Beyond Polynomials and Peano Arithmetic – Automation of Elementary and Ordinal Interpretations

Harald Zankl, Sarah Winkler, Aart Middeldorp

Institute of Computer Science, University of Innsbruck, 6020 Innsbruck, Austria

Abstract

Kirby and Paris (1982) proved in a celebrated paper that a theorem of Goodstein (1944) cannot be established in Peano arithmetic. We present an encoding of Goodstein's theorem as a termination problem of a finite rewrite system. Using a novel implementation of algebras based on ordinal interpretations, we are able to automatically prove termination of this system, resulting in the first automatic termination proof for a system whose derivational complexity is not multiple recursive. Our method can also cope with the encoding by Touzet (1998) of the battle of Hercules and Hydra as well as a (corrected) encoding by Beklemishev (2006) of the Worm battle, two further systems which have been out of reach for automatic tools, until now. Based on our ideas of implementing ordinal algebras we also present a new approach for the automation of elementary interpretations for termination analysis.

Key words: term rewriting, termination, automation, ordinals

1. Introduction

Since the beginning of the millennium there has been much progress regarding automated termination tools for rewrite systems. [1] Despite the many different techniques that have been developed, it seems that (terminating) TRSs which admit very long derivations are out of reach even for the most powerful tools. This is not surprising since many base methods induce rather small upper bounds on the derivational complexity, which is a function that bounds the length of the longest possible derivation (rewrite sequence) by the size of its starting term. Hofbauer and Lautemann (1989) have shown that polynomial interpretations are limited to double exponential derivational complexity. They

Email addresses: `harald.zankl@uibk.ac.at` (Harald Zankl), `sarah.winkler@uibk.ac.at` (Sarah Winkler), `aart.middeldorp@uibk.ac.at` (Aart Middeldorp).

[1] `http://www.termination-portal.org/`

further showed that the derivational complexity of a rewrite system compatible with the Knuth-Bendix order (KBO) cannot be bounded by a primitive recursive function. Later, Lepper (2001) established the Ackermann function as an upper bound for KBO, whereas Weiermann (1995) proved a multiple recursive upper bound for the lexicographic path order (LPO). More recently, Moser and Schnabl (2011); Schnabl (2012) have studied upper bounds on the complexity when using these base methods in the dependency pair framework. Although dependency pairs significantly increase termination proving power, from the viewpoint of derivational complexity the limit is still multiple recursive. This has led to the conjecture (Schnabl, 2012, Conjecture 6.99) that for any system whose termination can be proved automatically by modern tools the length of its derivations can be bounded by a multiple recursive function (in the size of the starting terms).

Ordinals have been used in termination arguments for many decades (e.g., Turing (1949); Gentzen (1936)). In fact ordinals are essential to prove termination of the battle of Hercules and Hydra (also due to Kirby and Paris (1982)), or the sequences associated with Goodstein's theorem since these derivations cannot be bounded by a multiple recursive function (Cichon (1983)). Although TRS encodings of the Hydra battle are known for many years (e.g., by Touzet (1998)), they could so far not be handled by automatic termination tools, witnessing Schnabl's conjecture. Indeed a successful implementation of ordinals for automatic termination proofs is still lacking. Very recently, Urban and Miné (2014) presented an approach to conclude termination of imperative programs by inferring ordinal-valued ranking functions. Here ordinals are essential to handle nondeterminism, though only ordinals below $\omega^{\omega^{\omega}}$ are involved and hence the ranking functions are still multiple recursive. The theorem prover Vampire uses ordinal numbers (see (Kovács et al., 2011, Section 7)) in its implementation of KBO but only for weights of predicate symbols. Since these symbols occur only at the root of atomic expressions no ordinal arithmetic is needed but only comparison of ordinals.

In this article we first encode the computation of Goodstein sequences (see Theorem 9) as a rewrite system $\mathcal{G}$ such that termination of $\mathcal{G}$ implies Goodstein's theorem. Since these sequences cannot be bounded by a multiple recursive function, this also holds for the derivational complexity of $\mathcal{G}$. After presenting this motivating example, we discuss automation of a termination criterion based on ordinal interpretations which is capable of proving $\mathcal{G}$ terminating, thereby overcoming the limitations alleged by the above conjecture. Our implementation can also cope with Touzet's encoding (Touzet, 1998) of the battle of Hercules and Hydra, as well as a (corrected) encoding of the Worm battle (Beklemishev, 2006).

Automation of ordinal interpretations is challenging since ordinal arithmetic does, e.g., not satisfy commutativity. Hence in contrast to polynomial interpretations terms do not evaluate to expressions of a canonical shape. We tackle this deficiency by introducing approximations which yield expressions of a special shape. Approximations (albeit less involved) have already been used for polynomial interpretations with negative (Hirokawa and Middeldorp, 2004; Fuhs et al., 2007) or irrational (Zankl and Middeldorp, 2010) coefficients. In preliminary work Zankl et al. (2012); Winkler et al. (2012) already used ordinal domains to increase automatic termination proving power. However, in Zankl et al. (2012) the focus is on string rewriting and the interpretation functions have a very limited shape to avoid ordinal arithmetic. As a consequence the method is limited to systems with at most multiple *exponential* derivational complexity. Similarly, Winkler et al. (2012) use ordinal domains for generalized KBO, again for string rewriting only.

In the respective implementation, function symbol weights are moreover below ω^ω. We anticipate that our treatment of arithmetic for ordinals up to ϵ_0 could improve some of the results from Kovács et al. (2011); Winkler et al. (2012); Urban and Miné (2014).

Lescanne (1995) proposed elementary functions for proving (AC-)termination but his implementation is limited to checking the orientation of rules for *given* interpretations. Lucas (2009) considers so-called linear elementary interpretations (LEIs) of the shape $A(\overline{x}) + B(\overline{x})^{C(\overline{x})}$ where $A(\overline{x})$, $B(\overline{x})$, and $C(\overline{x})$ are linear polynomials. Furthermore, he proposes an approach based on rewriting, constraint logic programming (CLP), and constraint satisfaction problems (CSPs) to also *find* suitable interpretation functions. He leaves an actual implementation of his method as future work and mentions the need for heuristics to achieve an efficient implementation. In this article we propose a different shape of interpretation functions because LEIs are neither closed under (scalar) multiplication, addition, nor composition. Furthermore, the motivating example in Lucas (2009) (which is a simplified version of the leading example in Lescanne (1995)), uses a non-linear (elementary) interpretation for multiplication. We show that also an implementation of algebras with elementary interpretations can take advantage from an approximation-based approach. These findings are related to Problem #28 in the RTA List of Open Problems,[2] which asks to "develop effective methods to decide whether a system decreases with respect to some exponential interpretation". Our contribution is restricted to a subclass of elementary interpretations but also admits the search for suitable interpretations.

This article is organized as follows. In the next section we recall ordinal arithmetic and weakly monotone algebras for termination proofs. In Section 3 we present our encoding of Goodstein's theorem and prove its correctness. Section 4 discusses how ordinal algebras can be automated and applies the approach to several rewrite systems (some of them encoding the Hydra battle), where also the limitations of our method become apparent. Likewise, Section 5 adapts the approach to elementary interpretations. Experimental results are the topic of Section 6. We conclude in Section 7.

This article is an updated and extended version of Winkler et al. (2013). In particular, the extension to elementary interpretations (Section 5) and the experimental evaluation (Section 6) are new. Furthermore, in Section 4 the approximation $+_\mu$ has been refined (to succeed on the Worm battle) while tiny flaws in the approximations $+_\nu$ and $\oplus_\nu$ have been corrected (cf. Definition 22).

2. Preliminaries

We recall some preliminaries about ordinal numbers. Ordinals are transitive sets well-ordered with respect to $\in$. Hence $\alpha < \beta$ if and only if $\alpha \in \beta$. By identifying $\varnothing$, $\{\varnothing\}$, $\{\varnothing, \{\varnothing\}\}$, $\ldots$ with 0, 1, 2, $\ldots$, the natural numbers are embedded in the ordinals. If α is an ordinal then the ordinal $\alpha \cup \{\alpha\}$ is its successor, denoted by $\alpha + 1$. An ordinal β constitutes a successor ordinal if there is some α such that $\beta = \alpha + 1$, otherwise β is called a limit ordinal. For instance 1, 2, 3, $\ldots$ are successor ordinals, whereas 0 and the smallest infinite ordinal ω are limit ordinals. The latter is equivalent to the set of all natural numbers. The following ordinal arithmetic operations constitute extensions of the respective operations on natural numbers (see Just and Weese (1996) for details).

[2] `http://www.win.tue.nl/rtaloop/`

Definition 1. For ordinals α and β their *sum* $\alpha + \beta$ is defined by recursion over β as (a) $\alpha + 0 = \alpha$, (b) $\alpha + \beta = (\alpha + \gamma) + 1$ if $\beta = \gamma + 1$, and (c) $\alpha + \beta = \bigcup_{\gamma < \beta} \alpha + \gamma$ if β is a non-zero limit ordinal.

Addition satisfies associativity $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$ but is not commutative, e.g., $1 + \omega = \omega \neq \omega + 1$.

Definition 2. For ordinals α and β their *product* $\alpha \cdot \beta$ is defined by recursion over β as (a) $\alpha \cdot 0 = 0$, (b) $\alpha \cdot \beta = \alpha \cdot \gamma + \alpha$ if $\beta = \gamma + 1$, and (c) $\alpha \cdot \beta = \bigcup_{\gamma < \beta} \alpha \cdot \gamma$ if β is a non-zero limit ordinal.

Since $2 \cdot \omega = \omega \neq \omega \cdot 2$ multiplication is not commutative, and as $(\omega + 1) \cdot 2 = (\omega + 1) + (\omega + 1) = \omega + \omega + 1 = \omega \cdot 2 + 1$ also not right-distributive, but associativity $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$ and left-distributivity $\alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma)$ hold. We mostly write αa for $\alpha \cdot a$ whenever α is an ordinal and a a finite ordinal, i.e., $a < \omega$.

Definition 3. For ordinals α and β, recursion over β allows to define *exponentiation* α^β as follows: (a) $\alpha^0 = 1$, (b) $\alpha^\beta = \alpha^\gamma \cdot \alpha$ if $\beta = \gamma + 1$, and (c) $\alpha^\beta = \bigcup_{\gamma < \beta} \alpha^\gamma$ if β is a non-zero limit ordinal.

Examples of infinite ordinals include $\omega^1 = \omega$, $\omega\, 3 = \omega + \omega + \omega$, $\omega^2 = \omega \cdot \omega$, $\omega^{\omega + 1}$, and ω^{ω^ω}. The ordinal ϵ_0 is the smallest ordinal α which satisfies $\alpha^\omega = \alpha$. Let $\mathbb{O}$ denote the class of ordinal numbers smaller than ϵ_0, $\mathbb{N}$ the ordinal numbers smaller than ω (the natural numbers), $>$ the standard order on ordinals, and $\geqslant$ its reflexive closure.

Recall that every ordinal $\alpha < \epsilon_0$ can be represented in Cantor normal form (CNF), i.e.,

$$\alpha = \omega^{\alpha_1} a_1 + \cdots + \omega^{\alpha_n} a_n \tag{1}$$

such that $\alpha_1 > \cdots > \alpha_n$ are in CNF as well and $a_1, \ldots, a_n \in \mathbb{N}_{>0}$. The ordinal 0 is represented as the empty sum.

Definition 4. Let $\alpha = \omega^{\alpha_1} a_1 + \cdots + \omega^{\alpha_n} a_n$ and $\beta = \omega^{\beta_1} b_1 + \cdots + \omega^{\beta_m} b_m$ be ordinals in CNF, and $\{\gamma_1, \ldots, \gamma_k\} = \{\alpha_1, \ldots, \alpha_n\} \cup \{\beta_1, \ldots, \beta_m\}$ such that $\gamma_1 > \cdots > \gamma_k$. The *natural sum* of α and β is defined as $\alpha \oplus \beta = \omega^{\gamma_1}(a'_1 + b'_1) + \cdots + \omega^{\gamma_k}(a'_k + b'_k)$ where $a'_i = a_j$ ($b'_i = b_j$) if $\gamma_i = \alpha_j$ ($\gamma_i = \beta_j$) for some j, and $a'_i = 0$ ($b'_i = 0$) otherwise.

In contrast to standard addition, natural addition on ordinals enjoys all properties known from addition on natural numbers, e.g., $2 \oplus \omega = \omega \oplus 2 = \omega + 2$. For ordinal algebras as considered later in this article we rely critically on the fact that addition, natural addition, multiplication, and exponentiation are weakly monotone in both arguments.

We assume familiarity with term rewriting and termination in particular (Baader and Nipkow, 1998; TeReSe, 2003).

The derivation height of a term t with respect to a well-founded and finitely branching rewrite relation $\to_{\mathcal{R}}$ is defined as $\mathrm{dh}_{\mathcal{R}}(t) = \max\{m \mid \exists u\ t \to_{\mathcal{R}}^m u\}$. The derivational complexity of $\mathcal{R}$ computes the maximal derivation height of all terms up to size n and is defined as $\mathrm{dc}_{\mathcal{R}}(n) = \max\{\mathrm{dh}_{\mathcal{R}}(t) \mid |t| \leqslant n\}$.

A relative TRS $\mathcal{R}/\mathcal{S}$ is a pair of TRSs $\mathcal{R}$ and $\mathcal{S}$ with the induced rewrite relation $\to_{\mathcal{R}/\mathcal{S}} = \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}} \cdot \to_{\mathcal{S}}^*$.

We consider well-founded algebras $\mathcal{A}$ with interpretation functions $f_\mathcal{A}$. An interpretation function $f_\mathcal{A}$ is *simple* if $f_\mathcal{A}(a_1, \ldots, a_n) \geqslant a_i$ for all $1 \leqslant i \leqslant n$. It is *monotone* if $a > b$ implies $f_\mathcal{A}(\ldots, a_{i-1}, a, a_{i+1}, \ldots) > f_\mathcal{A}(\ldots, a_{i-1}, b, a_{i+1}, \ldots)$ and *weakly monotone* if $a > b$ implies $f_\mathcal{A}(\ldots, a_{i-1}, a, a_{i+1}, \ldots) \geqslant f_\mathcal{A}(\ldots, a_{i-1}, b, a_{i+1}, \ldots)$. An algebra is *simple/monotone/weakly monotone* if all its interpretation functions are simple/monotone/weakly monotone. An assignment α maps variables to values in the carrier of $\mathcal{A}$. By $[\alpha]_\mathcal{A}(t)$ we denote the interpretation of the term t based on the assignment α. A TRS $\mathcal{R}$ is *compatible* with an algebra $\mathcal{A}$ if $[\alpha]_\mathcal{A}(\ell) > [\alpha]_\mathcal{A}(r)$ for every $\ell \to r \in \mathcal{R}$ and assignment α (also written $\mathcal{R} \subseteq >_\mathcal{A}$). Algebras may yield termination proofs.

Theorem 5. *A TRS is terminating if and only if it is compatible with a well-founded monotone algebra.* $\square$

Theorem 6 (Touzet (1998); Zantema (2001))**.** *A TRS is terminating if it is compatible with a well-founded weakly monotone simple algebra.* $\square$

3. The Goodstein Sequence

In this section we present a TRS for the Goodstein sequence, whose definition requires the following key notion. Given $n > 1$, a natural number α is in *hereditary base n representation*, which we indicate by writing $(\alpha)_n$, if

$$(\alpha)_n = n^{(\alpha_k)_n} \cdot a_k + n^{(\alpha_{k-1})_n} \cdot a_{k-1} + \cdots + n^{(\alpha_0)_n} \cdot a_0 \tag{2}$$

such that $(\alpha_k)_n > \cdots > (\alpha_0)_n$ are in hereditary base n representation and $0 < a_i < n$ for all $0 \leqslant i \leqslant k$. For $m > n$ we denote by $(\alpha)_n^m$ the result of replacing n by m in $(\alpha)_n$, so $(\alpha)_n^m = m^{(\alpha_k)_n^m} \cdot a_k + m^{(\alpha_{k-1})_n^m} \cdot a_{k-1} + \cdots + m^{(\alpha_1)_n^m} \cdot a_1 + a_0$ is in hereditary base m representation.

For instance, $(1)_2 = 2^0 \cdot 1$, where we drop the coefficient 1 and simply write $(1)_2 = 2^0$. Moreover, $(2)_2 = 2^1 = 2^{2^0}$ and $(5)_2 = 2^2 + 1 = 2^{2^{2^0}} + 2^0$, whereas $(5)_2^3 = 3^{3^{3^0}} + 3^0 = 28$.

Definition 7. The *Goodstein sequence* g_α with starting value α is defined by $g_\alpha(0) = \alpha$ and $g_\alpha(i+1) = (g_\alpha(i))_{i+2}^{i+3} - 1$ for all $i \geqslant 0$.

Example 8. For $\alpha = 2$ the Goodstein sequence yields

$$\begin{aligned}
&g_2(0) = 2\\
&g_2(1) = (2)_2^3 - 1 = (2^1)_2^3 - 1 = 3^1 - 1 = 2\\
&g_2(2) = (2)_3^4 - 1 = 2 - 1 = 1\\
&g_2(3) = (1)_4^5 - 1 = 1 - 1 = 0
\end{aligned}$$

while for $\alpha = 5$ we obtain

$$\begin{aligned}
&g_5(0) = 5\\
&g_5(1) = (5)_2^3 - 1 = (2^2 + 2^0)_2^3 - 1 = 3^3 + 3^0 - 1 = 27\\
&g_5(2) = (27)_3^4 - 1 = (3^3)_3^4 - 1 = 4^4 - 1 = 255\\
&g_5(3) = (255)_4^5 - 1 = (4^3 \cdot 3 + 4^2 \cdot 3 + 4 \cdot 3 + 3)_4^5 - 1 = 5^3 \cdot 3 + 5^2 \cdot 3 + 5 \cdot 3 + 2 = 467
\end{aligned}$$

Theorem 9 (Goodstein (1944)). *For all α there exists a k such that $g_\alpha(k) = 0$.* $\square$

By $G(\alpha)$ we denote the smallest number k with this property. Totality of this function is not provable in Peano arithmetic, as shown by Kirby and Paris (1982). Cichon (1983) presented a very short proof using results concerning recursion theoretic hierarchies of functions. In particular, he showed that the growth rate of G cannot be bounded by any H_α such that $\alpha < \epsilon_0$. [3]

Definition 10. For all $n > 1$ we define a mapping $[\cdot]_n$ to represent natural numbers in (hereditary) base n as ground terms over $\{\mathsf{c}, \mathsf{0}\}$, where c is a binary function symbol and $\mathsf{0}$ a constant. Let $(\alpha)_n$ be a natural number in hereditary base n representation as in (2). The term $\mathsf{c}(x, \mathsf{c}(x, \cdots \mathsf{c}(x, y) \cdots))$ containing $k \geqslant 0$ occurrences of c is denoted $\mathsf{c}^k(x, y)$. In particular, $\mathsf{c}^0(x, y) = y$. Then $[\cdot]_n$ is recursively defined such that $[0]_n = \mathsf{0}$ and

$$[\alpha]_n = \mathsf{c}^{a_0}([\alpha_0]_n, \ldots \mathsf{c}^{a_{k-1}}([\alpha_{k-1}]_n, \mathsf{c}^{a_k}([\alpha_k]_n, \mathsf{0})) \ldots)$$

Intuitively, given base n, the term $\mathsf{c}([\alpha]_n, [\beta]_n)$ represents the number $n^\alpha + \beta$, and terms contributing to the base n representation of a number are combined in increasing order. This is in contrast to (2), where terms are sorted in a decreasing way.

Example 11. For $(1)_2 = 2^0$ we have $[1]_2 = \mathsf{c}(\mathsf{0}, \mathsf{0})$, for $(2)_2 = 2^{2^0}$ we have $[2]_2 = \mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{0})$, for $(7)_2 = 2^{2^{2^0}} + 2^{2^0} + 2^0$ we have $[7]_2 = \mathsf{c}(\mathsf{0}, \mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{c}(\mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{0}))))$, and for $(7)_3 = 3^{3^0} \cdot 2 + 3^0$ we have $[7]_3 = \mathsf{c}(\mathsf{0}, \mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{0})))$. Note that different numbers over different bases might be represented by the same term, for instance $(2)_2 = (3)_3 = \mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{0})$.

The following TRS $\mathcal{G}$ works on inputs of the form $[\cdot]_n$ to model g_α. Its definition is inspired by Touzet's encoding of the Hydra battle (Touzet, 1998) (see Example 30).

Definition 12. Consider the following TRS $\mathcal{G}$ over a signature consisting of unary function symbols $\bullet$, $[\!]$, $\circ$ and binary function symbols f, h, in addition to $\mathsf{0}$ and c:

$$\begin{aligned}
[\!] \circ x &\to \circ [\!] \, x && \text{(A1)}\\
\bullet [\!] \, x &\to [\!] \bullet \bullet x && \text{(A2)}\\
\circ x &\to \bullet [\!] \, x && \text{(A3)}\\
\mathsf{c}(\mathsf{0}, x) &\to \circ x && \text{(B1)}\\
\bullet \mathsf{c}(\mathsf{c}(x, y), z) &\to \bullet \mathsf{f}(\mathsf{c}(x, y), z) && \text{(B2)}\\
\bullet \mathsf{f}(\mathsf{0}, x) &\to \circ x && \text{(C1)}\\
\bullet \mathsf{f}(\mathsf{c}(x, y), z) &\to \mathsf{h}(\bullet \mathsf{f}(x, y), \bullet \bullet \mathsf{f}(\mathsf{f}(x, y), z)) && \text{(C2)}\\
\bullet \mathsf{h}(x, y) &\to \mathsf{h}(\bullet x, \bullet \bullet \mathsf{c}(x, y)) && \text{(D1)}\\
\mathsf{h}(x, y) &\to \circ y && \text{(D2)}\\
\bullet \mathsf{f}(x, y) &\to \mathsf{f}(\bullet x, y) && \text{(E1)}\\
\bullet \mathsf{c}(x, y) &\to \mathsf{c}(\bullet x, \bullet y) && \text{(E2)}\\
\bullet x &\to x && \text{(E3)}\\
\circ x &\to x && \text{(E4)}
\end{aligned}$$

[3] Here H is the Hardy function: $H_0(n) = n + 1$, $H_{\alpha+1}(n) = H_\alpha(n + 1)$, and $H_\lambda(n) = H_{\lambda_n}(n)$ for a limit ordinal λ which is the supremum of an ordinal sequence $(\lambda_n)_{n \in \mathbb{N}}$.

The basic idea of the encoding is to perform a step in the Goodstein sequence as follows. The current base n and sequence element α are encoded as $\bullet\,[\!]^n\,[\alpha]_n$. Using rule (A2), the symbol $\bullet$ is repeatedly duplicated while moving over the $[\!]$'s, until a term of the form $[\!]^n\,\bullet^{2^n}\,[\alpha]_n$ is reached. The copies of $\bullet$ can move to places (using rules (E1) and (E2)) in $[\alpha]_n$ where changes are required to turn $[\alpha]_n$ into $[\beta]_{n+1}$ for $\beta = (\alpha)_n^{n+1} - 1$. This is achieved using rules (B1) – (D2) and the symbols f and h for auxiliary purposes, and produces at least one $\circ$ symbol which can then travel back up the $[\!]$'s using (A1). Finally, the base is increased by (A3), which yields a term $\bullet\,[\!]^{n+1}\,[\beta]_{n+1}$. Whenever there are too many $\bullet$ or $\circ$ symbols, they are removed with rules (E3) and (E4).

This idea is made precise in the following theorem, according to which $\mathcal{G}$ simulates for any starting value the computation of the Goodstein sequence. In particular, termination of $\mathcal{G}$ (Theorem 15) enforces for any $\alpha \in \mathbb{N}$ the existence of a k with $\bullet\,[\!]^2\,[\alpha]_2 \to_{\mathcal{G}}^* \bullet\,[\!]^k\,[0]_k$, and thus implies Theorem 9.

Theorem 13. *Let $\alpha, n \in \mathbb{N}$ such that $\alpha > 0$ and $n > 1$. Then $\bullet\,[\!]^n\,[\alpha]_n \to_{\mathcal{G}}^+ \bullet\,[\!]^{n+1}\,[\beta]_{n+1}$ where $\beta = (\alpha)_n^{n+1} - 1$.*

The proof of this result requires some auxiliary facts about $\mathcal{G}$.

Lemma 14.
(a) $\bullet^n\,\mathsf{h}(s,t) \to_{\mathcal{G}}^+ \circ\,\mathsf{c}^n(\bullet^n\,s, \bullet^{2n}\,t)$ *for all terms s and t.*
(b) *Let $\alpha, \beta \in \mathbb{N}$ and $n \in \mathbb{N}$ such that $n > 1$, $\beta + n^\alpha$ is positive, $s = [\alpha]_n$ and $t = [\beta]_n$. Then $\bullet^n\,\mathsf{f}(s,t) \to_{\mathcal{G}}^+ \circ\,u$ where $u = [\beta + n^\alpha - 1]_n$.*

Proof.
(a) By induction on n. If $n = 0$ then $\mathsf{h}(s,t) \to_{\mathcal{G}} \circ\,t$ in a single step using (D2). If $n > 0$ then

$$\begin{aligned}
\bullet^{n+1}\,\mathsf{h}(s,t) &\to_{\mathcal{G}} \bullet^n\,\mathsf{h}(\bullet\,s, \bullet\,\bullet\,\mathsf{c}(s,t)) && \text{(D1)}\\
&\to_{\mathcal{G}}^+ \circ\,\mathsf{c}^n(\bullet^{n+1}\,s, \bullet^{2(n+1)}\,\mathsf{c}(s,t)) && (\star)\\
&\to_{\mathcal{G}}^+ \circ\,\mathsf{c}^n(\bullet^{n+1}\,s, \mathsf{c}(\bullet^{2(n+1)}\,s, \bullet^{2(n+1)}\,t)) && \text{(E2)}\\
&\to_{\mathcal{G}}^+ \circ\,\mathsf{c}^n(\bullet^{n+1}\,s, \mathsf{c}(\bullet^{n+1}\,s, \bullet^{2(n+1)}\,t)) && \text{(E3)}\\
&= \circ\,\mathsf{c}^{n+1}(\bullet^{n+1}\,s, \bullet^{2(n+1)}\,t)
\end{aligned}$$

where $(\star)$ applies the induction hypothesis.
(b) By induction on α. If $\alpha = 0$ then $[\alpha]_n = \mathsf{0}$ and $\bullet^n\,\mathsf{f}(\mathsf{0},t) \to_{\mathcal{G}} \bullet^{n-1} \circ t \to_{\mathcal{G}}^* \circ\,t$ using rules (C1) and (E3). Since $\beta + n^0 - 1 = \beta$ and $t = [\beta]_n$ the claim holds. If $\alpha > 0$ then $[\alpha]_n = \mathsf{c}(s',t')$ and $s' = [\gamma]_n$ and $t' = [\delta]_n$ for some $\gamma, \delta \in \mathbb{N}$, so $\alpha = \delta + n^\gamma$. We have

$$\begin{aligned}
\bullet^n\,\mathsf{f}(\mathsf{c}(s',t'),t) &\to_{\mathcal{G}} \bullet^{n-1}\,\mathsf{h}(\bullet\mathsf{f}(s',t'), \bullet\,\bullet\,\mathsf{f}(\mathsf{f}(s',t'),t)) && \text{(C2)}\\
&\to_{\mathcal{G}}^+ \circ\,\mathsf{c}^{n-1}(\bullet^n\,\mathsf{f}(s',t'), \bullet^{2n}\,\mathsf{f}(\mathsf{f}(s',t'),t)) && \text{(a)}\\
&\to_{\mathcal{G}}^* \circ\,\mathsf{c}^{n-1}(\bullet^n\,\mathsf{f}(s',t'), \bullet^n\,\mathsf{f}(\bullet^n\,\mathsf{f}(s',t'),t)) && \text{(E1)}\\
&\to_{\mathcal{G}}^+ \circ\,\mathsf{c}^{n-1}(\circ\,w, \bullet^n\,\mathsf{f}(\circ\,w,t)) && (\star)\\
&\to_{\mathcal{G}}^+ \circ\,\mathsf{c}^{n-1}(w, \bullet^n\,\mathsf{f}(w,t)) && \text{(E4)}\\
&\to_{\mathcal{G}}^+ \circ\,\mathsf{c}^{n-1}(w, \circ\,w') && (\star\star)\\
&\to_{\mathcal{G}} \circ\,\mathsf{c}^{n-1}(w, w') && \text{(E4)}
\end{aligned}$$

where in ($\star$) we apply the induction hypothesis since $\gamma < \alpha$ and so we obtain a term $w = [\delta + n^\gamma - 1]_n$. Since $\delta + n^\gamma - 1 < \alpha$, we can apply the induction hypothesis again in ($\star\star$), which yields a term w' such that $w' = \left[\beta + n^{\delta+n^\gamma-1} - 1\right]_n$. Let $\nu = \delta + n^\gamma - 1$. For the term $v = \mathsf{c}^{n-1}(w, w')$ we thus have

$$v = [\beta + n^\nu \cdot (n-1) + n^\nu - 1]_n = \left[\beta + n^{\nu+1} - 1\right]_n = [\beta + n^\alpha - 1]_n \qquad \square$$

Proof of Theorem 13. Since $\alpha > 0$, we have $[\alpha]_n = \mathsf{c}(s, t)$ for some terms s and t. We apply case analysis on s. If $s = \mathsf{0}$ then $t = [\alpha - 1]_n$ and we have

$$\begin{aligned}
\bullet \Box^n \mathsf{c}(\mathsf{0}, t) &\to_\mathcal{G} \Box^n \mathsf{c}(\mathsf{0}, t) && \text{(E3)}\\
&\to_\mathcal{G} \Box^n \circ t && \text{(B1)}\\
&\to_\mathcal{G}^+ \circ \Box^n t && \text{(A1)}\\
&\to_\mathcal{G} \bullet \Box^{n+1} t && \text{(A3)}
\end{aligned}$$

Otherwise, $s = \mathsf{c}(u, v)$ so let $\mathsf{c}(u, v) = [\gamma]_n$ and $t = [\delta]_n$ for some $\gamma, \delta \in \mathbb{N}$. There is the following rewrite sequence:

$$\begin{aligned}
\bullet \Box^n \mathsf{c}(\mathsf{c}(u, v), t) &\to_\mathcal{G}^+ \Box^n \bullet^{2^n} \mathsf{c}(\mathsf{c}(u, v), t) && \text{(A2)}\\
&\to_\mathcal{G}^* \Box^n \bullet^{n+1} \mathsf{c}(\mathsf{c}(u, v), t) && \text{(E3)}\\
&\to_\mathcal{G}^* \Box^n \bullet^{n+1} \mathsf{f}(\mathsf{c}(u, v), t) && \text{(B2)}\\
&\to_\mathcal{G}^+ \Box^n \circ w && (\star)\\
&\to_\mathcal{G}^+ \circ \Box^n w && \text{(A1)}\\
&\to_\mathcal{G} \bullet \Box^{n+1} w && \text{(A3)}
\end{aligned}$$

where ($\star$) applies Lemma 14(b), according to which $w = [\delta + (n+1)^\gamma - 1]_{n+1}$. $\square$

Theorem 15. *The TRS $\mathcal{G}$ is terminating.*

Proof. We show termination of $\mathcal{G}$ by employing Theorem 6. Consider the following algebra $\mathcal{A}$ over the well-founded domain $\mathbb{O} \times \mathbb{N} \times \mathbb{N}$:

$$\begin{aligned}
\mathsf{0}_\mathcal{A} &= (0, 0, 0) & \Box_\mathcal{A}(x, m, n) &= (x, 2m+2, n)\\
\mathsf{c}_\mathcal{A}((x, m, n), (y, k, l)) &= (\omega^x \oplus y + 1, 0, 0) & \circ_\mathcal{A}(x, m, n) &= (x, 2m+3, n)\\
\mathsf{f}_\mathcal{A}((x, m, n), (y, k, l)) &= (\omega^x \oplus y, 0, 0) & \bullet_\mathcal{A}(x, m, n) &= (x, m, n+m+1)\\
\mathsf{h}_\mathcal{A}((x, m, n), (y, k, l)) &= (y + \omega^{x+1}, 0, 0)
\end{aligned}$$

Note that $\mathcal{A}$ is simple and weakly monotone, and it strictly orients all rules of $\mathcal{G}$:

$$\begin{aligned}
(x, 4m+8, n) &> (x, 4m+7, n) && \text{(A1)}\\
(x, 2m+2, 2m+n+3) &> (x, 2m+2, 2m+n+2) && \text{(A2)}\\
(x, 2m+3, n) &> (x, 2m+2, n+2m+3) && \text{(A3)}\\
(x+2, 0, 0) &> (x, 2m+3, n) && \text{(B1)}\\
(\omega^{\omega^x \oplus y+1} \oplus z + 1, 0, 1) &> (\omega^{\omega^x \oplus y+1} \oplus z, 0, 1) && \text{(B2)}\\
(x+1, 0, 1) &> (x, 2m+3, n) && \text{(C1)}\\
(\omega^{\omega^x \oplus y+1} \oplus z, 0, 1) &> (z + \omega^{\omega^x \oplus y+1}, 0, 0) && \text{(C2)}\\
(y + \omega^{x+1}, 0, 1) &> (y + \omega^{x+1}, 0, 0) && \text{(D1)}
\end{aligned}$$

$$(y + \omega^{x+1}, 0, 0) > (y, 2k + 3, l) \tag{D2}$$

$$(\omega^x \oplus y, 0, 1) > (\omega^x \oplus y, 0, 0) \tag{E1}$$

$$(\omega^x \oplus y + 1, 0, 1) > (\omega^x \oplus y + 1, 0, 0) \tag{E2}$$

$$(x, m, n + m + 1) > (x, m, n) \tag{E3}$$

$$(x, 2m + 3, n) > (x, m, n) \tag{E4}$$

Hence $\mathcal{G}$ is terminating. Note that rule (C2) has a weak decrease in its first component since ordinal addition might consume its left argument but natural addition does not, i.e., $\alpha \oplus \beta = \beta \oplus \alpha \geqslant \beta + \alpha$ for all ordinals α and β in CNF. $\square$

The proof of Theorem 15 (again inspired by the termination proof in Touzet (1998)) lexicographically combines ordinal with linear polynomial interpretations. However, we remark that weak monotonicity of the lexicographic product does not follow from weak monotonicity of the single interpretations (cf. Example 26). Still, the search for suitable interpretation functions can be automated (see Section 4.2.2).

4. Automation of Ordinal Algebras

In order to automate the search for suitable ordinal interpretations, we restrict ourselves to interpretation functions of a certain shape (see Definition 16). In Section 4.1 we show how for a given algebra with interpretation functions of this shape one can encode whether the interpretation of one term is larger than that of another term. In contrast to other termination criteria, ordinal arithmetic (non-commutative, expressions may be consumed) significantly complicates the encoding. Section 4.2 elaborates on implementation issues needed for a successful automation, where we also explain how to find suitable coefficients for the interpretation functions. Section 4.3 considers different encodings of Hydra battles where also the limitations of the approach are discussed.

In the sequel we consider ordinal expressions of the following shape. By $\overline{x}$ we abbreviate $x_1, \ldots, x_n$.

Definition 16. A *restricted ordinal expression* (*ROE*) over variables $\overline{x}$ is either 0 or [4]

$$\sum_{1 \leqslant i \leqslant n} x_i f_i + \omega^{f'(\overline{x})} f_\omega \oplus \bigoplus_{1 \leqslant i \leqslant n} x_i \hat{f}_i \oplus f_0 \tag{3}$$

where $f_0, f_1, \ldots, f_n, \hat{f}_1, \ldots, \hat{f}_n, f_\omega$ are natural numbers and $f'(\overline{x})$ is an ROE over $\overline{x}$. The *depth* of an ROE is the height of the tower of ω's. An *ROE algebra* is an algebra $\mathcal{O}$ where for every n-ary function symbol f the interpretation function $f_\mathcal{O}$ is an ROE over $\overline{x}$.

[4] To enhance readability we drop parentheses in expressions of the form $x + y \oplus z$, which are to be read as $(x + y) \oplus z$ rather than $x + (y \oplus z)$. Note that these expressions are in general not equivalent, e.g., $(1 + 0) \oplus \omega = \omega + 1$ but $1 + (0 \oplus \omega) = \omega$.

4.1. Encodings

Let $f(\overline{x})$ and $g(\overline{x})$ be ROEs of the form

$$\begin{aligned} f(\overline{x}) &= \sum_{1\leqslant i\leqslant n} x_i f_i + \omega^{f'(\overline{x})} f_\omega \oplus \bigoplus_{1\leqslant i\leqslant n} x_i \hat{f}_i \oplus f_0 \\ g(\overline{x}) &= \sum_{1\leqslant i\leqslant n} x_i g_i + \omega^{g'(\overline{x})} g_\omega \oplus \bigoplus_{1\leqslant i\leqslant n} x_i \hat{g}_i \oplus g_0 \end{aligned} \tag{4}$$

We assume that these expressions depend on the same variables $\overline{x}$ (otherwise the respective coefficients can be set to 0), and that variables appear in the same order. We first encode some auxiliary properties of ROEs.

4.1.1. Useful Abbreviations

Let $\mathrm{zero}(f(\overline{x}))$ be true if and only if $f(\overline{x}) = 0$ or all of f_0, f_i, $\hat{f}_i$ and f_ω are 0. Let $c_i = \max(f_i, g_i)$ for all $i \in \{0, \dots, n, \omega\}$. An upper bound $\mathrm{omax}(f, g)(\overline{x})$ is then given by $\mathrm{omax}(f, 0)(\overline{x}) = \mathrm{omax}(0, f)(\overline{x}) = f(\overline{x})$ and

$$\mathrm{omax}(f,g)(\overline{x}) = \sum_{1\leqslant i\leqslant n} x_i c_i + \omega^{\mathrm{omax}(f',g')(\overline{x})} c_\omega \oplus \bigoplus_{1\leqslant i\leqslant n} x_i \max(\hat{f}_i, \hat{g}_i) \oplus c_0$$

otherwise. For instance, if $f(\overline{x}) = x_1 + \omega^{x_2+1} \oplus x_3$ and $g(\overline{x}) = \omega^{x_1} 2 \oplus x_2 + 1$ then $\mathrm{omax}(f, g)(\overline{x}) = x_1 + \omega^{x_1+x_2+1} 2 \oplus x_2 \oplus x_3 + 1$. Clearly, $[\alpha](f(\overline{x})) \leqslant [\alpha](\mathrm{omax}(f, g)(\overline{x}))$ and $[\alpha](g(\overline{x})) \leqslant [\alpha](\mathrm{omax}(f, g)(\overline{x}))$ for all assignments α. Whether a variable x_i *contributes* to the value of $f(\overline{x})$ can be recursively encoded as follows:

$$\mathrm{con}_i(f(\overline{x})) = \begin{cases} \bot & \text{if } f(\overline{x}) = 0 \\ f_i > 0 \vee \hat{f}_i > 0 \vee (\mathrm{con}_i(f'(\overline{x})) \wedge f_\omega > 0) & \text{otherwise} \end{cases}$$

If $f(\overline{x})$ and $g(\overline{x})$ are defined as above then $\mathrm{con}_i(f(\overline{x})) = \mathrm{con}_j(g(\overline{x})) = \top$ for all $1 \leqslant i \leqslant 3$ and $1 \leqslant j \leqslant 2$, but $\mathrm{con}_3(g(\overline{x})) = \bot$.

4.1.2. Comparisons

Consider ROEs $f(\overline{x})$ and $g(\overline{x})$ as in (4). We want to derive sufficient (checkable) conditions such that $[\alpha](f(\overline{x})) > [\alpha](g(\overline{x}))$ for all assignments α. The following example shows that whether one ROE is larger than another one significantly depends on the assignment.

Example 17. Consider $x_1 + x_2$ and $x_2 + x_1$. Let α be an assignment such that $\alpha(x_1) = \omega$ and $\alpha(x_2) = 1$. Then $[\alpha](x_1 + x_2) = \omega + 1 > \omega = 1 + \omega = [\alpha](x_2 + x_1)$. Conversely we have $[\beta](x_1 + x_2) = 1 + \omega = \omega < \omega + 1 = [\beta](x_2 + x_1)$ when $\beta(x_1) = 1$ and $\beta(x_2) = \omega$.

We use the following underapproximation to check whether $[\alpha](f(\overline{x})) > [\alpha](g(\overline{x}))$ for all assignments α, which is a tradeoff between accuracy and efficiency.

Definition 18. Let $f(\overline{x})$ and $g(\overline{x})$ be ROEs as in (4).

$$\begin{aligned} [f(\overline{x}) \geqslant g(\overline{x})] &= [f(\overline{x}) \geqslant_0 g(\overline{x})] \wedge \bigwedge_{1\leqslant i\leqslant n} [f(\overline{x}) \geqslant_i g(\overline{x})] \\ [f(\overline{x}) \geqslant_0 g(\overline{x})] &= ([f'(\overline{x}) >_0 g'(\overline{x})] \wedge f_\omega > 0) \vee \\ &\quad ([f'(\overline{x}) \geqslant_0 g'(\overline{x})] \wedge f_\omega \geqslant g_\omega \wedge f_0 \geqslant g_0) \vee \\ &\quad (g_\omega = 0 \wedge f_0 \geqslant g_0) \end{aligned}$$

$$
\begin{aligned}
[f(\overline{x}) \geqslant_i g(\overline{x})] = {} & \neg \mathrm{con}_i(g(\overline{x})) \vee && \text{(a)}\\
& ([f'(\overline{x}) \geqslant_i g'(\overline{x})] \wedge f_\omega \geqslant g_\omega \wedge g_i = 0 \wedge \hat{g}_i = 0) \vee && \text{(b)}\\
& (\mathrm{con}_i(\omega^{f'(\overline{x})} f_\omega) \wedge \neg \mathrm{con}_i(\omega^{g'(\overline{x})} g_\omega)) \vee && \text{(c)}\\
& (\mathrm{con}_i(\omega^{f'(\overline{x})} f_\omega) \wedge [f'(\overline{x}) \geqslant_i g'(\overline{x})] \wedge f_\omega > g_\omega) \vee && \text{(d)}\\
& (\mathrm{con}_i(\omega^{f'(\overline{x})} f_\omega) \wedge [f'(\overline{x}) \geqslant_i g'(\overline{x})] \wedge f_\omega = g_\omega \wedge \hat{f}_i \geqslant \hat{g}_i) \vee && \text{(e)}\\
& (\neg \mathrm{con}_i(\omega^{g'(\overline{x})} g_\omega) \wedge \hat{f}_i \geqslant \hat{g}_i \wedge f_i + \hat{f}_i \geqslant g_i + \hat{g}_i) \vee && \text{(f)}\\
& ((\mathrm{zero}(g'(\overline{x})) \vee g_\omega = 0) \wedge f_i + \hat{f}_i \geqslant g_i + \hat{g}_i) && \text{(g)}\\
[f(\overline{x}) > g(\overline{x})] = {} & [f(\overline{x}) \geqslant g(\overline{x})] \wedge [f(\overline{x}) >_0 g(\overline{x})]\\
[f(\overline{x}) >_0 g(\overline{x})] = {} & ([f'(\overline{x}) >_0 g'(\overline{x})] \wedge f_\omega > 0) \vee\\
& ([f'(\overline{x}) \geqslant_0 g'(\overline{x})] \wedge f_\omega \geqslant g_\omega \wedge f_0 > g_0) \vee\\
& (g_\omega = 0 \wedge f_0 > g_0)
\end{aligned}
$$

Here $[f(\overline{x}) >_0 g(\overline{x})]$ ($[f(\overline{x}) \geqslant_0 g(\overline{x})]$) encodes that the constant part in $f(\overline{x})$ is greater (or equal) than the constant part in $g(\overline{x})$, whereas $[f(\overline{x}) \geqslant_i g(\overline{x})]$ encodes that the coefficients of the variable x_i in $f(\overline{x})$ are greater than or equal to the respective coefficients in $g(\overline{x})$. The last disjunct in the definition of $[f(\overline{x}) >_0 g(\overline{x})]$ was added to the earlier version of our encoding (Winkler et al., 2013); it is essential to handle the last rule of the TRS $\mathcal{W}_3'$ in Example 31. Our comparisons are (much) more involved than the absolute positiveness approach (Hong and Jakuš, 1998) for polynomials because of ordinal arithmetic. We illustrate the different cases in the encoding of $\geqslant_i$ in the following example.

Example 19. Case (a) yields $[\omega^{x_1+x_2} \geqslant_1 \omega^{x_2}]$ while (b) admits $[\omega^{x_1 2}3 \geqslant_1 \omega^{x_1}3]$. From (c) validity of $[\omega^{x_1}2 \geqslant_1 x_1 3]$ is obtained while $[\omega^{x_1}2 \geqslant_1 \omega^{x_1}1 \oplus x_1 5]$ is due to (d). Case (e) obviously allows $[\omega^{x_1}2 \oplus x_1 2 \geqslant_1 \omega^{x_1}2 \oplus x_1 1]$ but also $[\omega^{x_1} \geqslant_1 x_1 10 + \omega^{x_1}]$. Case (f) implies $[x_1 2 + \omega^{x_2} \oplus x_1 3 \geqslant_1 x_1 3 + \omega^{x_2} \oplus x_1 2]$. Finally, (g) ensures $[x_1 4 + \omega^{x_2} \oplus x_1 1 \geqslant_1 x_1 2 \oplus x_1 3]$. It is not hard to check that for all these example ROEs satisfying $[f(x_1, x_2) \geqslant_1 g(x_1, x_2)]$ we indeed have $[\alpha](f(x_1, x_2)) \geqslant [\alpha](g(x_1, x_2))$ for any assignment α (though additional constraints are required to ensure this). In the example for case (f), the test $\hat{f}_1 \geqslant \hat{g}_1$ is required if $\omega^{\alpha(x_2)}$ consumes the preceding $\alpha(x_1)2$ (and hence $\alpha(x_1)3$) for some assignment α. Otherwise the test $f_1 + \hat{f}_1 \geqslant g_1 + \hat{g}_1$ is required. For case (g), if for some α the term $\omega^{\alpha(x_2)}$ consumes $\alpha(x_1)4$ then it also dominates $\alpha(x_1)2$. Otherwise we need the test $f_1 + \hat{f}_1 \geqslant g_1 + \hat{g}_1$.

Clearly, the encoding of $\geqslant$ is only an approximation. E.g., $[\omega^{x_1+1} \geqslant_1 \omega^{x_1}2]$ is not valid, despite the fact that $\omega^{\alpha(x_1)+1} > \omega^{\alpha(x_1)}2$ for any α. While it is straightforward to extend Definition 18(b) accordingly for this particular case, we do not strive for a precise encoding, which seems out of reach for practical applications.

The encodings of comparisons are sound.

Lemma 20. *Let $f(\overline{x})$ and $g(\overline{x})$ be ROEs as in (4).*

(a) *If $[f(\overline{x}) > g(\overline{x})]$ then $[\alpha](f(\overline{x})) > [\alpha](g(\overline{x}))$ for all assignments α.*
(b) *If $[f(\overline{x}) \geqslant g(\overline{x})]$ then $[\alpha](f(\overline{x})) \geqslant [\alpha](g(\overline{x}))$ for all assignments α.*

Proof. Each of the disjunctions (a)–(g) in Definition 18 is a sound criterion for the comparison $[\alpha](f(\overline{x})) \geqslant_i [\alpha](g(\overline{x}))$ for all $1 \leqslant i \leqslant n$. $\square$

4.1.3. Composition

In contrast to e.g. polynomial interpretations, ROEs are not closed under scalar multiplication and standard/natural addition (cf. Example 21), and thus also not under composition. Hence we cannot compute an ROE corresponding to the interpretation of a term t with respect to an ROE algebra $\mathcal{O}$. Instead, we define ROEs $\mu(t)$ and $\nu(t)$ to under- and overapproximate $t_{\mathcal{O}}$. To this end we present in Definition 22 bounds for the results of ordinal arithmetic operations (based on the algorithms given in Manolios and Vroon (2005) for ordinals in CNF) and demonstrate them in Example 23 before Lemma 24 shows their soundness.

Example 21.

(a) Consider the ROEs $x+1$ and 2. If $\alpha(x) < \omega$ then $[\alpha]((x+1)\cdot 2) = [\alpha](x2+2)$ but $[\alpha]((x+1)\cdot 2) = [\alpha](x2+1)$ otherwise.
(b) Consider the ROEs ω^2 and ω^3. There is no ROE for $\omega^2 \oplus \omega^3$.
(c) Consider the ROEs $x \oplus 1$ and y. If $\alpha(y) < \omega$ then $[\alpha]((x\oplus 1)+y) = [\alpha](x+y+1)$ but $[\alpha]((x\oplus 1)+y) = [\alpha](x+y)$ otherwise.

Definition 22. Let $f(\overline{x})$ and $g(\overline{x})$ be ROEs as in (4).

(a) For $a \in \mathbb{N}$, let $(f \cdot_\mu a)(\overline{x}) = (f \cdot_\nu a)(\overline{x}) = 0$ if $a = 0$ or $f(\overline{x}) = 0$, and otherwise

$$(f \cdot_\mu a)(\overline{x}) = \sum_{1\leqslant i\leqslant n} x_i f_i + \omega^{f'(\overline{x})}(f_\omega \cdot a) \oplus \bigoplus_{1\leqslant i\leqslant n} x_i(\hat{f}_i \cdot a) \oplus (f_0 \cdot a)$$
$$(f \cdot_\nu a)(\overline{x}) = \sum_{1\leqslant i\leqslant n} x_i (f_i \cdot a) + \omega^{f'(\overline{x})}(f_\omega \cdot a) \oplus \bigoplus_{1\leqslant i\leqslant n} x_i(\hat{f}_i \cdot a) \oplus (f_0 \cdot a)$$

(b) Let $(f \oplus_\mu g)(\overline{x}) = (f \oplus_\nu g)(\overline{x}) = g(\overline{x})$ if $f(\overline{x}) = 0$ and similarly $(f \oplus_\mu g)(\overline{x}) = (f \oplus_\nu g)(\overline{x}) = f(\overline{x})$ if $g(\overline{x}) = 0$. Otherwise, let s_i and t_i abbreviate $\mathrm{con}_i(\omega^{f'(\overline{x})} f_\omega)\ ?\ 0 : 1$ and $\mathrm{con}_i(\omega^{g'(\overline{x})} g_\omega)\ ?\ 0 : 1$, where $b\ ?\ t : e$ encodes "if b then t else e". Let

$$(h(\overline{x}), h_\omega) = \begin{cases} (f'(\overline{x}), f_\omega + 1) & \text{if } [\omega^{f'(\overline{x})} f_\omega > \omega^{g'(\overline{x})} g_\omega] \\ (g'(\overline{x}), g_\omega + 1) & \text{if } [\omega^{g'(\overline{x})} g_\omega > \omega^{f'(\overline{x})} f_\omega] \\ (\mathrm{omax}(f', g')(\overline{x}), f_\omega + g_\omega) & \text{otherwise} \end{cases}$$

and $(k(\overline{x}), k_\omega) = [\omega^{f'(\overline{x})} f_\omega > \omega^{g'(\overline{x})} g_\omega]\ ?\ (f'(\overline{x}), f_\omega) : (g'(\overline{x}), g_\omega)$. Then

$$(f \oplus_\mu g)(\overline{x}) = \sum_{1\leqslant i\leqslant n} x_i \max(f_i s_i, g_i t_i) + \omega^{k(\overline{x})} k_\omega \oplus \bigoplus_{1\leqslant i\leqslant n} x_i(\hat{f}_i + \hat{g}_i) \oplus (f_0 + g_0)$$

$$\begin{aligned}(f \oplus_\nu g)(\overline{x}) = \mathrm{nat}(g(\overline{x}))\ ?\ &\sum_{1\leqslant i\leqslant n} x_i f_i + \omega^{f'(\overline{x})} f_\omega \oplus \bigoplus_{1\leqslant i\leqslant n} x_i(\hat{f}_i + \hat{g}_i) \oplus (f_0 + g_0) : \\ \mathrm{nat}(f(\overline{x}))\ ?\ &\sum_{1\leqslant i\leqslant n} x_i g_i + \omega^{g'(\overline{x})} g_\omega \oplus \bigoplus_{1\leqslant i\leqslant n} x_i(\hat{f}_i + \hat{g}_i) \oplus (f_0 + g_0) : \\ \omega^{h(\overline{x})} h_\omega \oplus &\bigoplus_{1\leqslant i\leqslant n} x_i(\hat{f}_i + \hat{g}_i + g_i t_i + f_i s_i) \oplus (f_0 + g_0)\end{aligned}$$

Here $\operatorname{nat}(f(\overline{x}))$ abbreviates $f_1 = 0 \wedge \cdots \wedge f_n = 0 \wedge f_\omega = 0$, and similarly for $g(\overline{x})$. This definition of $(f \oplus_\nu g)(\overline{x})$ allows for a more precise encoding compared to the version in Winkler et al. (2013). In particular, a tighter upper bound is obtained for the cases where $f(\overline{x})$ or $g(\overline{x})$ are just linear polynomials (i.e., where $\operatorname{nat}(f(\overline{x}))$ or $\operatorname{nat}(g(\overline{x}))$ is true).

(c) Let $(f +_\mu g)(\overline{x}) = (f +_\nu g)(\overline{x}) = g(\overline{x})$ if $f(\overline{x}) = 0$ and $(f +_\mu g)(\overline{x}) = (f +_\nu g)(\overline{x}) = f(\overline{x})$ if $g(\overline{x}) = 0$. Otherwise, we define lower and upper bounds for $f(\overline{x}) + g(\overline{x})$ by distinguishing different cases using if-then-else expressions:

$$(f +_\mu g)(\overline{x}) = [\omega^{f'(\overline{x})} f_\omega > \omega^{g'(\overline{x})} g_\omega] \mathbin{?} f(\overline{x}) : \Bigl(\sum_{1 \leqslant i \leqslant n} (g_i = 0 \mathbin{?} x_i f_i : 0) + g(\overline{x}) \Bigr)$$

$$(f +_\nu g)(\overline{x}) = [\omega^{g'(\overline{x})} g_\omega > \omega^{f'(\overline{x})} f_\omega] \mathbin{?} \phi_1 : \bigl([\omega^{f'(\overline{x})} f_\omega > \omega^{g'(\overline{x})} g_\omega] \mathbin{?} \phi_2 : (f \oplus_\nu g)(\overline{x}) \bigr)$$

where

$$\phi_1 = \sum_{1 \leqslant i \leqslant n} x_i (f_i s_i t_i + \hat{f}_i t_i u + g_i t_i) + \omega^{g'(\overline{x})} g_\omega \oplus \bigoplus_{1 \leqslant i \leqslant n} x_i (\hat{f}_i t_i (1 - u) + \hat{g}_i) \oplus c_0$$

$$\phi_2 = \sum_{1 \leqslant i \leqslant n} x_i f_i s_i + \omega^{f'(\overline{x})} (f_\omega + 1) \oplus \bigoplus_{1 \leqslant i \leqslant n} x_i (\hat{f}_i t_i + g_i t_i + \hat{g}_i) \oplus c_0$$

with $c_0 = ([g'(\overline{x}) > 0] \wedge g_\omega > 0) \mathbin{?} g_0 : f_0 + g_0$ and u is 1 if all $f_i s_i t_i$ are zero and at most one of $\hat{f}_i t_i$ is greater zero and 0 otherwise.

(d) Definitions (a)–(c) can be used to inductively set lower and upper bounds for the composition $f(\overline{g})(\overline{x}) = f(g_1(\overline{x}), \ldots, g_n(\overline{x}))$. We write $\sum^\mu_{1 \leqslant i \leqslant n} h_i$ to abbreviate $h_1 +_\mu \cdots +_\mu h_n$, and use similar shorthands for $\oplus$ and ν. We set

$$f(\overline{g})_\mu(\overline{x}) = \sum^\mu_{1 \leqslant i \leqslant n} g_i(\overline{x}) \cdot_\mu f_i +_\mu \omega^{f'(\overline{g})_\mu(\overline{x})} f_\omega \oplus_\mu \bigoplus^\mu_{1 \leqslant i \leqslant n} g_i(\overline{x}) \cdot_\mu \hat{f}_i \oplus_\mu f_0$$

$$f(\overline{g})_\nu(\overline{x}) = \sum^\nu_{1 \leqslant i \leqslant n} g_i(\overline{x}) \cdot_\nu f_i +_\nu \omega^{f'(\overline{g})_\nu(\overline{x})} f_\omega \oplus_\nu \bigoplus^\nu_{1 \leqslant i \leqslant n} g_i(\overline{x}) \cdot_\nu \hat{f}_i \oplus_\nu f_0$$

(e) Let t be a term, and $\mathcal{O}$ be an ROE algebra. By induction on the term structure we define ROEs $\mu_{\mathcal{O}}(t)$ and $\nu_{\mathcal{O}}(t)$ such that

$$\mu_{\mathcal{O}}(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f_{\mathcal{O}}(\mu_{\mathcal{O}}(t_1), \ldots, \mu_{\mathcal{O}}(t_n))_\mu & \text{otherwise} \end{cases}$$

$$\nu_{\mathcal{O}}(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f_{\mathcal{O}}(\nu_{\mathcal{O}}(t_1), \ldots, \nu_{\mathcal{O}}(t_n))_\nu & \text{otherwise} \end{cases}$$

The following example illustrates these definitions of upper and lower bounds for ROE arithmetic.

Example 23.

(a) Consider the ROE $f(\overline{x}) = x_1 + x_2$. Then $(f \cdot_\mu 2)(\overline{x}) = x_1 + x_2$ and $(f \cdot_\nu 2)(\overline{x}) = x_1 2 + x_2 2$. We clearly have $x_1 + x_2 \leqslant (x_1 + x_2)2 \leqslant x_1 2 + x_2 2$ for all values of x_1 and x_2. Note that $(x_1 + x_2)2 \neq x_1 2 + x_2 2$ since $\cdot$ does not right-distribute over $+$, as shown after Definition 2.

(b) Consider the ROEs $f(\overline{x}) = \omega^{x_1+x_2+1} \oplus x_3 + 1$ and $g(\overline{x}) = x_2 + \omega^{x_1}2 \oplus x_3$. As $\omega^{x_1+x_2+1} > \omega^{x_1}2$ we have $(k(\overline{x}), k_\omega) = (x_1 + x_2 + 1, 1)$ and $(h(\overline{x}), h_\omega) = (x_1 + x_2 + 1, 2)$. Thus $(f \oplus_\mu g)(\overline{x}) = x_2 + \omega^{x_1+x_2+1} \oplus x_3 2 + 1$ and $(f \oplus_\nu g)(\overline{x}) = \omega^{x_1+x_2+1}2 \oplus x_2 \oplus x_3 2 + 1$. It is not difficult to see that

$$x_2 + \omega^{x_1+x_2+1} \oplus x_3 2 + 1 \leqslant f(\overline{x}) \oplus g(\overline{x}) \leqslant \omega^{x_1+x_2+1}2 \oplus x_2 \oplus x_3 2 + 1$$

for all values of x_1, x_2, and x_3.

(c) Consider the ROEs $f(\overline{x}) = x_3 + \omega^{x_2} \oplus x_1$ and $g(\overline{x}) = \omega^{x_1+x_2+1} + 1$. As $\omega^{x_2} \not> \omega^{x_1+x_2+1}$ we have $(f +_\mu g)(\overline{x}) = x_3 + g(\overline{x}) = x_3 + \omega^{x_1+x_2+1} + 1$. Since $x_1 + x_2 + 1 > x_2$ the first case for $+_\nu$ applies, where $u = 0$ as $f_3 s_3 t_3 = 1$. We thus have $(f +_\nu g)(\overline{x}) = \omega^{x_1+x_2+1} \oplus x_3 + 1$. Note that the term $\oplus x_1$ in $f(\overline{x})$ disappears as x_1 contributes to the exponent of $g(\overline{x})$. We have

$$x_3 + \omega^{x_1+x_2+1} + 1 \leqslant (x_3 + \omega^{x_2} \oplus x_1) + (\omega^{x_1+x_2+1} + 1) \leqslant \omega^{x_1+x_2+1} \oplus x_3 + 1$$

for all values of x_1, x_2, and x_3.

(d) For the ROEs $f(\overline{x}) = x_2 + \omega^{x_1+1}$, $g_1(\overline{x}) = \omega^{x_1} \oplus x_2$, and $g_2(\overline{x}) = \omega^{\omega^{x_1} \oplus x_2} \oplus x_3$ we obtain

$$f(\overline{g})_\mu(\overline{x}) = (\omega^{\omega^{x_1} \oplus x_2} \oplus x_3) +_\mu \omega^{\omega^{x_1} \oplus x_2 + 1} = \omega^{\omega^{x_1} \oplus x_2 + 1}$$
$$f(\overline{g})_\nu(\overline{x}) = (\omega^{\omega^{x_1} \oplus x_2} \oplus x_3) +_\nu \omega^{\omega^{x_1} \oplus x_2 + 1} = x_3 + \omega^{\omega^{x_1} \oplus x_2 + 1}$$

(e) Consider the terms $\ell = \bullet\,\mathsf{f}(\mathsf{c}(x_1, x_2), x_3)$ and $r = \mathsf{h}(\bullet\,\mathsf{f}(x_1, x_2), \bullet\bullet\,\mathsf{f}(\mathsf{f}(x_1, x_2), x_3))$ from rule (C2) of $\mathcal{G}$. Let $\mathcal{O}$ be the ordinal part of the ROE algebra defined in the proof of Theorem 15 such that $\mathsf{h}_\mathcal{O}(x_1, x_2) = x_2 + \omega^{x_1+1}$, $\mathsf{c}_\mathcal{O}(x_1, x_2) = \omega^{x_1} \oplus x_2 + 1$, $\bullet_\mathcal{O}(x_1) = x_1$, and $\mathsf{f}_\mathcal{O}(x_1, x_2) = \omega^{x_1} \oplus x_2$. We have $\mu_\mathcal{O}(\ell) = \nu_\mathcal{O}(\ell) = \omega^{\omega^{x_1} \oplus x_2 + 1} \oplus x_3$. It is easy to see that for $r' = \mathsf{f}(\mathsf{f}(x_1, x_2), x_3)$ we get $\mu_\mathcal{O}(r') = \nu_\mathcal{O}(r') = \omega^{\omega^{x_1} \oplus x_2} \oplus x_3$. From the computation in (d) we thus obtain $\nu_\mathcal{O}(r) = x_3 + \omega^{\omega^{x_1} \oplus x_2 + 1}$. Note that $[\mu_\mathcal{O}(\ell) \geqslant \nu_\mathcal{O}(r)]$ holds: We obviously have $[\mu_\mathcal{O}(\ell) \geqslant_0 \nu_\mathcal{O}(r)]$, $[\mu_\mathcal{O}(\ell) \geqslant_1 \nu_\mathcal{O}(r)]$, and $[\mu_\mathcal{O}(\ell) \geqslant_2 \nu_\mathcal{O}(r)]$ as the two expressions are equal in the relevant parts, and $[\mu_\mathcal{O}(\ell) \geqslant_3 \nu_\mathcal{O}(r)]$.

Note that in (Winkler et al., 2013, Definition 17) we approximated $(x + \omega^0 0) \oplus_\nu \omega^x$ by $x + \omega^x$ (but $[\alpha](x \oplus \omega^x) > [\alpha](x + \omega^x)$ for $\alpha(x) = 1$), and $(x \oplus y) +_\nu \omega$ by $x + y + \omega$ (whereas $[\alpha]((x \oplus y) + \omega) > [\alpha](x + y + \omega)$ for $\alpha(x) = \omega$ and $\alpha(y) = \omega^2$). Definition 22 corrects these flaws and sets $(x + \omega^0 0) \oplus_\nu \omega^x = \omega^x \oplus x$ and $(x \oplus y) +_\nu \omega = \omega \oplus x \oplus y$. We now show that Definition 22 yields valid over- and underapproximations.

Lemma 24. *Let $\mathcal{O}$ be an ROE algebra and t be a term. Then $[\alpha](\mu_\mathcal{O}(t)) \leqslant [\alpha]_\mathcal{O}(t) \leqslant [\alpha](\nu_\mathcal{O}(t))$ for all assignments α.*

Proof. We argue that all approximations in Definition 22 constitute valid lower and upper bounds. Let α be an arbitrary assignment.

(a) It is easy to see that $[\alpha](f(\overline{x}) \cdot a) \leqslant [\alpha](f \cdot_\nu a)(\overline{x})$. For any β in CNF as in (1) and $a \in \mathbb{N}_{>0}$ we have $\beta a = \omega^{\beta_1} a_1 a + \omega^{\beta_2} a_2 + \cdots + \omega^{\beta_n} a_n$ (Manolios and Vroon, 2005). Since for any $1 \leqslant i \leqslant n$ we have $\omega^{\beta_1} a_1 a + \cdots + \omega^{\beta_n} a_n \geqslant \omega^{\beta_1} a_1 + \cdots + \omega^{\beta_i} a_i a + \cdots + \omega^{\beta_n} a_n$, $(f \cdot_\mu a)(\overline{x})$ constitutes a safe (though modest) lower bound for $f(\overline{x})a$.

(b) We have

$$f(\overline{x}) \oplus g(\overline{x}) = \left(\sum_{1 \leqslant i \leqslant n} x_i f_i + \omega^{f'(\overline{x})} f_\omega\right) \oplus \left(\sum_{1 \leqslant i \leqslant n} x_i g_i + \omega^{g'(\overline{x})} g_\omega\right) \qquad (5)$$
$$\oplus \bigoplus_{1 \leqslant i \leqslant n} x_i(\hat{f}_i + \hat{g}_i) \oplus (f_0 + g_0)$$

Note that the term $x_i f_i$ disappears in $f(\overline{x}) \oplus g(\overline{x})$ if x_i contributes to $\omega^{f'(\overline{x})}$ and $f_\omega > 0$, and the term $x_i g_i$ disappears in $f(\overline{x}) \oplus g(\overline{x})$ if x_i contributes to $\omega^{g'(\overline{x})}$ and $g_\omega > 0$. Hence we may multiply all occurrences of f_i by s_i, and occurrences of g_i by t_i. We then have $[\alpha](f \oplus_\mu g)(\overline{x}) \leqslant [\alpha](f(\overline{x}) \oplus g(\overline{x}))$ as $(f \oplus_\mu g)(\overline{x})$ underapproximates

$$\left(\sum_{1 \leqslant i \leqslant n} x_i f_i + \omega^{f'(\overline{x})} f_\omega\right) \oplus \left(\sum_{1 \leqslant i \leqslant n} x_i g_i + \omega^{g'(\overline{x})} g_\omega\right)$$

by a coefficient-wise maximum of the respective components in $f(\overline{x})$ and $g(\overline{x})$. Concerning the upper bound, the first two cases are obvious from Equation (5). Otherwise, it is easy to see that $\omega^{f'(\overline{x})} f_\omega \oplus \omega^{g'(\overline{x})} g_\omega \leqslant \omega^{h(\overline{x})} h_\omega$. As the sum of $x_i f_i$ and $x_i g_i$ can be overapproximated by the natural sum of all terms $(f_i s_i + g_i t_i) x_i$ we have $[\alpha](f(\overline{x}) \oplus g(\overline{x})) \leqslant [\alpha](f \oplus_\nu g)(\overline{x})$.

(c) We clearly have $[\alpha](f +_\mu g)(\overline{x}) \leqslant [\alpha](f(\overline{x}) + g(\overline{x}))$. Concerning the upper bound, assume for a first case $[\omega^{g'(\overline{x})} g_\omega > \omega^{f'(\overline{x})} f_\omega]$, so $\omega^{f'(\overline{x})} f_\omega + \omega^{g'(\overline{x})} g_\omega = \omega^{g'(\overline{x})} g_\omega$. Note that the term $x_i \hat{f}_i$ disappears in $f(\overline{x}) + g(\overline{x})$ if x_i is contained in $\omega^{g'(\overline{x})} g_\omega$, i.e., if x_i occurs with a positive coefficient somewhere in $g'(\overline{x})$ and $g_\omega > 0$. The term $g_i x_i$ disappears as well if x_i is contained in $\omega^{g'(\overline{x})} g_\omega$, and $f_i x_i$ disappears if x_i occurs in $\omega^{f'(\overline{x})} f_\omega$, or if x_i occurs in $\omega^{g'(\overline{x})} g_\omega$. Hence all occurrences of $\hat{f}_i$ and g_i may be multiplied by t_i, and occurrences of f_i may be multiplied by $s_i t_i$. Clearly all terms $x_i f_i s_i t_i$ and $x_i g_i t_i$ may be put in the standard addition part of $(f +_\nu g)(\overline{x})$, and $x_i \hat{g}_i$ occurs in the natural addition part. As far as the terms $x_i \hat{f}_i t_i$ are concerned, adding them to the natural addition part is obviously sound; but note that we may also put $x_i \hat{f}_i t_i$ into the standard addition part if $x_i \hat{f}_i$ is the only part of $f(\overline{x})$ that survives, which is captured by the condition $u = 1$. Now suppose $[\omega^{f'(\overline{x})} f_\omega > \omega^{g'(\overline{x})} g_\omega]$, so $\omega^{f'(\overline{x})} f_\omega + \omega^{g'(\overline{x})} g_\omega \leqslant \omega^{f'(\overline{x})}(f_\omega + 1)$. The term $\hat{f}_i x_i$ disappears in $f(\overline{x}) + g(\overline{x})$ if x_i is contained in $\omega^{g'(\overline{x})} g_\omega$, the term $g_i x_i$ disappears as well if x_i is contained in $\omega^{g'(\overline{x})} g_\omega$. Hence for any variable x_i the sum of $x_i \hat{f}_i$, $x_i g_i$, and $x_i \hat{g}_i$ can be overapproximated by $x_i(\hat{f}_i t_i + g_i t_i + \hat{g}_i)$ such that $[\alpha](f(\overline{x}) + g(\overline{x})) \leqslant [\alpha](f +_\nu g)(\overline{x})$. Finally, $f(\overline{x}) + g(\overline{x}) \leqslant f(\overline{x}) \oplus g(\overline{x}) \leqslant (f \oplus_\nu g)(\overline{x})$ holds in any case.

(d) By (a)–(c) and weak monotonicity of the ordinal operations $\cdot$, $+$, and $\oplus$.

(e) By induction on the term structure of t, using (d). □

4.1.4. Main Theorem

Any ROE is weakly monotone and well-defined by definition. It is easy to encode a criterion for an ROE $f(\overline{x})$ to be simple:

$$\text{simple}(f(\overline{x})) = \bigwedge_{1 \leqslant i \leqslant n} \text{con}_i(f(\overline{x}))$$

Finally we obtain the main result of this section.

Theorem 25. *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and $\mathcal{O}$ an ROE algebra on $\mathcal{F}$. If*

$$\bigwedge_{\ell \to r \in \mathcal{R}} [\mu_{\mathcal{O}}(\ell) > \nu_{\mathcal{O}}(r)] \wedge \bigwedge_{f \in \mathcal{F}} \text{simple}(f_{\mathcal{O}}(\overline{x}))$$

holds then $\mathcal{R}$ is terminating.

Proof. We already observed that any ROE is weakly monotone and well-defined. By the assumption, every $f_{\mathcal{O}}$ is simple. Hence the result follows by Theorem 6 in combination with Lemmata 20 and 24. □

4.2. Implementation

In this section we discuss crucial issues for a successful implementation. Section 4.2.1 explains the search for suitable interpretations. Section 4.2.2 shows how to ensure that the lexicographic combination of partial proofs preserves weak monotonicity. Section 4.2.3 deals with the problem of a compatible variable order and Section 4.2.4 is dedicated to efficiency considerations.

4.2.1. Search for Interpretations

In automatic termination proofs suitable interpretation functions must be constructed. While easy heuristics can be employed for the depth of an ROE (see Section 4.2.4), the main challenge is to establish suitable coefficients. To this end we consider *parametric* ROEs which are of the shape (4) with the exception that now $f_0, f_1, \ldots, f_n, \hat{f}_1, \ldots, \hat{f}_n, f_\omega$ are *unknowns* over the naturals. The encodings from the previous section then allow to reduce the search for suitable coefficients to finding models in existentially quantified non-linear integer arithmetic for which suitable SMT solvers exist (see e.g. Zankl and Middeldorp (2010)).

4.2.2. Lexicographic Combination of Interpretations

The termination proof of the TRS $\mathcal{G}$ (Theorem 15) performs a lexicographic combination of algebras into a simple and weakly monotone algebra. The proof can be seen as the lexicographic product of (1) an ordinal algebra and (2) a linear (polynomial) interpretation and (3) a matrix interpretation of dimension 2 (Endrullis et al., 2008). [5] Regarding automation one can either encode the search for the lexicographic combination or search for (partial) proofs and combine them lexicographically. We adopted the latter, although the lexicographic combination of weakly monotone algebras need not be weakly monotone, as shown by the following example.

Example 26. Consider the nonterminating TRS $\mathcal{R} = \{\mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{b}), \mathsf{b} \to \mathsf{a}\}$. For the weakly monotone simple interpretation $\mathsf{f}_{\mathcal{O}}(x) = x + \omega, \mathsf{b}_{\mathcal{O}} = 1, \mathsf{a}_{\mathcal{O}} = 0$ we have $[\mathsf{f}(\mathsf{a})]_{\mathcal{O}} = \omega \geqslant \omega = [\mathsf{f}(\mathsf{b})]_{\mathcal{O}}$ and $[\mathsf{b}]_{\mathcal{O}} = 1 > 0 = [\mathsf{a}]_{\mathcal{O}}$. If we removed the second rule, then the weakly monotone simple interpretation $\mathsf{f}_{\mathcal{N}}(x) = x + 1, \mathsf{a}_{\mathcal{N}} = 1, \mathsf{b}_{\mathcal{N}} = 0$ shows termination of the remaining rule $\mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{b})$. Note that the lexicographic combination is no longer weakly monotone, i.e., $[\mathsf{b}]_{\mathcal{O} \times \mathcal{N}} = (1, 0) >_{\mathsf{lex}} (0, 1) = [\mathsf{a}]_{\mathcal{O} \times \mathcal{N}}$ but $[\mathsf{f}(\mathsf{b})]_{\mathcal{O} \times \mathcal{N}} = (\omega, 1) \not\geqslant_{\mathsf{lex}} (\omega, 2) = [\mathsf{f}(\mathsf{a})]_{\mathcal{O} \times \mathcal{N}}$.

[5] Item (3) can also be seen as a lexicographic combination of two linear (polynomial) interpretations.

However, weak monotonicity of a lexicographic interpretation can be partially recovered: If both $f_{\mathcal{O}}(\overline{x})$ and $f_{\mathcal{A}}(\overline{x})$ are weakly monotone this also holds for the lexicographic combination $(f_{\mathcal{O}}(\overline{x}), f_{\mathcal{A}}(\overline{x}))$ provided that an argument x_i is ignored in $f_{\mathcal{A}}(\overline{x})$ whenever $f_{\mathcal{O}}(\overline{x})$ is not strictly—but still weakly—monotone with respect to x_i. This fact was already exploited in the termination proof by Touzet (see Example 30) and is also used in the proof of Theorem 15.

Example 27. For the interpretations $\mathsf{g}_{\mathcal{O}}(x) = x + \omega$ and $\mathsf{g}_{\mathcal{N}}(x) = 1$ the lexicographic combination $\mathsf{g}_{\mathcal{O}\times\mathcal{N}}((x, k)) = (x+\omega, 1)$ is weakly monotone. Similarly, for $\mathsf{h}_{\mathcal{O}}(x, y) = y+x$ and $\mathsf{h}_{\mathcal{N}}(x, y) = x+1$ also $\mathsf{h}_{\mathcal{O}\times\mathcal{N}}((x, k), (y, m)) = (y+x, k+1)$ is weakly monotone (note that $h_{\mathcal{O}\times\mathcal{N}}$ is strictly monotone in its first argument).

Hence we have to encode monotonicity of an ROE $f(\overline{x})$ in its i-th argument. We set

$$\mathrm{mon}_i(f(\overline{x})) = \big(f_i > 0 \wedge (\bigwedge_{i<j\leqslant n} f_j = 0) \wedge f_\omega = 0\big) \vee \hat{f}_i > 0 \vee (f_\omega > 0 \wedge \mathrm{mon}_i(f'(\overline{x})))$$

$$\mathrm{mon}(f(\overline{x})) = \bigwedge_{1\leqslant i\leqslant n} \mathrm{mon}_i(f(\overline{x}))$$

Then it follows that $\neg\mathrm{mon}(f(\overline{x}))$ $(\neg\mathrm{mon}_i(f(\overline{x})))$ holds whenever $f(\overline{x})$ is not strictly monotone (in its i-th argument). In the implementation we consider relative rewriting and add a rule $f'(\pi(x_1, \ldots, x_n)) \to f(x_1, \ldots, x_n)$ in the relative part whenever $\mathrm{mon}(f(\overline{x}))$ is not satisfied. Here f' is a fresh function symbol and $\pi(x_1, \ldots, x_n)$ returns the variables $x_{i_1}, \ldots, x_{i_m}$ for $1 \leqslant i_1 \leqslant \cdots \leqslant i_m \leqslant n$ in which $f(\overline{x})$ is strictly monotone, i.e., $\mathrm{mon}_{i_j}(f(\overline{x}))$ is satisfied. In presence of a rule $f'(\pi(x_1, \ldots, x_n)) \to f(x_1, \ldots, x_n)$, compatible interpretation functions $f_{\mathcal{A}}$ cannot depend on variables $x_j \notin \pi(x_1, \ldots, x_n)$. The idea is demonstrated by the following examples.

Example 28 (Example 26 revisited)**.** Consider the TRS $\mathcal{R}$ from Example 26. After applying the first interpretation we obtain the relative TRS $\{\mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{b})\}/\{\mathsf{f}' \to \mathsf{f}(x)\}$. Although this system is terminating there is no compatible interpretation since f may not depend on its arguments due to the second rule.

Example 29 (Example 27 revisited)**.** If $\mathsf{h}_{\mathcal{O}}(x_1, x_2) = x_2+x_1$ then $\mathrm{mon}_1(\mathsf{h}_{\mathcal{O}}(x_1, x_2)) = \top$ and $\mathrm{mon}_2(\mathsf{h}_{\mathcal{O}}(x_1, x_2)) = \bot$. Hence $\pi(x_1, x_2) = x_1$ and subsequent interpretations have to orient $\mathsf{h}'(x_1) \to \mathsf{h}(x_1, x_2)$ weakly and cannot depend on h's second argument while e.g., $\mathsf{h}_{\mathcal{A}}(x_1, x_2) = x_1 + 1$ is possible.

However, adding rules $f'(\pi(x_1, \ldots, x_n)) \to f(x_1, \ldots, x_n)$ is likely to disable the orientation of rules whose left-hand sides are rooted by f (to satisfy $[\alpha]_{\mathcal{A}}(f'(\pi(x_1, \ldots, x_n))) \geqslant [\alpha]_{\mathcal{A}}(f(x_1, \ldots, x_n))$ the interpretation of f may not depend on arguments x_i which do not occur in $\pi(x_1, \ldots, x_n)$) and consequently the termination proof might not be successful. To avoid this situation in the implementation we add constraints demanding to orient such rules only if the interpretation of f is not strictly monotone. Then rules rooted with f must be oriented before a rule $f'(\pi(x_1, \ldots, x_n)) \to f(x_1, \ldots, x_n)$ is added.

Another necessary requirement is that the (lexicographic) algebra is simple. Again we avoid an explicit lexicographic encoding. Rather, in a preprocessing step for every

$f \in \mathcal{F}$ we add the embedding rules $f(x_1, \ldots, x_n) \to x_i$ (for $1 \leqslant i \leqslant n$) into the relative component of the TRS. This then ensures $[\alpha]_{\mathcal{A}}(f(x_1, \ldots, x_n)) \geqslant [\alpha]_{\mathcal{A}}(x_i)$ for each $1 \leqslant i \leqslant n$.

All in all, for a TRS $\mathcal{R}$ over a signature $\mathcal{F}$ we execute the following procedure:

$\mathcal{S} := \{f(x_1, \ldots, x_n) \to x_i \mid f \in \mathcal{F}$ has arity n and $1 \leqslant i \leqslant n\}$
while $\mathcal{R} \neq \varnothing$ do
  find an algebra $\mathcal{A}$ satisfying $\mathcal{R} \cup \mathcal{S} \subseteq \geqslant_{\mathcal{A}}$ and $(\mathcal{R} \cup \mathcal{S}) \cap >_{\mathcal{A}} \neq \varnothing$
  $\mathsf{Nmon}_{\mathcal{A}}(\mathcal{R}) := \{f'(\pi(x_1, \ldots, x_n)) \to f(x_1, \ldots, x_n) \mid f \in \mathcal{F}$ has arity n, $1 \leqslant i \leqslant n$, and
   $x_i \notin \pi(x_1, \ldots, x_n)$ if $f_{\mathcal{A}}$ is not strictly monotone in i-th argument$\}$
  $\mathcal{R} := \mathcal{R} \setminus >_{\mathcal{A}}$ and $\mathcal{S} := (\mathcal{S} \setminus >_{\mathcal{A}}) \cup \mathsf{Nmon}_{\mathcal{A}}(\mathcal{R})$
report *terminating*

Instead of proving termination of $\mathcal{R}$ we try to establish termination of $\mathcal{R}$ relative to $\mathcal{S}$. This pre-processing step ensures that the algebras constructed in the body of the while loop are simple. We use SMT to find appropriate ROEs and matrix interpretations (of different dimensions), respectively. If no suitable algebra is found, the while loop is aborted and the procedure fails. Adding $\mathsf{Nmon}_{\mathcal{A}}(\mathcal{R})$ to the relative part ensures that the lexicographic combination of the employed algebras is weakly monotone.

4.2.3. Compatible Variable Orders

When interpreting or comparing terms we might get ROEs not having the same variable order. E.g., the rule $\mathsf{s}(\mathsf{g}(x, y)) \to \mathsf{g}(y, x)$ results in the constraint $x + y + 1 > y + x$, if $\mathsf{g}_{\mathcal{O}}(x, y) = x + y$ and $\mathsf{s}_{\mathcal{O}}(x) = x + 1$. The assignment $\alpha(x) = 1$ and $\alpha(y) = \omega$ yields $1+\omega+1 = \omega+1 \not> \omega+1$ but according to Definition 18 the constraint $[x+y+1 > y+x]$ is valid. The same effect also happens in arithmetic operations, e.g., the overapproximation of $+$ in Lemma 24(d). Taking $\mathsf{f}_{\mathcal{O}}(x, y) = \mathsf{g}_{\mathcal{O}}(x, y) = x+y$ with $\alpha(x) = 1$ and $\alpha(y) = \omega$, the term $\mathsf{f}(\mathsf{g}(x, y), \mathsf{g}(y, x))$ evaluates to $(1+\omega)+(\omega+1) = \omega 2+1$ but the overapproximation based on the variable order $[x, y]$ yields $2+\omega 2 = \omega 2$. Clearly $\omega 2+1 \not\leqslant \omega 2$. Hence we have to ensure our global assumption that two ROEs have *compatible* variable orders (in the standard addition part) when comparing, composing, or adding them. Let $\sum_{1 \leqslant i \leqslant n} x_i f_i$ and $\sum_{1 \leqslant i \leqslant n} y_i g_i$ be ordinal expressions over the same variables (so $\overline{y}$ is a permutation of $\overline{x}$). Let $i < j$. Two variables x_i and x_j are not compatible if there exist i' and j' with $1 \leqslant i' < j' \leqslant n$ such that $x_i = y_{j'}$, $x_j = y_{i'}$ and $f_i, f_j, g_{i'}, g_{j'}$ are positive. In such a case we constrain one of the coefficients to be zero, i.e., $f_i = 0 \vee f_j = 0 \vee g_{i'} = 0 \vee g_{j'} = 0$. For example consider $e_1 = x_1 1 + x_2 1$, $e_2 = x_2 1 + x_1 1$, and $e_3 = x_2 1 + x_1 0$. Then e_1 and e_2 do not have compatible variable orders while e_1 and e_3 do.

4.2.4. Efficiency

While the implementation fixes some initial depth d for the interpretation of function symbols, this depth increases when evaluating terms (when approximating compositions $f(g_1(\overline{x}), \ldots, g_n(\overline{x}))$). Not surprisingly, for efficiency it is necessary to bound the depth of expressions occurring in evaluations of terms. Dropping parts of an interpretation is sound as an underapproximation while for the overapproximation we add constraints (to the SMT solver) that the dropped part must evaluate to zero.

We have implemented the search for suitable ROEs in $\mathsf{T_TT_2}$ (Korp et al., 2009) (see Section 6 for the global setup). For the automatic termination proof of the TRS $\mathcal{G}$ in $\mathsf{T_TT_2}$ we (lexicographically) combine ordinal algebras with matrix interpretations (Endrullis et al., 2008). Then, $\mathsf{T_TT_2}$ manages $\mathcal{G}$ within nine seconds when using depth 1 for interpreting function symbols and limiting the depth of evaluations to 2. The CNF of the underlying SAT problem has approximately 120,000 variables and 300,000 clauses.

In their influential paper Kirby and Paris (1982) also presented the battle of Hercules and Hydra as a combinatorial game on trees. Generalizations of the Hydra battle are found in many papers (Fleischer (2009) contains a nice survey) and several different encodings of the battle into a termination problem of a specific TRS can be found in the literature (Buchholz, 2006; Dershowitz, 1993; Dershowitz and Moser, 2007; Dershowitz and Jouannaud, 1990; Lepper, 2004; Touzet, 1998). Not all of these TRSs faithfully model the battle, and termination of some of them is not independent of Peano arithmetic.

Example 30. Touzet (1998) presents the following TRS $\mathcal{H}$ to describe the battle between Hercules and Hydra for starting terms corresponding to ordinals $\alpha < \omega^{\omega^\omega}$ and using the *standard strategy*:

$$
\begin{array}{lll}
\circ\, x \to \bullet \, [\!]\, x & \mathsf{H}(\mathsf{0}, x) \to \circ\, x & \bullet\, \mathsf{c}^1(x,y) \to \mathsf{c}^1(x, \mathsf{H}(x,y)) \\
\bullet\, [\!]\, x \to [\!]\, \bullet\bullet\, x & \bullet\, \mathsf{H}(\mathsf{H}(\mathsf{0},y),z) \to \mathsf{c}^1(y,z) & \bullet\, \mathsf{c}^2(x,y,z) \to \mathsf{c}^2(x, \mathsf{H}(x,y), z) \\
[\!]\, \circ\, x \to \circ\, [\!]\, x & \bullet\, \mathsf{H}(\mathsf{H}(\mathsf{H}(\mathsf{0},x),y),z) \to \mathsf{c}^2(x,y,z) & \mathsf{c}^1(y,z) \to \circ\, z \\
\bullet\, x \to x & \mathsf{c}^2(x,y,z) \to \circ\, \mathsf{H}(y,z) &
\end{array}
$$

So far all termination tools failed on this example whose derivational complexity cannot be bounded by a multiple recursive function. Its termination can be shown by the following simple and weakly monotone interpretation $\mathcal{A}$ over the domain $\mathbb{O} \times \mathbb{N} \times \mathbb{N}$, where $f(x,y) = y + \omega^{x+1}$ (Touzet, 1998):

$$
\begin{array}{rl}
\mathsf{0}_{\mathcal{A}} = (0,0,0) & [\!]_{\mathcal{A}}(x,m,n) = (x, 2m+2, n) \\
\mathsf{H}_{\mathcal{A}}((x,m,n),(y,k,l)) = (\omega^x \oplus y, 0, 0) & \circ_{\mathcal{A}}(x,m,n) = (x, 2m+3, n) \\
\mathsf{c}^1_{\mathcal{A}}((x,m,n),(y,k,l)) = (f(x,y), 0, 0) & \bullet_{\mathcal{A}}(x,m,n) = (x, m, n+m+1) \\
\mathsf{c}^2_{\mathcal{A}}((x,m,n),(y,k,l),(z,i,j)) = (\omega^{f(x,y)} \oplus z, 0, 0) &
\end{array}
$$

Compared to $\mathcal{G}$, $\mathsf{T_TT_2}$ requires more resources (initial depth 2, intermediate depth 3, 12 seconds, 160,000 variables, 410,000 clauses) to automatically prove termination of $\mathcal{H}$. This is surprising as the derivational complexity of $\mathcal{G}$ far exceeds that of the Hydra system $\mathcal{H}$, which is bounded by the Hardy function $H_{\omega^{\omega^\omega}}$.

Example 31. Beklemishev (2006) presents two infinite TRSs and one finite TRS describing the Worm battle (corresponding to a one-dimensional version of Buchholz' Hydra battle (Buchholz, 1987), first introduced by Hamano and Okada (1997)). The second infinite TRS $\mathcal{W}_2$ consists of the rules

$$
(x \cdot y) \cdot z \to x \cdot (y \cdot z) \qquad \mathsf{f}(\mathsf{0}) \to \mathsf{0}^m \qquad \mathsf{f}(\mathsf{0} \cdot x) \to (\mathsf{0} \cdot \mathsf{f}(x))^m
$$

for $m \geqslant 1$, where t^m abbreviates the term $t \cdot (\cdots (t \cdot (t \cdot t)) \cdots)$ with m copies of t. The ROE algebra $\mathsf{0}_{\mathcal{O}} = 1$, $f_{\mathcal{O}}(x) = \omega^x$, and $x \cdot_{\mathcal{O}} y = 2x \oplus y \oplus 1$ is weakly monotone and simple on $\mathbb{O}$ and orients $\mathcal{W}_2$:

$$4x \oplus 2y \oplus z \oplus 3 > 2x \oplus 2y \oplus z \oplus 2$$
$$\omega > 3m - 2$$
$$\omega^{x \oplus 3} > \omega^x(2m-1) \oplus (7m-4)$$

The finite system $\mathcal{W}_3'$ to simulate the Worm sequence consists of the following rules: [6]

$$\begin{array}{rrcl@{\qquad}rrcl}
1\colon & (x \cdot y) \cdot z & \to & x \cdot (y \cdot z) & 2\colon & \mathsf{f}(0 \cdot x) & \to & \mathsf{b}(0 \cdot \mathsf{f}(x)) \\
3\colon & \mathsf{f}(0) & \to & \mathsf{b}(0) & 4\colon & \mathsf{a}(\mathsf{f}(x)) & \to & \mathsf{f}(\mathsf{a}(x)) \\
5\colon & \mathsf{a}(x \cdot y) & \to & \mathsf{a}(x) \cdot y & 6\colon & \mathsf{a}(\mathsf{b_1}(x)) & \to & \mathsf{b_1}(\mathsf{a}(x)) \\
7\colon & \mathsf{f}(\mathsf{b}(x)) & \to & \mathsf{b}(\mathsf{f}(x)) & 8\colon & \mathsf{b}(x) \cdot y & \to & \mathsf{b}(x \cdot y) \\
9\colon & \mathsf{a}(\mathsf{f}(0 \cdot x)) & \to & \mathsf{b_1}(\mathsf{f}(0 \cdot x) \cdot (0 \cdot \mathsf{f}(x))) & 10\colon & \mathsf{a}(\mathsf{f}(0)) & \to & \mathsf{b_1}(\mathsf{f}(0) \cdot 0) \\
11\colon & \mathsf{b_1}(\mathsf{b}(x)) & \to & \mathsf{b}(\mathsf{b}(x)) & 12\colon & \mathsf{c}(\mathsf{b}(x)) & \to & \mathsf{c}(\mathsf{a}(x)) \\
13\colon & \mathsf{a}(\mathsf{b}(x)) & \to & \mathsf{b}(\mathsf{a}(x)) & 14\colon & \mathsf{a}(0 \cdot x) & \to & \mathsf{b}(\mathsf{b}(x))
\end{array}$$

Consider the algebra $\mathcal{A}$ on $\mathbb{O}_{>0} \times \mathbb{N} \times \mathbb{N}$:

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{c}_{\mathcal{A}}(x,m,n) & = & (x+1, 2m+2, 2m) & \mathsf{0}_{\mathcal{A}} & = & (1,0,0) \\
\mathsf{b}_{\mathcal{A}}(x,m,n) & = & (x, m+1, n) & \mathsf{b_1}_{\mathcal{A}}(x,m,n) & = & (x, 2m, n+1) \\
\mathsf{a}_{\mathcal{A}}(x,m,n) & = & (x, m, m+2n) & \mathsf{f}_{\mathcal{A}}(x,m,n) & = & (\omega^x, m, m+3) \\
(x,m,n) \cdot_{\mathcal{A}} (y,k,l) & = & (y+x, m, m+n+1) & & &
\end{array}$$

This algebra is simple (note that $(x,m,n) \cdot_{\mathcal{A}} (y,k,l) \geqslant (y,k,l)$ since $x \neq 0$), weakly monotone, and orients all rules of $\mathcal{W}_3'$:

$$\begin{array}{rrcl}
1\colon & (z+y+x, m, 2m+n+2) & > & (z+y+x, m, m+n+1) \\
2\colon & (\omega^{x+1}, 0, 3) & > & (\omega^x + 1, 1, 1) \\
3\colon & (\omega, 0, 3) & > & (1,1,0) \\
4\colon & (\omega^x, m, 3m+6) & > & (\omega^x, m, m+3) \\
5\colon & (y+x, m, 3m+2n+2) & > & (y+x, m, 2m+2n+1) \\
6\colon & (x, 2m, 2m+2n+2) & > & (x, 2m, m+2n+1) \\
7\colon & (\omega^x, m+1, m+4) & > & (\omega^x, m+1, m+3) \\
8\colon & (y+x, m+1, m+n+2) & > & (y+x, m+1, m+n+1) \\
9\colon & (\omega^{x+1}, 0, 6) & > & (\omega^x + 1 + \omega^{x+1}, 0, 5) \\
10\colon & (\omega, 0, 6) & > & (\omega, 0, 5) \\
11\colon & (x, 2m+2, n+1) & > & (x, m+2, n) \\
12\colon & (x+1, 2m+4, 2m+2) & > & (x+1, 2m+2, 2m) \\
13\colon & (x, m+1, m+2n+1) & > & (x, m+1, m+2n) \\
14\colon & (x+1, 0, 2) & > & (x, m+2, n)
\end{array}$$

for all $x, y, z \in \mathbb{O}_{>0}$ and $m, n, k, l \in \mathbb{N}$. Thus this algebra shows termination of $\mathcal{W}_3'$ by Theorem 6.

---

[6] Note that we added rule (14) to the TRS $\mathcal{W}_3$ originally presented in (Beklemishev, 2006) since personal communication with Lev Beklemishev revealed that such an additional rule is in fact required to faithfully model the worm sequence. We believe the derivational complexity of $\mathcal{W}_3$ to be actually smaller than that of $\mathcal{W}_2$ and $\mathcal{W}_3'$, which is also supported by the fact that termination of $\mathcal{W}_3$ can be shown by $\mathsf{T_TT_2}$ with standard techniques.

The termination proof from the above example cannot be reproduced within $\mathsf{T_TT_2}$, since the interpretation function of $\cdot$ is simple on the carrier $\mathbb{O}_{>0} \times \mathbb{N}^2$ but not on $\mathbb{O} \times \mathbb{N}^2$, which would be used by $\mathsf{T_TT_2}$. However, $\mathsf{T_TT_2}$ succeeds in the dependency pair setting where it manages the crucial SCC by lexicographically combining an ROE algebra of degree 2 with a linear interpretation. The overall time is about four seconds while the CNF of the underlying SAT problem has approximately 87,000 variables and 205,000 clauses.

5. Automation of Elementary Algebras

Similar to ordinal algebras (Section 4), we give encodings of elementary interpretation functions (Section 5.1) before implementation aspects are addressed in Section 5.2 and examples (and limitations) are discussed in Section 5.3.

The shape of FBIs (see below) suffices to go beyond polynomial interpretations, which fail on Examples 33 and 34.[7] Furthermore, a fixed base allows to use more powerful approximations of comparisons/arithmetic.

Definition 32. A *fixed-base elementary interpretation function* (FBI) of depth 0 is

$$f(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i f_i + f_0$$

and an FBI of depth $d+1$ is

$$f(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i f_i + f_0 + b^{f'(\overline{x})}\left(\sum_{1 \leqslant i \leqslant n} x_i \hat{f}_i + \hat{f}_0\right) \tag{6}$$

where $f_0, f_1, \ldots, f_n, \hat{f}_0, \hat{f}_1, \ldots, \hat{f}_n$ are naturals, $f'(\overline{x})$ is an FBI of depth d, and $b \geqslant 2$ is a fixed natural number. Throughout this section we use the following abbreviations:

$$\dot{f}(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i f_i + f_0 \qquad\qquad \hat{f}(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i \hat{f}_i + \hat{f}_0$$

An *FBI algebra* has $\mathbb{N}_{\geqslant 1}$ as carrier and FBIs as interpretation functions for all function symbols in the signature.

It is known that for polynomial interpretations the carriers $\mathbb{N}$ and $\mathbb{N}_{\geqslant \mu}$ admit the same termination proving power for any $\mu \in \mathbb{N}$ (see e.g. TeReSe (2003); Contejean et al. (2005)). However, the situation is different for FBI's. The function $2^x y$ is not monotone on $\mathbb{N}$ but it is on $\mathbb{N}_{\geqslant 1}$. The typical transformation converts $2^x y$ into the function $2^{x+1}(y+1) - 1$, but the latter does not admit an FBI representation. The following examples show the usefulness of $2^x y$, so we restrict to the carrier $\mathbb{N}_{\geqslant 1}$ in the sequel.

Example 33. Termination of Lescanne's factorial example (Lescanne, 1995)

$$\begin{array}{ccc}
\mathsf{0} + x \to x & \mathsf{0} \cdot x \to \mathsf{0} & \mathsf{fact}(\mathsf{0}) \to \mathsf{s}(\mathsf{0}) \\
\mathsf{s}(x) + y \to \mathsf{s}(x+y) & \mathsf{s}(x) \cdot y \to x \cdot y + y & \mathsf{fact}(\mathsf{s}(x)) \to \mathsf{s}(x) \cdot \mathsf{fact}(x) \\
x \cdot (y+z) \to x \cdot y + x \cdot z & &
\end{array}$$

[7] We remark that establishing termination of these systems becomes much easier when using dependency pairs but then totality of the order is lost, which is essential for applications such as ordered completion.

can be shown by an FBI algebra $\mathcal{A}$ of depth 2 with base $b = 2$ and interpretation functions $\mathsf{0}_\mathcal{A} = 2$, $\mathsf{s}_\mathcal{A}(x) = x + 2$, $x +_\mathcal{A} y = 2x + y + 1$, $x \cdot_\mathcal{A} y = 2^x y$, and $\mathsf{fact}_\mathcal{A}(x) = 2^{2^x}$. We have

$$
\begin{array}{ccc}
x + 5 > x & 2^2 x > 2 & 2^{2^2} > 4 \\
2x + y + 5 > 2x + y + 3 & 2^{x+2} y > 2^{x+1} y + y + 1 & 2^{2^{x+2}} > 2^{x+2} 2^{2^x} = 2^{x+2+2^x} \\
2^x(2y + z + 1) > 2^{x+1} y + 2^x z + 1 & &
\end{array}
$$

for all $x, y, z \geqslant 1$.

Example 34. Termination of Lucas' factorial example (Lucas, 2009)

$$
\begin{array}{ccc}
x + \mathsf{0} \to x & \mathsf{0} \cdot x \to \mathsf{0} & \mathsf{fact}(\mathsf{0}) \to \mathsf{s}(\mathsf{0}) \\
x + \mathsf{s}(y) \to \mathsf{s}(x + y) & \mathsf{s}(x) \cdot y \to x \cdot y + y & \mathsf{fact}(\mathsf{s}(x)) \to \mathsf{s}(x) \cdot \mathsf{fact}(x)
\end{array}
$$

can also be shown by an FBI algebra $\mathcal{A}$ of depth 2 with base $b = 2$ and interpretation functions $\mathsf{0}_\mathcal{A} = 2$, $\mathsf{s}_\mathcal{A}(x) = x + 2$, $x +_\mathcal{A} y = x + 2y + 1$, $x \cdot_\mathcal{A} y = 2^x y$, and $\mathsf{fact}_\mathcal{A}(x) = 2^{2^x}$. We have

$$
\begin{array}{ccc}
x + 5 > x & 2^2 x > 2 & 2^{2^2} > 4 \\
x + 2y + 5 > x + 2y + 3 & 2^{x+2} y > 2^x y + 2y + 1 & 2^{2^{x+2}} > 2^{x+2} 2^{2^x} = 2^{x+2+2^x}
\end{array}
$$

for all $x, y \geqslant 1$.

In the sequel we sometimes treat an FBI $f(\overline{x})$ of depth 0 as $\sum_{1 \leqslant i \leqslant n} x_i f_i + f_0 + b^0 0$ to avoid case distinctions.

5.1. *Encodings*

Let $f(\overline{x})$ and $g(\overline{x})$ be FBIs of the form

$$
\begin{aligned}
f(\overline{x}) &= \sum_{1 \leqslant i \leqslant n} x_i f_i + f_0 + b^{f'(\overline{x})} \left(\sum_{1 \leqslant i \leqslant n} x_i \hat{f}_i + \hat{f}_0 \right) \\
g(\overline{x}) &= \sum_{1 \leqslant i \leqslant n} x_i g_i + g_0 + b^{g'(\overline{x})} \left(\sum_{1 \leqslant i \leqslant n} x_i \hat{g}_i + \hat{g}_0 \right)
\end{aligned}
\tag{7}
$$

5.1.1. *Useful Abbreviations*

First we introduce lower and upper bounds for two FBIs:

$$
\begin{aligned}
\mathrm{fmin}(f, g)(\overline{x}) = {} & \sum_{1 \leqslant i \leqslant n} x_i \min(f_i, g_i) + \min(f_0, g_0) \\
& + b^{\mathrm{fmin}(f', g')(\overline{x})} \left(\sum_{1 \leqslant i \leqslant n} x_i \min(\hat{f}_i, \hat{g}_i) + \min(\hat{f}_0, \hat{g}_0) \right) \\
\mathrm{fmax}(f, g)(\overline{x}) = {} & \sum_{1 \leqslant i \leqslant n} x_i \max(f_i, g_i) + \max(f_0, g_0) \\
& + b^{\mathrm{fmax}(f', g')(\overline{x})} \left(\sum_{1 \leqslant i \leqslant n} x_i \max(\hat{f}_i, \hat{g}_i) + \max(\hat{f}_0, \hat{g}_0) \right)
\end{aligned}
$$

Again we introduce the notation of contribution of a variable x_i to $f(\overline{x})$, which we denote by $\mathrm{con}_i(f(\overline{x}))$:[8]

$$\mathrm{con}_i(f(\overline{x})) = f_i > 0 \vee \hat{f}_i > 0 \vee (\mathrm{con}_i(f'(\overline{x})) \wedge \hat{f}(\overline{x}) > 0)$$

5.1.2. Comparisons

The following recursive definition reduces the comparison of FBIs to the comparison of non-linear polynomials. The latter can be compared by the absolute positiveness approach, see Hong and Jakuš (1998). For comparing polynomials we take the carrier $\mathbb{N}_{\geqslant 1}$ into account such that e.g. $3y \geqslant y + 1$ evaluates to true.

Definition 35. Let $f(\overline{x})$ and $g(\overline{x})$ be FBIs as in (7). Let

$$\lfloor b^{f'(\overline{x})} \rfloor = \big((\dot{f}'(\overline{x}) + \hat{f}'(\overline{x}) = 0) \mathbin{?} 1 : b\,(\dot{f}'(\overline{x}) + \hat{f}'(\overline{x}))\big)$$

Note that $b^{f(\overline{x})} \geqslant \lfloor b^{f(\overline{x})} \rfloor$. Furthermore let $p(\overline{x}) = \dot{f}'(\overline{x}) + \hat{f}'(\overline{x}) - \dot{g}'(\overline{x}) - \hat{g}'(\overline{x})$ and $h(\overline{x}) = \lfloor b^{p(\overline{x})} \rfloor \hat{f}(\overline{x}) - \hat{g}(\overline{x})$. We set

$$\begin{aligned}
[f(\overline{x}) \geqslant g(\overline{x})] \;=\; & (\hat{g}(\overline{x}) > 0 \rightarrow [f'(\overline{x}) \geqslant g'(\overline{x})]) \;\wedge\; \big(\\
& (\hat{f}(\overline{x}) > 0 \wedge [f'(\overline{x})\, b \geqslant g(\overline{x})]) \;\vee && \text{(a)} \\
& (\dot{f}(\overline{x}) \geqslant \dot{g}(\overline{x}) \wedge \hat{f}(\overline{x}) \geqslant \hat{g}(\overline{x})) \;\vee && \text{(b)} \\
& (h(\overline{x}) \geqslant 0 \wedge p(\overline{x}) \geqslant 0 \wedge \hat{f}'(\overline{x}) \geqslant \hat{g}'(\overline{x}) \wedge \\
& \quad \dot{f}(\overline{x}) + \lfloor b^{g'(\overline{x})} \rfloor \lfloor b^{p(\overline{x})} \rfloor \hat{f}(\overline{x}) \geqslant \dot{g}(\overline{x}) + \lfloor b^{g'(\overline{x})} \rfloor \hat{g}(\overline{x})) \;\big) && \text{(c)} \\
[f(\overline{x}) > g(\overline{x})] \;=\; & (\hat{g}(\overline{x}) > 0 \rightarrow [f'(\overline{x}) \geqslant g'(\overline{x})]) \;\wedge\; \big(\\
& (\hat{f}(\overline{x}) > 0 \wedge [f'(\overline{x})\, b > g(\overline{x})]) \;\vee && \text{(d)} \\
& (\dot{f}(\overline{x}) \geqslant \dot{g}(\overline{x}) \wedge \hat{f}(\overline{x}) \geqslant \hat{g}(\overline{x}) \wedge \\
& \quad ((\hat{f}(\overline{x}) > 0 \wedge [f'(\overline{x}) > g'(\overline{x})]) \vee \dot{f}(\overline{x}) > \dot{g}(\overline{x}) \vee \hat{f}(\overline{x}) > \hat{g}(\overline{x}))) \;\vee && \text{(e)} \\
& (h(\overline{x}) \geqslant 0 \wedge p(\overline{x}) \geqslant 0 \wedge \hat{f}'(\overline{x}) \geqslant \hat{g}'(\overline{x}) \wedge \\
& \quad \dot{f}(\overline{x}) + \lfloor b^{g'(\overline{x})} \rfloor \lfloor b^{p(\overline{x})} \rfloor \hat{f}(\overline{x}) > \dot{g}(\overline{x}) + \lfloor b^{g'(\overline{x})} \rfloor \hat{g}(\overline{x})) \;\big) && \text{(f)}
\end{aligned}$$

The difference between $[f(\overline{x}) \geqslant g(\overline{x})]$ and $[f(\overline{x}) > g(\overline{x})]$ is that in the latter we demand at least one strict decrease. The following example shows that our encodings of comparisons are very accurate.

Example 36. The encoding of $[2^{x+1} > 1 + 2^x]$ evaluates to true. The only interesting case is (f) where $p(\overline{x}) = 1$ and $\lfloor 2^x \rfloor \lfloor 2^1 \rfloor = 4x > 1 + 2x = 1 + \lfloor 2^x \rfloor$, i.e., $2x > 1$, which holds for all $x \in \mathbb{N}_{\geqslant 1}$.

The encodings of comparisons are sound.

Lemma 37. *Let $f(\overline{x})$ and $g(\overline{x})$ be FBIs as in* (7).
(a) *If $[f(\overline{x}) > g(\overline{x})]$ then $[\alpha](f(\overline{x})) > [\alpha](g(\overline{x}))$ for all assignments α.*
(b) *If $[f(\overline{x}) \geqslant g(\overline{x})]$ then $[\alpha](f(\overline{x})) \geqslant [\alpha](g(\overline{x}))$ for all assignments α.*

[8] Here $\hat{f}(\overline{x}) > 0$ tests the linear polynomial $\hat{f}(\overline{x})$ for positiveness.

Proof. We only show (b) the argument for (a) is similar. Case (a) in Definition 35 approximates the situation when $b^{f'(\overline{x})}\hat{f}(\overline{x}) \geqslant g(\overline{x})$ and case (b) is obvious. Finally, case (c) follows from the argument below. Let $p(\overline{x})$ and $h(\overline{x})$ be as in Definition 35. For the moment assume $f'(\overline{x}) \geqslant p(\overline{x}) + g'(\overline{x})$. Then

$$\begin{aligned}
& \dot{f}(\overline{x}) + b^{f'(\overline{x})}\hat{f}(\overline{x}) \geqslant \dot{g}(\overline{x}) + b^{g'(\overline{x})}\hat{g}(\overline{x}) \\
\Longleftarrow \quad & \dot{f}(\overline{x}) + b^{p(\overline{x})+g'(\overline{x})}\hat{f}(\overline{x}) \geqslant \dot{g}(\overline{x}) + b^{g'(\overline{x})}\hat{g}(\overline{x}) \\
\Longleftrightarrow \quad & \frac{\dot{f}(\overline{x})}{b^{g'(\overline{x})}} + b^{p(\overline{x})}\hat{f}(\overline{x}) \geqslant \frac{\dot{g}(\overline{x})}{b^{g'(\overline{x})}} + \hat{g}(\overline{x}) \\
\Longleftrightarrow \quad & \frac{\dot{f}(\overline{x})}{b^{g'(\overline{x})}} + b^{p(\overline{x})}\hat{f}(\overline{x}) + h(\overline{x}) \geqslant \frac{\dot{g}(\overline{x})}{b^{g'(\overline{x})}} + \hat{g}(\overline{x}) + h(\overline{x}) \\
\Longleftarrow \quad & \frac{\dot{f}(\overline{x})}{b^{g'(\overline{x})}} + h(\overline{x}) \geqslant \frac{\dot{g}(\overline{x})}{b^{g'(\overline{x})}} \wedge b^{p(\overline{x})}\hat{f}(\overline{x}) \geqslant \hat{g}(\overline{x}) + h(\overline{x}) \\
\Longleftrightarrow \quad & \dot{f}(\overline{x}) + b^{g'(\overline{x})}h(\overline{x}) \geqslant \dot{g}(\overline{x}) \wedge b^{p(\overline{x})}\hat{f}(\overline{x}) \geqslant \hat{g}(\overline{x}) + h(\overline{x}) \\
\Longleftarrow \quad & h(\overline{x}) \geqslant 0 \wedge \dot{f}(\overline{x}) + \lfloor b^{g'(\overline{x})}\rfloor h(\overline{x}) \geqslant \dot{g}(\overline{x}) \wedge \lfloor b^{p(\overline{x})}\rfloor \hat{f}(\overline{x}) \geqslant \hat{g}(\overline{x}) + h(\overline{x}) \qquad (\star) \\
\Longleftrightarrow \quad & h(\overline{x}) \geqslant 0 \wedge \dot{f}(\overline{x}) + \lfloor b^{g'(\overline{x})}\rfloor h(\overline{x}) \geqslant \dot{g}(\overline{x}) \qquad (\dagger) \\
\Longleftrightarrow \quad & h(\overline{x}) \geqslant 0 \wedge \dot{f}(\overline{x}) + \lfloor b^{g'(\overline{x})}\rfloor \lfloor b^{p(\overline{x})}\rfloor \hat{f}(\overline{x}) \geqslant \dot{g}(\overline{x}) + \lfloor b^{g'(\overline{x})}\rfloor \hat{g}(\overline{x})
\end{aligned}$$

In the step $(\star)$ the non-negativity of $h(\overline{x})$ is used and the step $(\dagger)$ follows from the definition of $h(\overline{x})$. Finally we have to show that our assumption $f'(\overline{x}) \geqslant p(x) + g'(\overline{x})$ follows from the constraints. We observe

$$\begin{aligned}
& f'(\overline{x}) \geqslant p(\overline{x}) + g'(\overline{x}) \\
\Longleftrightarrow \quad & \dot{f}'(\overline{x}) + b^{f''(\overline{x})}\hat{f}'(\overline{x}) \geqslant \dot{f}'(\overline{x}) + \hat{f}'(\overline{x}) - \dot{g}'(\overline{x}) - \hat{g}'(\overline{x}) + \dot{g}'(\overline{x}) + b^{g''(\overline{x})}\hat{g}'(\overline{x}) \\
\Longleftrightarrow \quad & (b^{f''(\overline{x})} - 1)\hat{f}'(\overline{x}) \geqslant (b^{g''(\overline{x})} - 1)\hat{g}'(\overline{x}) \\
\Longleftarrow \quad & (\hat{g}'(\overline{x}) > 0 \to [f''(\overline{x}) \geqslant g''(\overline{x})]) \wedge \hat{f}'(\overline{x}) \geqslant \hat{g}'(\overline{x})
\end{aligned}$$

While the above proof does not rely on $p(\overline{x}) \geqslant 0$ this (redundant) constraint in Definition 35 might cut the search space. □

5.1.3. Composition

Similar as for ROEs, FBIs are not closed under addition and composition.

Example 38. The sum $2^x + 2^y$ of the FBIs 2^x and 2^y has no FBI representation. Also, substituting the FBI 2^y+1 for x in the FBI $2^x x$ results in $2^{2^y+1}(2^y+1) = 2^{2^y+y+1} + 2^{2^y+1}$, which also has no equivalent FBI representation.

We thus define under- and overapproximations for addition, multiplication, and composition.

Definition 39. Let $f(\overline{x})$ and $g(\overline{x})$ be FBIs as in (7).

(a) Multiplication of an FBI by a scalar again yields an FBI, i.e.

$$f(\overline{x})\, a = \sum_{1 \leqslant i \leqslant n} x_i f_i a + f_0 a + b^{f'(\overline{x})}\left(\sum_{1 \leqslant i \leqslant n} x_i \hat{f}_i a + \hat{f}_0 a\right)$$

(b) For addition we use fmin (fmax) to estimate a lower (upper) bound for both $f(\overline{x})$ and $g(\overline{x})$, and introduce approximations by FBIs as follows:

$$f(\overline{x}) +_\mu g(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i(f_i + g_i) + (f_0 + g_0) + b^{e_\mu(\overline{x})}\Bigg(\sum_{1 \leqslant i \leqslant n} x_i(\hat{f}_i + \hat{g}_i) + (\hat{f}_0 + \hat{g}_0)\Bigg)$$

$$f(\overline{x}) +_\nu g(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i(f_i + g_i) + (f_0 + g_0) + b^{e_\nu(\overline{x})}\Bigg(\sum_{1 \leqslant i \leqslant n} x_i(\hat{f}_i + \hat{g}_i) + (\hat{f}_0 + \hat{g}_0)\Bigg)$$

with $e_\mu(\overline{x})$ abbreviating $\hat{f}(\overline{x}) = 0 \ ?\ g'(\overline{x}) : \big(\hat{g}(\overline{x}) = 0 \ ?\ f'(\overline{x}) : \mathrm{fmin}(f', g')(\overline{x})\big)$ and $e_\nu(\overline{x})$ abbreviating $\hat{f}(\overline{x}) = 0 \ ?\ g'(\overline{x}) : \big(\hat{g}(\overline{x}) = 0 \ ?\ f'(\overline{x}) : \mathrm{fmax}(f', g')(\overline{x})\big)$.

(c) To approximate multiplication of an expression of the form $b^{g'(\overline{x})}$ with $f(\overline{x})$ by an FBI, we may use

$$b^{g'(\overline{x})} \cdot_\mu f(\overline{x}) = \hat{f}(\overline{x}) > 0 \ ?\ \dot{f}(\overline{x}) + b^{f'(\overline{x}) +_\mu g'(\overline{x})}\hat{f}(\overline{x}) : b^{g'(\overline{x})}\dot{f}(\overline{x})$$

$$b^{g'(\overline{x})} \cdot_\nu f(\overline{x}) = \hat{f}(\overline{x}) > 0 \ ?\ b^{f'(\overline{x}) +_\nu g'(\overline{x})}\Bigg(\sum_{1 \leqslant i \leqslant n} x_i(\hat{f}_i + f_i) + (\hat{f}_0 + f_0)\Bigg) : b^{g'(\overline{x})}\dot{f}(\overline{x})$$

(d) Finally we can give approximations for the composition $f(\overline{g})(\overline{x}) = f(g_1(\overline{x}), \ldots, g_n(\overline{x}))$:

$$f(\overline{g})_\mu(\overline{x}) = \sum_{1 \leqslant i \leqslant n}^{\mu} g_i(\overline{x}) f_i +_\mu f_0 +_\mu b^{f'(\overline{g})_\mu(\overline{x})} \cdot_\mu \Bigg(\sum_{1 \leqslant i \leqslant n}^{\mu} g_i(\overline{x})\hat{f}_i +_\mu \hat{f}_0\Bigg)$$

$$f(\overline{g})_\nu(\overline{x}) = \sum_{1 \leqslant i \leqslant n}^{\nu} g_i(\overline{x}) f_i +_\nu f_0 +_\nu b^{f'(\overline{g})_\nu(\overline{x})} \cdot_\nu \Bigg(\sum_{1 \leqslant i \leqslant n}^{\nu} g_i(\overline{x})\hat{f}_i +_\nu \hat{f}_0\Bigg)$$

(e) Let t be a term and $\mathcal{A}$ an FBI algebra. By induction on the term structure we define FBIs $\mu_\mathcal{A}(t)$ and $\nu_\mathcal{A}(t)$ such that

$$\mu_\mathcal{A}(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f_\mathcal{A}(\mu_\mathcal{A}(t_1), \ldots, \mu_\mathcal{A}(t_n))_\mu & \text{otherwise} \end{cases}$$

$$\nu_\mathcal{A}(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f_\mathcal{A}(\nu_\mathcal{A}(t_1), \ldots, \nu_\mathcal{A}(t_n))_\nu & \text{otherwise} \end{cases}$$

The following example illustrates Definition 35.

Example 40. We consider the cases for addition and multiplication.

(b) We have $\mathrm{fmin}(x+1, x) = x$ and $\mathrm{fmax}(x+1, x) = x+1$, thus $2^{x+1}y +_\mu 2^x(z+1) = 2^x(y+z+1)$ but $2^{x+1}y +_\nu 2^x(z+1) = 2^{x+1}(y+z+1)$.

In certain pathological cases the approximations of addition are not commutative. To be more precise, the resulting FBIs may be syntactically different but denote the same elementary function. For instance, $2^x \cdot 0 +_\mu 2^{x+1} \cdot 0 = 2^{x+1} \cdot 0$ while $2^{x+1} \cdot 0 +_\mu 2^x \cdot 0 = 2^x \cdot 0$. Still, we do not regard this a problem for our application as the encoding of comparisons takes these cases into account.

(c) For multiplication we have $2^{x+1} \cdot_\mu 2^{2^x} = 2^{(x+1) +_\mu 2^x} = 2^{x+1+2^x}$ and $2^{x+1} \cdot_\nu 2^{2^x} = 2^{x+1+2^x}$, the approximation is thus precise in these cases. On the other hand, as

$(x+1) +_\mu 2^x = (x+1) +_\nu 2^x = x+1+2^x$ we have $2^{x+1} \cdot_\mu (z+1+2^{2^x} y) = z+1+2^{x+1+2^x} y$, while $2^{x+1} \cdot_\nu (z+1+2^{2^x} y) = 2^{x+1+2^x}(y+z+1)$.

The following example shows that in practice our approximations are very accurate, i.e., for Examples 33 and 34 the approximations are exact.

Example 41. For Example 33 we get the following constraints

$$\begin{array}{ccc} x+5 > x & 2^2 x > 2 & 2^{2^2} > 4 \\ 2x+y+5 > 2x+y+3 & 2^{x+2} y > y+1+2^x 2y & 2^{2^{x+2}} > 2^{x+2+2^x} \\ 2^x(2y+z+1) > 1+2^x(2y+z) & & \end{array}$$

while Example 34 yields

$$\begin{array}{ccc} x+5 > x & 2^2 x > 2 & 2^{2^2} > 4 \\ x+2y+5 > x+2y+3 & 2^{x+2} y > 2y+1+2^x y & 2^{2^{x+2}} > 2^{x+2+2^x} \end{array}$$

We now show that Definition 39 yields valid over- and underapproximations.

Lemma 42. *Let $\mathcal{A}$ be an FBI algebra and t be a term. Then $[\alpha](\mu_\mathcal{A}(t)) \leqslant [\alpha]_\mathcal{A}(t) \leqslant [\alpha](\nu_\mathcal{A}(t))$ for all assignments α.*

Proof. We argue that all approximations in Definition 39 constitute valid lower and upper bounds. Let α be an arbitrary assignment.

(a) Since scalar multiplication is no approximation there is nothing to show.

(b) For $+_\mu$ (the reasoning for $+_\nu$ is analogous) one of the three cases applies:

$$f(\overline{x}) + g(\overline{x}) \geqslant \dot{f}(\overline{x}) + \dot{g}(\overline{x}) + \begin{cases} b^{g'(\overline{x})}\hat{g}(\overline{x}) & \text{if } \hat{f}(\overline{x}) = 0 \\ b^{f'(\overline{x})}\hat{f}(\overline{x}) & \text{if } \hat{g}(\overline{x}) = 0 \\ b^{h'(\overline{x})}(\hat{f}(\overline{x}) + \hat{g}(\overline{x})) & \text{if } [\alpha](f'(\overline{x})) \geqslant [\alpha](h'(\overline{x})) \\ & \text{and } [\alpha](g'(\overline{x})) \geqslant [\alpha](h'(\overline{x})) \end{cases}$$

(c) If $\hat{f}(\overline{x}) = 0$ then $b^{g'(\overline{x})} f(\overline{x}) = b^{g'(\overline{x})} \dot{f}(\overline{x})$ and if $\hat{f}(\overline{x}) > 0$ we obtain for $\cdot_\mu$

$$\begin{aligned} b^{g'(\overline{x})} \cdot f(\overline{x}) &= b^{g'(\overline{x})} \dot{f}(\overline{x}) + b^{f'(\overline{x})+g'(\overline{x})} \hat{f}(\overline{x}) \geqslant \dot{f}(\overline{x}) + b^{f'(\overline{x})+g'(\overline{x})} \hat{f}(\overline{x}) \\ &\geqslant \dot{f}(\overline{x}) + b^{f'(\overline{x}) +_\mu g'(\overline{x})} \hat{f}(\overline{x}) = b^{g'(\overline{x})} \cdot_\mu f(\overline{x}) \end{aligned}$$

while $\cdot_\nu$ is justified by

$$\begin{aligned} b^{g'(\overline{x})} \cdot f(\overline{x}) &= b^{g'(\overline{x})} \dot{f}(\overline{x}) + b^{f'(\overline{x})+g'(\overline{x})} \hat{f}(\overline{x}) \leqslant b^{f'(\overline{x})+g'(\overline{x})} \dot{f}(\overline{x}) + b^{f'(\overline{x})+g'(\overline{x})} \hat{f}(\overline{x}) \\ &= b^{f'(\overline{x})+g'(\overline{x})}(\dot{f}(\overline{x}) + \hat{f}(\overline{x})) \leqslant b^{f'(\overline{x}) +_\nu g'(\overline{x})}(\dot{f}(\overline{x}) + \hat{f}(\overline{x})) = b^{g'(\overline{x})} \cdot_\nu f(\overline{x}) \end{aligned}$$

(d) By (a)–(c) and weak monotonicity of addition, multiplication, and exponentiation.

(e) By induction on the term structure of t, using (d). □

5.1.4. Main Theorem

An FBI $f(\overline{x})$ is monotone if all variables x_i contribute to it. Monotonicity of $f(\overline{x})$ is thus expressed by

$$\mathrm{mon}(f(\overline{x})) = \bigwedge_{1 \leqslant i \leqslant n} \mathrm{con}_i(f(\overline{x}))$$

An FBI $f(\overline{x})$ is well-defined if $[f(\overline{x}) \geqslant 1]$ holds.

Finally we obtain the main result of this section.

Theorem 43. *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and $\mathcal{A}$ be an FBI algebra on $\mathcal{F}$. If*

$$\bigwedge_{\ell \to r \in \mathcal{R}} [\mu_{\mathcal{A}}(\ell) > \nu_{\mathcal{A}}(r)] \wedge \bigwedge_{f \in \mathcal{F}} \big([f_{\mathcal{A}}(\overline{x}) \geqslant 1] \wedge \operatorname{mon}(f_{\mathcal{A}}(\overline{x}))\big)$$

holds then $\mathcal{R}$ is terminating.

Proof. By the assumption any $f_{\mathcal{A}}$ is well-defined and monotone. Hence the result follows by Theorem 5 in combination with Lemmata 37 and 42. $\square$

5.2. Implementation

To find suitable coefficients we consider *parametric* FBIs where we let the coefficients $f_0, f_1, \ldots, f_n, \hat{f}_0, \hat{f}_1, \ldots, \hat{f}_n$ in (7) be *unknowns* over the naturals. Then the encodings from the previous section reduce the problem to finding models in existentially quantified non-linear integer arithmetic. For an efficient implementation the following heuristics (which are applied to interpretations of a function symbol but not enforced for FBIs occurring when evaluating terms) have been proved useful:

(d) depth: A function symbol f is interpreted by an FBI of depth $\max\{0, d_{\mathcal{R}}(f) - 2\}$ where $d_{\varnothing}(f) = 0$ and $d_{\mathcal{S}}(f) = 1 + \max\{d_{\mathcal{S} \setminus \mathcal{S}_f}(g) \mid \ell \to r \in \mathcal{S}_f \text{ and } g \text{ occurs in } r\}$ otherwise. Here $\mathcal{S}_f$ denotes the rules in $\mathcal{S}$ whose left-hand sides have root f. For Examples 33 and 34 the heuristic yields depth 2 for fact, depth 1 for $\cdot$, and depth 0 for the remaining function symbols.

(1) shape: Every variable may only appear once in each FBI, i.e., either in $\dot{f}(\overline{x})$ or in $\hat{f}(\overline{x})$ or in $f'(\overline{x})$. We enforce this by adding a side constraint.

(2) shape: Note that in the motivating examples every function symbol is interpreted by an FBI $f(\overline{x})$ satisfying

$$\bigwedge_{1 \leqslant i \leqslant n} (f_i = 0 \vee \hat{f}_i = 0) \tag{8}$$

Heuristic (2) shares the variables for the coefficients f_i and $\hat{f}_i$. This is achieved by using fresh boolean variables b_i and interpreting a function by (here $f_i = c_i b_i$ and $\hat{f}_i = c_i(1 - b_i)$)

$$\sum_{1 \leqslant i \leqslant n} x_i b_i c_i + c_0 b_0 + b^{f'(\overline{x})} \Bigg(\sum_{1 \leqslant i \leqslant n} x_i (1 - b_i) c_i + c_0 (1 - b_0) \Bigg)$$

Heuristic (2) does not work recursively but takes the constant part into account (in contrast to heuristic (1)).

(3) shape: At most one of $\hat{f}_i$ $(0 \leqslant i \leqslant n)$ is greater than zero.

(4) shape: If $\hat{f}(\overline{x}) = 0$ then we demand all coefficients in $\dot{f}'(\overline{x})$ and $\hat{f}'(\overline{x})$ to be zero.

(5) shape: If $\hat{f}(\overline{x}) = 0$ then we demand all coefficients in $f'(\overline{x})$ to be zero.

Note that (5) is more restrictive than (4); while the latter admits an interpretation of the form $b^{b^{2 \cdot 0}} \cdot 0$, this is not allowed when applying (5).

It is not hard to construct TRSs where FBI termination proofs require interpretations of arbitrary depth.

Example 44. Let $\mathcal{R}_n$ for $n > 0$ consist of the rules

$$\begin{array}{ccc} x + \mathsf{0} \to x & x + \mathsf{s}(y) \to \mathsf{s}(x + y) & \mathsf{exp}_0(x) \to x \\ \mathsf{exp}_{i+1}(\mathsf{0}) \to \mathsf{exp}_i(\mathsf{s}(\mathsf{0})) & \mathsf{exp}_{i+1}(\mathsf{s}(x)) \to \mathsf{exp}_i(\mathsf{exp}_1(x) + \mathsf{exp}_1(x)) & \end{array}$$

for all $0 \leqslant i < n$. Termination of $\mathcal{R}_n$ can be shown by the FBI algebra $\mathcal{A}$ with base $b = 2$ and interpretations $\mathsf{0}_\mathcal{A} = 1$, $\mathsf{s}_\mathcal{A}(x) = x + 1$, $x +_\mathcal{A} y = x + 2y$, and $\mathsf{exp}_{i,\mathcal{A}}(x) = \exp_2^i(2x + 1)$ where $\exp_2^i(x)$ denotes i-fold exponentiation with base 2, i.e., $\exp_2^0(x) = x$ and $\exp_2^{i+1}(x) = 2^{\exp_2^i(x)}$:

$$\begin{array}{ccc} x + 2 > x & x + 2y + 2 > x + 2y + 1 & 2x + 1 > x \\ \exp_2^{i+1}(3) > \exp_2^i(5) & \exp_2^{i+1}(2x + 3) > \exp_2^i(2^{2x+1} + 2 \cdot 2^{2x+1}) & \end{array}$$

The last two inequalities can be verified by simple inductive arguments. It is easy to see that any FBI algebra that orients $\mathcal{R}_n$ needs to have at least depth n.

It can be shown that already $\mathcal{R}_1$ admits multiple exponential complexity. As to be expected, actually any TRS compatible with an FBI algebra is bounded by a multiple exponential function. A more precise upper bound is given by the following lemma.

Lemma 45. *For any TRS $\mathcal{R}$ compatible with an FBI algebra $\mathcal{A}$ having base b and maximal depth $d - 1$, $\mathrm{dh}_\mathcal{R}(n) \in \exp_b^{d\,n}(\mathcal{O}(n))$.*

Proof. As $\mathrm{dh}_\mathcal{R}(t) \leqslant [t]_\mathcal{A}$, it suffices to find a $k \in \mathbb{N}$ such that any ground term t satisfies $[t]_\mathcal{A} \leqslant \exp_b^{d|t|}(k\,d\,|t|)$. Let $m - 1$ be the maximal arity in $\mathcal{F}$, c the maximum of 2 and all coefficients occurring in $f_\mathcal{A}$ for $f \in \mathcal{F}$, and $k = 1 + \log_b(c\,m)$. We apply induction on t.

Suppose t is a constant a. In order to show $a_\mathcal{A} \leqslant \exp_b^d(k\,d)$ we consider a slightly more general statement. Let α be an FBI of depth e with base b and maximum coefficient smaller than or equal to c, such that α depends on no variables. We verify $\alpha \leqslant \exp_b^{e+1}(k\,(e+1))$ by induction on e, such that in particular $a_\mathcal{A} \leqslant \exp_b^d(k\,d)$. If $e = 0$ then $\alpha \leqslant c \leqslant c\,m = \exp_b^1(\log_b(c\,m)) \leqslant \exp_b^1(k)$. Otherwise, α has depth $e + 1$ and thus α can be written as $\alpha = b^{\alpha'} c_1 + c_2$ where $c_1, c_2 \in \mathbb{N}$ and α' has depth e. By the induction hypothesis, $\alpha' \leqslant \exp_b^{e+1}(k\,(e+1))$, and hence

$$\begin{aligned} \alpha &\leqslant b^{\exp_b^{e+1}(k\,(e+1))} c + c = (b^{\exp_b^{e+1}(k\,(e+1))} + 1)c \leqslant b^{\exp_b^{e+1}(k\,(e+1))+1} b^{\log_b(c)} \\ &\leqslant b^{\exp_b^{e+1}(k\,(e+1))+k} \leqslant b^{\exp_b^{e+1}(k\,(e+2))} = \exp_b^{e+2}(k(e+2)) \end{aligned}$$

Suppose $t = g(t_1, \ldots, t_n)$ is not a constant. Let $\alpha_i = [t_i]_\mathcal{A}$. Since $|t_i| \leqslant |t| - 1$, the induction hypothesis yields $\alpha_i \leqslant \exp_b^{d\,(|t|-1)}(k\,d\,(|t| - 1))$. To verify $[t]_\mathcal{A} \leqslant \exp_b^{d\,|t|}(k\,d\,|t|)$ we consider a more general statement. Let $f(\overline{x})$ be an FBI of the shape (6), having base b, maximum coefficient at most c and depth e. We abbreviate $f(\alpha_1, \ldots, \alpha_n)$ by α and show $\alpha \leqslant \exp_b^{d\,(|t|-1)+e+1}(k\,d\,(|t| - 1) + k\,(e + 1))$ by induction on e.

If $e = 0$ then $f(\overline{x})$ is just a linear function and thus

$$\begin{aligned}\alpha &\leqslant \exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))\,c\,m \leqslant b^{\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))+\log_b(c\,m)}\\ &\leqslant b^{\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1)+k)} = \exp_b^{d\,(|t|-1)+1}(k\,d\,(|t|-1)+k)\end{aligned}$$

Now suppose $f(\overline{x})$ has depth $e+1$. Thus $f'(\overline{x})$ has depth e and, by the induction hypothesis, $f'(\alpha_1, \ldots, \alpha_n) \leqslant \exp_b^{d\,(|t|-1)+e+1}(k\,d\,(|t|-1)+k\,(e+1)) = \beta$. Therefore

$$\begin{aligned}\alpha &= \sum_{1 \leqslant i \leqslant n} \alpha_i f_i + f_0 + b^{f'(\overline{\alpha})}\Bigg(\sum_{1 \leqslant i \leqslant n} \alpha_i \hat{f}_i + \hat{f}_0\Bigg)\\ &\leqslant b^{\beta}\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))\,c\,m + \exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))\,c\,m\\ &= (b^{\beta}+1)\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))\,c\,m \leqslant b^{\beta+1}\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))\,c\,m\\ &\leqslant b^{\beta+1}b^{\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))}b^{\log_b(c\,m)} = b^{\beta+\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))+k}\\ &= b^{\exp_b^{d\,(|t|-1)+e+1}(k\,d\,(|t|-1)+k\,(e+1))+\exp_b^{d\,(|t|-1)}(k\,d\,(|t|-1))+k}\\ &\leqslant b^{\exp_b^{d\,(|t|-1)+e+1}(k\,d\,(|t|-1)+k\,(e+2))} = \exp_b^{d\,(|t|-1)+e+2}(k\,d\,(|t|-1)+k\,(e+2))\end{aligned}$$

In particular, $g_{\mathcal{A}}(\overline{\alpha}) \leqslant \exp_b^{d\,(|t|-1)+d}(k\,d\,(|t|-1)+k\,d) = \exp_b^{d\,|t|}(k\,d\,|t|)$ as the depth of $g_{\mathcal{A}}$ is smaller than d. $\square$

The next example shows that the lack of multiplication poses a weakness of FBIs.

Example 46. The following TRS (from Lescanne (1995, Fig. 2)) cannot be oriented by FBIs:

$$\begin{array}{ccc} \mathsf{0} + x \to x & \mathsf{s}(x) + y \to \mathsf{s}(x+y) & \\ \mathsf{0} \cdot x \to \mathsf{0} & \mathsf{s}(x) \cdot y \to x \cdot y + y & x \cdot (y+z) \to x \cdot y + x \cdot z \\ x \uparrow \mathsf{0} \to \mathsf{s}(\mathsf{0}) & x \uparrow \mathsf{s}(y) \to x \cdot (x \uparrow y) & \\ x \uparrow (y+z) \to (x \uparrow y) \cdot (x \uparrow z) & (x \cdot y) \uparrow z \to (x \uparrow z) \cdot (y \uparrow z) & (x \uparrow y) \uparrow z \to x \uparrow (y \cdot z) \end{array}$$

This is because for any linear function $x +_{\mathcal{A}} y$ the interpretation $x \cdot_{\mathcal{A}} y$ has to involve exponentiation (as in Examples 33 and 34). But then the rule $x \uparrow (y+z) \to (x \uparrow y) \cdot (x \uparrow z)$ is no longer orientable since the maximal power of b occurring in the (approximated) interpretation of the right-hand side exceeds the maximal power for the left-hand side. In contrast, elementary interpretations with non-fixed base succeed (cf. Lescanne (1995, Fig. 2)).

6. Experimental Results

We implemented the algebras from Sections 4 and 5 in the termination tool $\mathsf{T_TT_2}$ (Korp et al., 2009). In version 1.15, which is available from the tool's website, [9] ordinal algebras can be used by executing `./ttt2 -s HYDRA <file>` and FBI algebras by `./ttt2 -s FBI <file>`, respectively. Furthermore, the web interface has been updated accordingly.

[9] `http://cl-informatik.uibk.ac.at/software/ttt2/`

method	YES	avg. time	$\mathcal{G}$ (Definition 12)	$\mathcal{H}$ (Example 30)	$\mathcal{W}_3'$ (Example 31)
ROE algebras	329	2.1	8.2	11.4	4.0

Table 1. Experimental Results for ROE Algebras.

method	YES	avg. time	Example 33	Example 34	(Lescanne, 1995, Fig. 1)
poly	125	0.3	(0.2)	(0.4)	(0.3)
fbi	41	29.7	*1443.4*	*731.0*	*13540.5*
fbi[d]	170	4.7	16.1	10.0	27.8
fbi[d12345]	174	4.2	8.9	7.9	24.1

Table 2. Experimental Results for FBI Algebras.

For experiments [10] we considered the 1463 TRSs in the Standard TRS category of the Termination Problems Data Base (TPDB 8.0.7) [11] and the examples from the paper. The experiments have been performed using a single node of a machine equipped with 12 quad-core AMD Opteron™ processors 6174 running at a clock rate of 2.2 GHz and 330 GB of main memory. If a TRS could not be handled within 60 seconds, the execution of TTT2 was aborted.

For the evaluation in Table 1 the following setup is used. ROE algebras (according to the description in Section 4.2) with interpretation functions of initial depth two are used in combination with weakly monotone matrix interpretations of dimension two. The coefficients are represented with up to four bits. The method is applied directly (establishing simple termination, if successful) and in the DP setting in combination with dependency graphs, SCC analysis and the subterm criterion. The left part of Table 1 shows the performance of this ROE algebra-based strategy on TPDB while the relevant examples from the paper are considered in the right part of the table where the numbers indicate the execution time in seconds.

Table 2 compares the power of FBIs (of depth at most 2) with linear polynomial interpretations when used in direct termination proofs (orient all rules by a single interpretation). For numbers in parentheses TTT2 was not successful. The numbers in brackets indicate which heuristics have been used. FBIs as well as linear interpretations use two bits to encode coefficients and six bits for arithmetic evaluations.

Our experiments show the need for a heuristic concerning the depth of the FBIs. The other heuristics are much less important, i.e., they either slightly decrease the execution time or increase the number of systems shown terminating. We remark that any proper subset of the heuristics $\{1, 2, 3, 4, 5\}$ has only tiny effects on the execution speed of the examples in the right part of Table 2 while the whole set admits significant gains. The systems where FBIs succeed but linear polynomials fail often require interpretation functions of non-linear shape.

While FBIs are successful on the examples from the right part of Table 2, TTT2 cannot establish termination using ROE algebras. On the other hand, FBIs cannot cope with the examples from the right part of Table 1 due to their derivational complexity.

[10] Details available from `http://cl-informatik.uibk.ac.at/ttt2/ordinals`

[11] Available from `http://termcomp.uibk.ac.at`.

7. Conclusion

7.1. Summary

We have encoded Goodstein's sequence as a TRS and discussed automation of a termination criterion which can cope with this system. Furthermore our implementation is also successful on an encoding of the battle of Hercules and Hydra, for which a (sound) automatic termination proof has been lacking so far. While preliminary experiments on the termination problems database TPDB did not yield proofs for previously unknown problems, we regard the main attraction of our method that it allows to go beyond multiple recursive derivation length. As shown in the article, automation of lexicographic combinations of termination proofs with respect to Theorem 6 is more challenging than with respect to Theorem 5.

Needless to say, there will always be TRSs whose termination is out of reach of automatic tools. Lepper (2004) presented an infinite sequence $(\mathcal{R}_k)_{k\geqslant 1}$ of TRSs that simulate Hydra battles. Each of these TRSs is simply terminating, but the derivational complexity of $\mathcal{R}_k$ cannot be bounded by an α-recursive function such that $\alpha < \Delta_k$, where Δ_k approaches the small Veblen ordinal $\vartheta(\Omega^\omega)$ when k tends to infinity. With ROE algebras one can only prove termination of TRSs whose derivational complexity is ϵ_0-recursive.

The very first encoding of the Hydra battle in Dershowitz and Jouannaud (1990) still defeats $\mathsf{T_TT_2}$. A (difficult) termination proof of this TRS can be found in Moser (2009).

$\mathsf{T_TT_2}$ also fails on the Hydra encoding of Buchholz (2006), which is not simply terminating although it admits a comparatively concise termination argument.

Furthermore, we have also shown how elementary interpretations can be automated using similar means, a challenge formulated as Problem #28 in the RTA List of Open Problems. Somehow surprisingly, FBIs require further heuristics to admit an efficient implementation. We believe that ordinal arithmetic is *easier* for the underlying SMT solver since expressions might be consumed while this is not the case for elementary arithmetic.

7.2. Future Work

The approximation of term interpretations could partially be made more precise. As an example, we discuss scalar multiplication for ordinals. Since the approximations must be correct for all values of $\overline{x}$, the overapproximation $(f \cdot_\nu a)(\overline{x})$ is already optimal. To see this consider $(x+y)\cdot_\nu 2$ for natural values of x and y. Inspecting the proof of Lemma 24(a), instead of the current underapproximation $(f \cdot_\mu a)(\overline{x})$ we could also use (when $a > 0$)

$$(f \cdot_{\mu'} a)(\overline{x}) = \sum_{1\leqslant i\leqslant n} x_i(f_i \cdot e_i) + \omega^{f'(\overline{x})}(f_\omega \cdot a) \oplus \bigoplus_{1\leqslant i\leqslant n} x_i(\hat{f}_i \cdot a) \oplus (f_0 \cdot a)$$

where exactly one of e_i is a and all others are one. The underlying SMT solver can then choose an appropriate summand to be multiplied with a such that subsequent operations (addition, comparison, etc.) benefit. Refining the approximations for other operations (addition/comparison) is more involved and it is unclear if the additional precision prevails the increasing difficulty of the resulting SMT problems. Moreover, currently we do not know of any other TRSs with high derivational complexity that are within reach of our technique and could benefit from such improvements.

It is non-trivial to decide whether, given two ROEs $f(\overline{x})$ and $g(\overline{x})$ with given coefficients, $[\alpha](f(\overline{x})) > [\alpha](g(\overline{x}))$ holds for all assignments α. Though this problem is undecidable for polynomials, note that in the case of ROEs only linear constraints are involved. Further investigation of this issue might also lead to a better approximation of the encoding $[f(\overline{x}) > g(\overline{x})]$.

Generalizing elementary interpretations to a non-fixed base is an obvious choice for future work. However, we anticipate that suitable approximations will neither give further deep insights nor significantly improve termination proving power and hence we propose a different line of research. Since they are elementary interpretations, FBIs yield a totalizable order on ground terms. This holds despite the fact that our implementation relies on approximations when evaluating or comparing terms (see Example 36). Hence our implementation cannot be used to decide ordered rewriting, for instance to decide word problems using ground-convergent systems. However, since unfailing completion procedures never rely on the fact that $s >_{\mathcal{A}} t$ does *not* hold, FBIs can be used for ordered completion as in Winkler and Middeldorp (2010).

It is easy to enforce AC compatibility of algebras based on elementary and ordinal interpretation functions. For the case of FBI algebras, any AC symbol f must be interpreted by an FBI of the shape

$$(x_1 + x_2)f_1 + f_0 + b^{f'(x_1,x_2)}\left((x_1 + x_2)\hat{f}_1 + \hat{f}_0\right)$$

where $f'(x_1, x_2)$ is AC compatible as well. For instance, if $+$ is considered an AC symbol then AC termination of all TRSs in Examples 33, 34, and 44 can be shown with AC compatible FBI algebras (by picking $x +_{\mathcal{A}} y = 2x + 2y + c$ for a suitable constant c, and adapting the interpretations of other symbols accordingly). An ROE algebra is AC compatible if any AC symbol f is interpreted by an ROE

$$\omega^{f'(x_1,x_2)} f_\omega \oplus x_1\hat{f}_1 \oplus x_2\hat{f}_1 \oplus f_0 \tag{9}$$

where $f'(x_1, x_2)$ is again AC compatible. However, note that if a TRS $\mathcal{R}$ can be oriented with an ROE algebra where all interpretations match the shape (9) then a similar argument as used in (Winkler et al., 2012, Theorem 13) shows that $\mathcal{R}$ is also compatible with an FBI algebra with sufficiently large base b.

Since non-linear polynomials give rise to an exponential size SMT encoding, such interpretations are hardly used within termination tools. We anticipate that suitable approximations could improve the performance of these implementations.

Formalizing our approximations in a theorem prover would extend the contributions from Manolios and Vroon (2005) and enable certification of such termination proofs.

Acknowledgements

We thank Lev Beklemishev, Georg Moser, and René Thiemann for helpful discussion and the reviewers for their detailed and constructive comments.

References

Baader, F., Nipkow, T., 1998. Term Rewriting and All That. Cambridge University Press, Cambridge.

Beklemishev, L., 2006. Representing Worms as term rewriting systems. In: Proc. Mini-Workshop: Logic, Combinatorics and Independence Results. Vol. 3(4) of Oberwolfach Reports. European Mathematical Society, pp. 3093–3095.

Buchholz, W., 1987. An independence result for $(\pi^1_1 - \mathrm{CA}) + \mathrm{BI}$. Ann. Pure Appl. Logic 33, 131–155.

Buchholz, W., 2006. Another rewrite system for the standard Hydra battle. In: Proc. Mini-Workshop: Logic, Combinatorics and Independence Results. Vol. 3(4) of Oberwolfach Reports. European Mathematical Society, pp. 3099–3102.

Cichon, E., 1983. A short proof of two recently discovered independence results using recursion theoretic methods. Proc. Am. Math. Soc. 87 (4), 704–706.

Contejean, E., Marché, C., Tomás, A.-P., Urbain, X., 2005. Mechanically proving termination using polynomial interpretations. JAR 34 (4), 325–363.

Dershowitz, N., 1993. Trees, ordinals, and termination. In: TAPSOFT. Vol. 668 of LNCS. pp. 243–250.

Dershowitz, N., Jouannaud, J.-P., 1990. Rewrite Systems. Handbook of Theoretical Computer Science. Vol. B: Formal Models and Semantics. Elsevier, pp. 243–320.

Dershowitz, N., Moser, G., 2007. The Hydra battle revisited. In: Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday. Vol. 4600 of LNCS. pp. 1–27.

Endrullis, J., Waldmann, J., Zantema, H., 2008. Matrix interpretations for proving termination of term rewriting. JAR 40 (2-3), 195–220.

Fleischer, R., 2009. Die another day. Theory of Computing Systems 44 (2), 205–214.

Fuhs, C., Giesl, J., Middeldorp, A., Schneider-Kamp, P., Thiemann, R., Zankl, H., 2007. SAT solving for termination analysis with polynomial interpretations. In: SAT. Vol. 4501 of LNCS. pp. 340–354.

Gentzen, G., 1936. Die Widerspruchsfreiheit der reinen Zahlentheorie. Mathematische Annalen 122, 493–565.

Goodstein, R. L., 1944. On the restricted ordinal theorem. J. Symb. Log. 9 (2), 33–41.

Hamano, M., Okada, M., 1997. A relationship among Gentzen's proof-reduction, Kirby-Paris' Hydra game and Buchholz's Hydra game. Math. Logic Quart. 43, 103–120.

Hirokawa, N., Middeldorp, A., 2004. Polynomial interpretations with negative coefficients. In: AISC. Vol. 3249 of LNCS (LNAI). pp. 185–198.

Hofbauer, D., Lautemann, C., 1989. Termination proofs and the length of derivations (preliminary version). In: RTA. Vol. 355 of LNCS. pp. 167–177.

Hong, H., Jakuš, D., 1998. Testing positiveness of polynomials. JAR 21 (1), 23–38.

Just, W., Weese, M., 1996. Discovering Modern Set Theory. I: The Basics. American Mathematical Society.

Kirby, L., Paris, J., 1982. Accessible independence results for Peano arithmetic. Bulletin of the London Mathematical Society 14, 285–325.

Korp, M., Sternagel, C., Zankl, H., Middeldorp, A., 2009. Tyrolean Termination Tool 2. In: Proc. 20th RTA. Vol. 5595 of LNCS. pp. 295–304.

Kovács, L., Moser, G., Voronkov, A., 2011. On transfinite Knuth-Bendix orders. In: CADE. Vol. 6803 of LNCS (LNAI). pp. 384–399.

Lepper, I., 2001. Derivation lengths and order types of Knuth-Bendix orders. TCS 269 (1-2), 433–450.

Lepper, I., 2004. Simply terminating rewrite systems with long derivations. Archive for Mathematical Logic 43 (1), 1–18.

Lescanne, P., 1995. Termination of rewrite systems by elementary interpretations. Formal Aspects of Computing 7 (1), 77–90.

Lucas, S., 2009. Automatic proofs of termination with elementary interpretations. In: Proc. 9th PROLE. Vol. 258 of ENTCS. pp. 41–61.

Manolios, P., Vroon, D., 2005. Ordinal arithmetic: Algorithms and mechanization. JAR 34 (4), 387–423.

Moser, G., 2009. The Hydra battle and Cichon's principle. AAECC 20 (2), 133–158.

Moser, G., Schnabl, A., 2011. The derivational complexity induced by the dependency pair method. LMCS 7 (3).

Schnabl, A., 2012. Derivational complexity analysis revisited. Ph.D. thesis, University of Innsbruck.

TeReSe, 2003. Term Rewriting Systems. Vol. 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Touzet, H., 1998. Encoding the Hydra battle as a rewrite system. In: MFCS. Vol. 1450 of LNCS. pp. 267–276.

Turing, A., 1949. Checking a large routine. In: Report of a Conference on High Speed Automatic Calculating Machines. pp. 67–68.

Urban, C., Miné, A., 2014. An abstract domain to infer ordinal-valued ranking functions. In: Proc. 23rd European Symposium on Programming (ESOP 2014). Vol. 8410 of LNCS. pp. 412–431.

Weiermann, A., 1995. Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths. TCS 139 (1-2), 355–362.

Winkler, S., Middeldorp, A., 2010. Termination tools in ordered completion. In: IJCAR. Vol. 6173 of LNCS (LNAI). pp. 518–532.

Winkler, S., Zankl, H., Middeldorp, A., 2012. Ordinals and Knuth-Bendix orders. In: LPAR. Vol. 7180 of LNCS (ARCoSS). pp. 420–434.

Winkler, S., Zankl, H., Middeldorp, A., 2013. Beyond Peano arithmetic – automatically proving termination of the Goodstein sequence. In: RTA. Vol. 21 of LIPIcs. pp. 335–351.

Zankl, H., Middeldorp, A., 2010. Satisfiability of non-linear (ir)rational arithmetic. In: LPAR. Vol. 6355 of LNCS (LNAI). pp. 481–500.

Zankl, H., Winkler, S., Middeldorp, A., 2012. Automating ordinal interpretations. In: WST. pp. 94–98.

Zantema, H., 2001. The termination hierarchy for term rewriting. AAECC 12 (1-2), 3–19.