of the LMPO and $f > g$ whenever $f \in \mathcal{T}_\mathcal{D}$ and $g \in \mathcal{T}_\mathcal{C}$. This can be easily seen by induction the the structure of s and t, where rule 1 of $>_0^{\text{lmpo}}$ corresponds to rule 1 of the multiset path order, rule 2 and 3 of $>_0^{\text{lmpo}}$ correspond to rule 2 of MPO, and rule 4 of $>_0^{\text{lmpo}}$ corresponds to the last rule of MPO.

LMPO can be used to certify that a function computed by a rewrite system is computable in FP. however, as already mentioned in Chapter 6, it says nothing about the derivational complexity of the rewrite system.

7.2 Polynomial interpretations over the reals

There are two main differences between implementations of traditional polynomial interpretations and our implementation of context-dependent interpretations. The first difference is that our implementation uses a subset of real numbers as domain of the interpretations, and the second difference is the addition of context-dependency. While we do not know about any other implementations in termination provers which use context-dependency, there has been recent work by Lucas about using the rational and algebraic real numbers as a domain for polynomial interpretations [12, 13, 14]. The following theorems are results from [12].

Theorem 7.6 (Lucas 2005)**.** *Let $\mathcal{R}$ be a rewrite system and suppose that we have an interpretation into an algebra $\mathcal{A} = (A, [\cdot]_\mathcal{A})$, where $A \subseteq \mathbb{R}$ such that for every function symbol $f \in \mathcal{F}$, the interpretation function $f_\mathcal{A}$ is continuous and differentiable in all arguments. If for all $i \in \{1, \dots, n\}$ with $n = \text{arity}(f)$, we have $\frac{\partial f_\mathcal{A}(x_1,\dots,x_i,\dots,x_n)}{\partial x_i} \geq 1$, then for all $\delta \in \mathbb{R}^+$, the function $f_\mathcal{A}$ is monotone with respect to the order $>_\delta$, where $>_\delta$ is defined like $>_\Delta$ in Definition 3.7.*

Since for each $\delta \in \mathbb{R}^+$, the order $>_\delta$ is well-founded, this implies that $(A, [\cdot]_\mathcal{A}, >_\delta)$ is a well-founded monotone algebra if the conditions of this theorem are fulfilled.

Theorem 7.7 (Lucas 2005)**.** *Let $\mathcal{R}$ be a rewrite system and suppose that we have an interpretation into an algebra $\mathcal{A} = (A, [\cdot]_\mathcal{A})$ such that $(A, [\cdot]_\mathcal{A}, >_\delta)$ is a well-founded monotone algebra for each $\delta \in \mathbb{R}^+$. Then for each rewrite rule $l \to r \in \mathcal{R}$, let $\delta_{l,r} = \inf_\alpha [\alpha]_\mathcal{A}(l) - [\alpha]_\mathcal{A}(r)$. Furthermore, let $\delta = \min\{\delta_{l,r} \mid l \to r \in \mathcal{R}\}$. If $\delta \in \mathbb{R}^+$, then $(A, [\cdot]_\mathcal{A}, >_\delta)$ is compatible with $\mathcal{R}$.*

If the conditions of this theorem are fulfilled, then by Theorem 2.20, $\mathcal{R}$ is terminating. Lucas suggests an algorithm similar to the algorithm of Contejean et al. [5] to find interpretations using $\mathbb{Q}_0^+$ or $\mathbb{R}_0^+$ as carrier. The algorithm starts by selecting a parametric interpretation (e.g. linear, simple or simple-mixed polynomials). Then, three kinds of constraints are generated:

1. For each function symbol $f \in \mathcal{F}$, the constraint $f_\mathcal{A}(x_1, \dots, x_n) \geq 0$ ensures that the interpretation function only returns values from the carrier ($\mathbb{Q}_0^+$ or $\mathbb{R}_0^+$). If only nonnegative coefficients are used in the interpretation, then this condition is obviously fulfilled and these constraints may be dropped.

2. For each function symbol $f \in \mathcal{F}$ and every $i \in \{1, \dots, n\}$, we have the constraint $\frac{\partial f_\mathcal{A}(x_1,\dots,x_i,\dots,x_n)}{\partial x_i} \geq 1$. This ensures that Theorem 7.6 can be applied in order to certify monotonicity of all interpretation functions with respect to the orders $>_\delta$ for all $\delta \in \mathbb{R}^+$. We do not have to calculate the differentiation automatically. This can already be done manually on the general parametric interpretations. For example, for linear polynomials, we have for each n-ary function symbol f and each $i \in \{1, \dots, n\}$:

$$\frac{\partial f_\mathcal{A}(x_1, \dots, x_i, \dots, x_n)}{\partial x_i} = \frac{\partial d_f + \sum_{i=1}^n a_{f,i} x_i}{\partial x_i} = a_{f,i} \geq 1$$

3. For each rewrite rule $l \to r \in \mathcal{R}$, the polynomial $P_{l,r} = [\alpha]_\mathcal{A}(l) - [\alpha]_\mathcal{A}(r)$is generated. The constant coefficient of this polynomial has to be positive, and all other coefficients have to be nonnegative. This ensures that all rewrite rules are oriented from left to right. Furthermore, $\delta_{l,r}$ will then be the value of the constant coefficient of $P_{l,r}$. Therefore, if these constraints are fulfilled, we have $\delta_{l,r} \in \mathbb{R}^+$ for all rewrite rules $l \to r \in \mathcal{R}$, and thus $\delta \in \mathbb{R}^+$. This means that if all other constraints are fulfilled as well, then Theorem 7.7 can be applied in order to conclude termination of $\mathcal{R}$.

Any constraints that contain variables which are not coefficient variables can now be transformed into a number of smaller constraints which only contain coefficient variables. This is done in a similar way as the constraints in $\mathrm{rp}'_\mathcal{A}(\mathcal{R}) \cup \mathrm{nz}_\mathcal{A}(\mathcal{R}) \cup \mathrm{ve}_\mathcal{A}(\mathcal{R})$ are transformed into the constraints in $\mathrm{cp}_\mathcal{A}(\mathcal{R})$ in Definition 4.20. Solving the resulting constraints over the coefficient variables then induces a polynomial interpretation over $\mathbb{Q}_0^+$ or $\mathbb{R}_0^+$ which proves termination of $\mathcal{R}$.

In [13], Lucas has shown that polynomial interpretations over the rational and real numbers are strictly more powerful than polynomial interpretations over the natural numbers. An interesting question is how the power of polynomial interpretations over the reals relates to the power of context-dependent interpretations over the reals. However, by generalizing Lemma 3.9, we can see that context-dependent interpretations are at least as powerful as interpretations over the reals.

Lemma 7.8. *Suppose that we have an interpretation into a well-founded monotone algebra $\mathcal{B} = (\mathbb{R}_0^+, [\cdot]_\mathcal{B}, >_\delta)$ with $\delta \in \mathbb{R}^+$ which is compatible with a TRS $\mathcal{R}$. Construct an interpretation into the context-dependent algebra $\mathcal{C}$ as follows:*

$$f_\mathcal{C}[\Delta](a_1, \ldots, a_n) = \Delta f_\mathcal{B}(a_1\delta/\Delta, \ldots, a_n\delta/\Delta)/\delta$$
$$f_\mathcal{C}^i(\Delta) = \Delta$$

Then $\mathcal{C}$ fulfills the following properties:

1. *for all ground terms t, all assignments α, and all Δ-assignments α' such that $\alpha'[\Delta](x) = \Delta\alpha(x)/\delta$ for every variable x, the equality $[\alpha']_\mathcal{C}[\Delta](t) = \Delta[\alpha]_\mathcal{B}(t)/\delta$ holds*
2. *$\mathcal{C}$ is Δ-compatible with $\mathcal{R}$ with respect to the set of orders $\{>_\Delta \mid \Delta \in \mathbb{R}^+\}$ from Definition 3.7*
3. *$\mathcal{C}$ is Δ-monotone with respect to the set of orders $\{>_\Delta \mid \Delta \in \mathbb{R}^+\}$ from Definition 3.7*

Proof. Property 1 can be verified by structural induction on t. If t is a variable, then the property holds by the assumptions about α and α'. If t is a constant, then the property holds by the definition of $f_\mathcal{C}$, which concludes the base cases. If t has the shape $f(t_1, \ldots, t_n)$ where f is a function symbol with arity $n > 0$, then expanding the definitions of $f_\mathcal{C}$ and $f_\mathcal{C}^i$ yields:

$$\begin{aligned}
[\alpha']_\mathcal{C}[\Delta](\ldots, t_i, \ldots) &= f_\mathcal{C}[\Delta](\ldots, [\alpha']_\mathcal{C}[f_\mathcal{C}^i(\Delta)](t_i), \ldots) \\
&= f_\mathcal{C}[\Delta](\ldots, [\alpha']_\mathcal{C}[\Delta](t_i), \ldots) \\
&= f_\mathcal{C}[\Delta](\ldots, \Delta[\alpha]_\mathcal{B}(t_i)/\delta, \ldots) \\
&= \Delta f_\mathcal{B}(\ldots, \Delta[\alpha]_\mathcal{B}(t_i)\delta/(\Delta\delta), \ldots)/\delta \\
&= \Delta f_\mathcal{B}(\ldots, [\alpha]_\mathcal{B}(t_i), \ldots)/\delta \\
&= \Delta[\alpha]_\mathcal{B}(f(\ldots, t_i, \ldots))/\delta
\end{aligned}$$

The third line in this proof follows from the induction hypothesis, the rest is straightforward.

For property 2, we know that the interpretation into $\mathcal{B}$ is compatible with $\mathcal{R}$ with respect to the order $>_\delta$, therefore $[\alpha]_\mathcal{B}(l) - [\alpha]_\mathcal{B}(r) \geq \delta$ for every rewrite rule $l \to r$ in $\mathcal{R}$ and all assignments α. Furthermore, we can conclude $[\alpha']_\mathcal{C}[\Delta](l) - [\alpha']_\mathcal{C}[\Delta](r) = \Delta[\alpha]_\mathcal{B}(l)/\delta - \Delta[\alpha]_\mathcal{B}(r)/\delta \geq \Delta$ from property 1 whenever $\alpha(x) = \alpha'[\Delta](x)\delta/\Delta$ for all variables x. Since $\Delta[\alpha]_\mathcal{B}(l)/\delta - \Delta[\alpha]_\mathcal{B}(r)/\delta \geq \Delta$ holds for all assignments α because of $[\alpha]_\mathcal{B}(l) - [\alpha]_\mathcal{B}(r) \geq \delta$, we also have $[\alpha']_\mathcal{C}[\Delta](l) - [\alpha']_\mathcal{C}[\Delta](r) \geq \Delta$ for all Δ-assignments α'.

For property 3, we know that the interpretation into $\mathcal{B}$ is monotone with respect to the order $>_\delta$, so we have $f_\mathcal{B}(\ldots, b_i', \ldots) - f_\mathcal{B}(\ldots, b_i, \ldots) \geq \delta$ whenever $b_i' - b_i \geq \delta$. We need to show that $f_\mathcal{C}[\Delta](\ldots, a_i', \ldots) - f_\mathcal{C}[\Delta](\ldots, a_i, \ldots) \geq \Delta$ whenever $a_i' - a \geq \Delta$, which is equivalent to showing $\Delta f_\mathcal{B}(\ldots, a_i'\delta/\Delta, \ldots)/\delta -$

$\Delta f_{\mathcal{B}}(\ldots, a_i\delta/\Delta, \ldots)/\delta \geq \Delta$ or $f_{\mathcal{B}}(\ldots, a_i\delta/\Delta, \ldots) - f_{\mathcal{B}}(\ldots, a_i\delta/\Delta, \ldots) \geq \delta$. We know that $a_i' - a_i \geq \Delta$, which is equivalent to $a_i'\delta/\Delta - a_i\delta/\Delta \geq \delta$. Therefore, $f_{\mathcal{B}}(\ldots, a_i'\delta/\Delta, \ldots) - f_{\mathcal{B}}(\ldots, a_i\delta/\Delta, \ldots) \geq \delta$ follows from the monotonicity of $\mathcal{B}$ which holds by assumption. □

This concludes that context-dependent interpretations over the reals are at least as powerful at showing termination as interpretations into algebras over the real numbers are. Also, as can be seen from the construction, context-dependent interpretations over the reals which use only polynomial interpretation functions are at least as powerful as polynomial interpretations into the reals. What remains open is whether context-dependent interpretations into the reals are strictly more powerful as a termination criterion than interpretations into algebras over the real numbers. For interpretations which only use polynomials as interpretation functions, we conjecture that context-dependent interpretations are strictly more powerful. As shown in section 5.2, it is relatively easy to find termination proofs for non-simply terminating rewrite systems with context-dependent interpretations, even if we only use very simple polynomials in the interpretation functions. For polynomial interpretations into the reals, it is not even sure whether they can handle non-simple termination at all. In [13], we have seen the construction of rather complex rewrite systems which can be proved terminating by polynomial interpretations over the rationals or reals, but not over the natural numbers. A non-simply terminating rewrite system would have provided a much simpler solution to that problem. Also, concerning the estimation of upper bounds on the derivational complexity, interpretations into the rationals and reals still use similar rules as interpretations into the naturals. In particular, repeated application of monomials of the form $ax_1^{n_1} \cdot \ldots \cdot x_k^{n_k}$ still gives us an exponential in $x_1, \ldots, x_k$ if $a > 1$.

However, the relative power of interpretations into the reals comes at a price. As illustrated by the running example of Section 4.3, even for simple examples, the constraints can already get rather complicated. This makes it more difficult to move through the search space effectively. While this is a drawback for termination proving, where we already have other, faster methods available, we still think that this method does very well at what it was actually designed for, namely derivational complexity analysis.

8 Summary and Future Work

After a very general definition of context-dependent interpretations, we have seen a solution to Hofbauer's open question of implementing an automated search for these interpretations [9]. We have defined a subclass of context-dependent interpretations for which we can adapt a known algorithm [5] from polynomial interpretations for proving termination. We have shown that a further, easily distinguishable, subclass is able to certify a quadratic, and therefore polynomial, derivational complexity. Tests of our implementation of this algorithm on the current version of the TPDB [17] have shown that this subclass of a subclass is still quite powerful. Moreover, even though it looks comparatively tame, it is still able to handle non-simple termination.

However, there are still quite some open problems for future work. We have seen that parametric context-dependent interpretations over the reals with (restricted) Δ-simple polynomials behave very nicely when we look for context-dependent interpretations automatically. It would be interesting to investigate whether there are other parametric interpretations for which this is true. Also, we currently only have nice criteria for linear and quadratic derivational complexity. Are there good criteria to certify other subclasses of polynomial derivational complexity, like cubic or general polynomial derivational complexity? Furthermore, there are other methods ($\Pi(0)$-interpretations, LMPO, POP, and POP*) which can show termination in polynomial time. However, their notion of encoding functions computable in FP is different from the notion of polynomial derivational complexity that we used in this thesis. Finding connections between these two notions would help to apply these other methods to derivational complexity. Another avenue for further research in this area would be to investigate whether the bound induced by specific matrix interpretations on string rewrite systems [26] can be generalized to term rewriting. A further interesting question is the relationship between context-dependent interpretations and polynomial interpretations over the reals. As we have seen in Section 7.2, context-dependent interpretations are at least as strong as polynomial interpretations over the reals. However, are there concrete examples which can be proved terminating by context-dependent interpretations, but not by polynomial interpretations over the reals? Last, we want to mention the example at the end of Section 4.3 again. As we have seen in the example, the induction part in the termination proof by the context-dependent makes it impossible for our current implementation termination of this seemingly simple example automatically. If this obstacle could be overcome automatically, this could increase the power of our implementation even more.

Bibliography

[1] Toshiyasu Arai and Georg Moser. Proofs of termination of rewrite systems for polytime functions. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 3821 of *Lecture Notes in Computer Science*, pages 529–540. Springer-Verlag, 2005.

[2] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[3] Guillaume Bonfante, Adam Cichon, Jean-Yves Marion, and Hélène Touzet. Complexity classes and rewrite systems with polynomial interpretation. In *Proceedings of the 12th International Workshop on Computer Science Logic*, pages 372–384, London, UK, 1999. Springer-Verlag.

[4] Guillaume Bonfante, Adam Cichon, Jean-Yves Marion, and Hélène Touzet. Algorithms with polynomial interpretation termination proof. *Journal of Functional Programming*, 11(1):33–53, 2001.

[5] Evelyne Contejean, Claude Marché, Ana Paula Tomás, and Xavier Urbain. Mechanically proving termination using polynomial interpretations. *Journal of Automated Reasoning*, 34(4):325–363, 2005.

[6] MiniSat, a SAT solver,
`http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/`, 2007.

[7] Jürgen Giesl, Stephan Swiderski, Peter Schneider-Kamp, and René Thiemann. Automated termination analysis for Haskell: From term rewriting to programming languages. In Frank Pfenning, editor, *RTA'06: Proceedings of the 17th International Conference on Rewriting Techniques and Applications*, volume 4098 of *Lecture Notes in Computer Science*, pages 297–312. Springer, 2006.

[8] Dieter Hofbauer. Termination proofs by multiset path orderings imply primitive recursive derivation lengths. *Theoretical Computer Science*, 105(1):129–140, 1992.

[9] Dieter Hofbauer. Termination proofs by context-dependent interpretations. In *RTA'01: Proceedings of the 12th International Conference on Rewriting Techniques and Applications*, volume 2051 of *Lecture Notes in Computer Science*, pages 108–121, Berlin, 2001. Springer.

[10] Dieter Hofbauer and Clemens Lautemann. Termination proofs and the length of derivations. In *RTA '89: Proceedings of the 3rd International Conference on Rewriting Techniques and Applications*, volume 355 of *Lecture Notes in Computer Science*, pages 167–177, London, UK, 1989. Springer-Verlag.

[11] Ingo Lepper. Derivation lengths and order types of knuth-bendix orders. *Theoretical Computer Science*, 269(1-2):433–450, 2001.

[12] Salvador Lucas. Polynomials over the reals in proofs of termination: from theory to practice. *RAIRO Theoretical Informatics and Applications*, 39(3):547–586, 2005.

[13] Salvador Lucas. On the relative power of polynomials with real, rational, and integer coefficients in proofs of termination of rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 17(1):49–73, 2006.

[14] Salvador Lucas. Practical use of polynomials over the reals in proofs of termination. In *PPDP '07: Proceedings of the 9th ACM SIGPLAN international symposium on Principles and practice of declarative programming*, pages 39–50, New York, NY, USA, 2007. ACM Press.

[15] The Termination Problems Data Base, `http://www.lri.fr/~marche/tpdb/`, version 3.2, 2006.

[16] Results of the Termination Competition 2007, `http://www.lri.fr/~marche/termination-competition/2007/`, 2007.

[17] The Termination Problems Data Base, `http://www.lri.fr/~marche/tpdb/`, version 4.0, 2007.

[18] Jean-Yves Marion. Analysing the implicit complexity of programs. *Information and Computation*, 183(1):2–18, 2003.

[19] Mathematica, a computer algebra system, and MathLink, an interface for Mathematica, `http://www.wolfram.com/`, 2007.

[20] Yuri Matiyasevich. Enumerable sets are diophantine. *Soviet Mathematics (Dokladi)*, 11(2):354–357, 1970.

[21] Plogic library of TTT2, an automated termination prover, `http://cl-informatik.uibk.ac.at/ttt2/`, 2007.

[22] Aart Middeldorp. "Term Rewriting" lecture notes, Universität Innsbruck, Institute for Computer Science, 2006.

[23] Georg Moser. Derivational complexity of Knuth-Bendix orders revisited. In *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 4246 of *Lecture Notes in Computer Science*, pages 75–89, 2006.

[24] Joachim Steinbach. Proving polynomials positive. In *Proceedings of the 12th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 191–202, London, UK, 1992. Springer-Verlag.

[25] TeReSe. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[26] Johannes Waldmann. Automatic certification of polynomial derivation lengths in string rewriting, Workshop on Proof Theory and Rewriting, `http://www.imn.htwk-leipzig.de/~waldmann/talk/06/rpt/poly/`, 2006.

[27] Andreas Weiermann. Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths. *Theoretical Computer Science*, 139(1-2):355–362, 1995.

[28] Sarah Winkler. A Diophantine constraint solver, Bachelor Thesis, Universität Innsbruck, Institute for Computer Science, Computational Logic, 2006.