Beyond Peano Arithmetic – Automatically Proving Termination of the Goodstein Sequence

Sarah Winkler*, Harald Zankl, and Aart Middeldorp

Institute of Computer Science, University of Innsbruck, 6020 Innsbruck, Austria
{sarah.winkler, harald.zankl, aart.middeldorp}@uibk.ac.at

Abstract

Kirby and Paris (1982) proved in a celebrated paper that a theorem of Goodstein (1944) cannot be established in Peano (1889) arithmetic. We present an encoding of Goodstein's theorem as a termination problem of a finite rewrite system. Using a novel implementation of 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, yet another system which has been out of reach for automated tools, until now.

1998 ACM Subject Classification F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases term rewriting, termination, automation, ordinals

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

Category Regular Research Paper

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. Hofbauer and Lautemann [14] have shown that polynomial interpretations are limited to double exponential derivational complexity. They further showed that the derivational complexity of a rewrite system compatible with KBO cannot be bounded by a primitive recursive function. Later, Lepper [19] established the Ackermann function as an upper bound for KBO, whereas Weiermann [30] proved a multiple recursive upper bound for LPO. More recently, Moser and Schnabl have studied upper bounds on the complexity when using these base methods in the dependency pair framework [25, 26]. 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 [26, 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).

In this paper we encode the computation of the sequences in Goodstein's theorem as a rewrite system $\mathcal{G}$ such that termination of $\mathcal{G}$ implies Goodstein's theorem. Since the latter is not provable in Peano arithmetic (Kirby and Paris [17]), the derivational complexity of

* Supported by a DOC-fFORTE fellowship of the Austrian Academy of Sciences.
[1] http://www.termination-portal.org/

submitted to 24th International Conference on Rewriting Techniques and Applications.
Editor: F. van Raamsdonk; pp. 1–16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$\mathcal{G}$ is not multiple recursive. Despite the fact that ordinals have been used in termination arguments since many decades [11, 29], until now a successful implementation for automatic termination proofs is lacking. In this paper we discuss automation of a termination criterion based on ordinal interpretations which is capable of proving $\mathcal{G}$ terminating, thereby disproving the above conjecture. Our implementation can also cope with Touzet's encoding [28] of the battle of Hercules and Hydra (also due to [17]), yet another system whose derivational complexity is not multiple recursive. In preliminary work [31, 33] we already used ordinal domains to increase automatic termination proving power. However, in [33] the focus in 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, [31] uses ordinal domains for generalized KBO, again for string rewriting only.

This paper 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 interpretations can be automated. Rewrite systems encoding the Hydra battle are the topic of Section 5, in which also the limitations of our implementation of ordinal interpretations become apparent. We conclude in Section 6.

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 $\varnothing$, $\{\varnothing\}$, $\{\varnothing, \{\varnothing\}\}$, ... with 0, 1, 2, ..., 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, ... 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 [16] for details).

▶ **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 positive 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 positive limit ordinal.

Since $2 \cdot \omega = \omega \neq \omega \cdot 2$ multiplication is not commutative, and as $(\omega + 1) \cdot 2 = (\omega + 1) + (\omega + 1) = \omega \cdot 2 + 1$ 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 $\alpha \cdot a$.

▶ **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 positive 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, $\mathbb{N}$ 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 interpretations as considered later in this paper it is crucial that addition, natural addition, multiplication, and exponentiation are weakly monotone in both arguments.

We assume familiarity with term rewriting and termination in particular [27]. By $\overline{x}$ we abbreviate $x_1, \ldots, x_n$. We consider well-founded algebras $\mathcal{A}$ where the interpretation functions $f_{\mathcal{A}}$ take the following very general shape. *Ordinal interpretations* over variables $\overline{x}$ are the smallest set of expressions containing $\mathbb{N}$ and x_i for all $1 \leqslant i \leqslant n$, they are closed under (standard and natural) addition and multiplication, composition, and $\omega^{(\cdot)}$. An interpretation function $f_{\mathcal{A}}$ is *weakly monotone* if $a > b$ implies $f_{\mathcal{A}}(\ldots, a_{i-1}, a, a_{i+1}, \ldots) \geqslant f_{\mathcal{A}}(\ldots, a_{i-1}, b, a_{i+1}, \ldots)$. It is *simple* if $f_{\mathcal{A}}(a_1, \ldots, a_n) \geqslant a_i$ for all $1 \leqslant i \leqslant n$. An algebra is *simple/weakly monotone* if all its interpretation functions are. 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** ([28, 34]). *A TRS is terminating if it is compatible with a well-founded weakly monotone simple algebra.* ◀

In order to prove termination of TRSs with non-multiple recursive derivation length, ordinal interpretations can be lexicographically combined with linear polynomial interpretations and matrix interpretations [9].

3 The Goodstein Sequence

In this section we present a TRS for the Goodstein sequence. 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.

▶ **Definition 6.** 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 $i \geqslant 0$.

▶ **Theorem 7** (Goodstein [12]). *For all α there exists a k such that $g_\alpha(k) = 0$.* ◀

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 [17]. Cichon [5] presented a very short proof using results concerning recursion theoretic hierarchies of functions. In particular, he showed that the growth rate of G is strongly related to H_{ϵ_0}.[2]

▶ **Definition 8.** For all $n > 1$ we define a mapping $[\cdot]_n$ to represent natural numbers in 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). We denote the term $\mathsf{c}(x, \mathsf{c}(x, \cdots \mathsf{c}(x, y) \cdots))$ containing $k \geqslant 0$ occurrences of c by $\mathsf{c}^k(x, y)$. 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.

▶ **Example 9.** 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 $[7]_3 = \mathsf{c}(\mathsf{0}, \mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{c}(\mathsf{c}(\mathsf{0}, \mathsf{0}), \mathsf{0})))$.

The following definition is inspired by Touzet's encoding of the Hydra battle [28] (see Example 22).

▶ **Definition 10.** 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:

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

The idea is to encode the current base n as $[\!]^n$, followed by a term $[\alpha]_n$. The marker symbols $\circ$ and $\bullet$ are used to trigger rewrite steps while f and h compute intermediate results.

According to the following theorem, $\mathcal{G}$ simulates for any starting value the computation of the Goodstein sequence. Since the term $\bullet [\!]^n [0]_n = \bullet [\!]^n \mathsf{0}$ is clearly terminating, it follows that Theorem 7 is a consequence of the termination of $\mathcal{G}$.

---

[2] Here H is the Hardy function: $H_0(n) = n + 1$, $H_{\alpha+1}(n) = H_\alpha(n + 1)$, and $H_\lambda(n) = H_{\lambda_n}(n)$.

▶ **Theorem 11.** *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 12.**

(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 step $(\star\star)$, 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

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

Proof of Theorem 11. 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

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

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:

$$\bullet\,[]^n\,\mathsf{c}(\mathsf{c}(u,v),t) \to_{\mathcal{G}}^+ []^n\,\bullet^{2^n}\,\mathsf{c}(\mathsf{c}(u,v),t) \qquad \text{(A2)}$$

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

where $(\star)$ applies Lemma 12(b), according to which $w = [\delta + (n+1)^{\gamma} - 1]_{n+1}$. ◀

▶ **Theorem 13.** *The TRS $\mathcal{G}$ is terminating.*

Proof. We show termination of $\mathcal{G}$ according to Theorem 5. Consider the following interpretation $\mathcal{A}$ over the well-founded domain $\mathbb{O} \times \mathbb{N} \times \mathbb{N}$:

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

This interpretation is simple and weakly monotone. Because

$$
\begin{aligned}
(x, 4m+8, n) &> (x, 4m+7, n) && \text{(A1)}\\
(x, 2m+2, 2m+n+3) &> (x, 2m+2, 2m+n+2) && \text{(A2)}\\
(x, 2m+3, n) &> (x, 2m+2, n+2m+3) && \text{(A3)}\\
(x+2, 0, 0) &> (x, 2m+3, n) && \text{(B1)}\\
(\omega^{\omega^{x} \oplus y+1} \oplus z + 1, 0, 1) &> (\omega^{\omega^{x} \oplus y+1} \oplus z, 0, 1) && \text{(B2)}\\
(x+1, 0, 1) &> (x, 2m+3, n) && \text{(C1)}\\
(\omega^{\omega^{x} \oplus y+1} \oplus z, 0, 1) &> (z + \omega^{\omega^{x} \oplus y+1}, 0, 0) && \text{(C2)}\\
(y + \omega^{x+1}, 0, 1) &> (y + \omega^{x+1}, 0, 0) && \text{(D1)}\\
(y + \omega^{x+1}, 0, 0) &> (y, 2k+3, l) && \text{(D2)}\\
(\omega^{x} \oplus y, 0, 1) &> (\omega^{x} \oplus y, 0, 0) && \text{(E1)}\\
(\omega^{x} \oplus y + 1, 0, 1) &> (\omega^{x} \oplus y + 1, 0, 0) && \text{(E2)}\\
(x, m, n+m+1) &> (x, m, n) && \text{(E3)}\\
(x, 2m+3, n) &> (x, m, n) && \text{(E4)}
\end{aligned}
$$

it strictly orients all rules of $\mathcal{G}$. Hence $\mathcal{G}$ is terminating. ◀

4 Automation

In order to automate the search for suitable ordinal interpretations, we restrict to interpretations of a certain shape (see Definition 14). In Section 4.1 we show how for a given (parametric) algebra of this shape one can derive over- and underapproximations for a term's interpretation, and encode the constraints on the (coefficients of the) interpretation as a problem in non-linear integer arithmetic for which suitable SMT solvers exist (see [32]). 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.

In the sequel we consider ordinal expressions of the following shape.

▶ **Definition 14.** A *restricted ordinal expression* (*ROE*) over variables $\overline{x}$ is either 0 or[3]

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

where $f_0, f_1, \ldots, f_n, \hat{f}_1, \ldots, \hat{f}_n, f_\omega$ are (unknowns over the) naturals 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 $\mathcal{O}$ in which for every n-ary function symbol f the interpretation function $f_\mathcal{O}$ is an ROE over $\overline{x}$.

4.1 Encodings

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

$$f(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i f_i + \omega^{f'(\overline{x})} f_\omega \oplus \bigoplus_{1 \leqslant i \leqslant n} x_i \hat{f}_i \oplus f_0 \tag{4}$$

$$g(\overline{x}) = \sum_{1 \leqslant i \leqslant n} x_i g_i + \omega^{g'(\overline{x})} g_\omega \oplus \bigoplus_{1 \leqslant i \leqslant n} x_i \hat{g}_i \oplus g_0 \tag{5}$$

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 (Section 4.2.2 explains how this is ensured). We first encode some auxiliary properties of (parametric) interpretations.

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 $i \in \{0, \ldots, n, \omega\}$. 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. Consideration of a variable x_i by $f(\overline{x})$ can be recursively encoded as follows:

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

If $f(\overline{x})$ and $g(\overline{x})$ are defined as above then $\mathrm{con}(x_i, f(\overline{x})) = \top$ for all $1 \leqslant i \leqslant 3$ and $\mathrm{con}(x_j, g(\overline{x})) = \top$ for $1 \leqslant j \leqslant 2$, but $\mathrm{con}(x_3, g(\overline{x})) = \bot$.

Next, we derive formulas expressing comparisons. Consider ROEs $f(\overline{x})$ and $g(\overline{x})$ as in (4), (5). As a criterion to check whether $[\alpha](f(\overline{x})) > [\alpha](g(\overline{x}))$ for all assignments α, we use the following underapproximation, which is a tradeoff between accuracy and efficiency.

[3] 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 $1 + (0 \oplus \omega) = \omega$.

▶ **Definition 15.** Let $f(\overline{x})$ and $g(\overline{x})$ be ROEs as in (4), (5).

$$[f(\overline{x}) \geqslant g(\overline{x})] = [f(\overline{x}) \geqslant_0 g(\overline{x})] \wedge \bigwedge_{1 \leqslant i \leqslant n} [f(\overline{x}) \geqslant_i g(\overline{x})]$$

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

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

$$[f(\overline{x}) > g(\overline{x})] = [f(\overline{x}) \geqslant g(\overline{x})] \wedge [f(\overline{x}) >_0 g(\overline{x})]$$

$$[f(\overline{x}) >_0 g(\overline{x})] = ([f'(\overline{x}) \geqslant_0 g'(\overline{x})] \wedge f_\omega \geqslant g_\omega \wedge f_0 > g_0) \vee ([f'(\overline{x}) >_0 g'(\overline{x})] \wedge f_\omega > 0)$$

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 $g(\overline{x})$. Our comparisons are more involved than the absolute positiveness approach [15] because of ordinal arithmetic. We illustrate the different cases in the encoding of $\geqslant_i$ in the following example.

▶ **Example 16.** 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) satisfiability of $\omega^{x_1}\,2 \geqslant_1 x_1\,3$ is obtained while $\omega^{x_1}\,2 \geqslant_1 \omega^{x_1}\,1 + x_1\,5$ is due to (d). Case (e) obviously allows $\omega^{x_1}\,2 + x_1\,2 \geqslant_1 \omega^{x_1}\,2 + x_1\,1$ but also $\omega^{x_1} \geqslant_1 x_1\,10 + \omega^{x_1}$. Case (f) implies $x_1\,2 + \omega^{x_2} \oplus x_1\,3 \geqslant_1 x_1\,3 + \omega^{x_2} \oplus x_1\,2$. Note that if ω^{x_2} consumes the preceding $x_1\,2$ ($x_1\,3$) then $\hat{f}_1 \geqslant \hat{g}_1$ must hold. In the other case the test $f_1 + \hat{f}_1 \geqslant g_1 + \hat{g}_1$ is required. Finally, (g) ensures $x_1\,4 + \omega^{x_2} \oplus x_1\,1 \geqslant_1 x_1\,2 \oplus x_1\,3$. If ω^{x_2} consumes $x_1\,4$ then it also dominates $x_1\,2$. In the other case 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 satisfiable, despite the fact that $\omega^{x_1+1} > \omega^{x_1}\,2$. However, it is straightforward to extend Definition 15(b) accordingly.

In contrast to e.g. polynomial and matrix interpretations, ROEs are not closed under composition and (standard/natural) addition. Hence we cannot compute an expression corresponding to the interpretation of a term t with respect to an algebra $\mathcal{O}$ either. Instead, we define ROEs $\mu(t)$ and $\nu(t)$ to under- and overapproximate $t_{\mathcal{O}}$. To this end we present in Definition 17 bounds for the results of ordinal arithmetic operations (based on the algorithms given in [23]) and demonstrate them in Example 18 before Lemma 19 shows their soundness.

▶ **Definition 17.** Let $f(\overline{x})$ and $g(\overline{x})$ be ROEs as in (4), (5).

(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 $(f \oplus_\mu g)(\overline{x}) = (f \oplus_\nu g)(\overline{x}) = f(\overline{x})$ if $g(\overline{x}) = 0$. Otherwise, let s_i and t_i abbreviate $\neg\mathrm{con}(x_i, \omega^{f'(\overline{x})} f_\omega)$ and $\neg\mathrm{con}(x_i, \omega^{g'(\overline{x})} g_\omega)$ and let

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

and $(k, k_\omega) = [\omega^{f'(\overline{x})} f_\omega > \omega^{g'(\overline{x})} g_\omega] \mathrel{?} (f', f_\omega) : (g', g_\omega)$. Here $b \mathrel{?} t : e$ encodes "if b then t else e". Then

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

(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 $f + g$ by distinguishing different cases using if-then-else expressions:

$$(f +_\mu g)(\overline{x}) = [\omega^{g'(\overline{x})} g_\omega > \omega^{f'(\overline{x})} f_\omega] \mathrel{?} g(\overline{x}) : f(\overline{x})$$
$$(f +_\nu g)(\overline{x}) = ([g'(\overline{x}) > f'(\overline{x})] \wedge g_\omega > 0) \mathrel{?} \phi_1 : \big([\omega^{f'(\overline{x})} f_\omega > \omega^{g'(\overline{x})} g_\omega] \mathrel{?} \phi_2 : (f \oplus_\nu g)(\overline{x})\big)$$

where $c_0 = ([g'(\overline{x}) > 0] \wedge g_\omega > 0) \mathrel{?} g_0 : f_0 + g_0$ and

$$\phi_1 = \sum_{1 \leqslant i \leqslant n} x_i(f_i s_i t_i + \hat{f}_i t_i + g_i t_i) + \omega^{g'(\overline{x})} g_\omega \oplus \bigoplus_{1 \leqslant i \leqslant n} x_i \hat{g}_i \oplus c_0$$
$$\phi_2 = \sum_{1 \leqslant i \leqslant n} x_i f_i s_i + \omega^{f'(\overline{x})}(f_\omega + 1) \oplus \bigoplus_{1 \leqslant i \leqslant n} x_i(\hat{f}_i t_i + g_i t_i + \hat{g}_i) \oplus c_0$$

(d) Definitions (a)–(c) can be used to inductively set lower and upper bounds for the composition $f(\overline{g}) = f(g_1(\overline{x}), \ldots, g_n(\overline{x}))$. We write $\sum^\mu_{1 \leqslant i \leqslant n} h_i$ to abbreviate $h_1 +_\mu \ldots +_\mu h_n$, 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 $\mu_\mathcal{O}(t) = \nu_\mathcal{O}(t) = t$ if $t \in \mathcal{V}$, whereas $\mu_\mathcal{O}(t) = f_\mathcal{O}(\mu_\mathcal{O}(t_1), \ldots, \mu_\mathcal{O}(t_n))_\mu$ and $\nu_\mathcal{O}(t) = f_\mathcal{O}(\nu_\mathcal{O}(t_1), \ldots, \nu_\mathcal{O}(t_n))_\nu$ if $t = f(t_1, \ldots, t_n)$.

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

▶ **Example 18.**

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

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

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 $g(\overline{x}) = \omega^{x_1+x_2+1} + 1$. We have $(f +_\mu g)(\overline{x}) = g(\overline{x}) = \omega^{x_1+x_2+1} + 1$ and $(f +_\nu g)(\overline{x}) = x_3 + \omega^{x_1+x_2+1} + 1$. Note that the term $\oplus x_1$ in $f(\overline{x})$ disappears as x_1 is considered in the exponent of $g(\overline{x})$. We have

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

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 13 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 have $\mu_\mathcal{O}(r') = \nu_\mathcal{O}(r') = \omega^{\omega^{x_1} \oplus x_2} \oplus x_3$. 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 $\mu_\mathcal{O}(\ell) \geqslant_3 \nu_\mathcal{O}(r)$.

We now show that Definition 17 yields valid over- and underapproximations.

▶ **Lemma 19.** *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 17 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 α in CNF as in (1) and $a \in \mathbb{N}_{>0}$, $\alpha a = \omega^{\alpha_1} a_1 a + \omega^{\alpha_2} a_2 + \cdots + \omega^{\alpha_n} a_n$ [23]. Since for any $1 \leqslant i \leqslant n$ we have $\omega^{\alpha_1} a_1 a + \cdots + \omega^{\alpha_n} a_n \geqslant \omega^{\alpha_1} a_1 + \cdots + \omega^{\alpha_i} a_i a + \cdots + \omega^{\alpha_n} a_n$, $(f \cdot_\mu a)(\overline{x})$ constitutes a safe (though modest) lower bound for $f(\overline{x}) a$.

(b) We have

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

Note that the term $x_i f_i$ disappears in $f(\overline{x}) \oplus g(\overline{x})$ if x_i is considered in $\omega^{f'(\overline{x})}$ and $f_\omega > 0$, and the term $x_i g_i$ disappears in $f(\overline{x}) \oplus g(\overline{x})$ if x_i is considered in $\omega^{g'(\overline{x})}$ 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, 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 $(f_i s_i + g_i t_i) x_i$ 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 $g' > f'$ and $g_\omega > 0$, 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 $\omega^{g'(\overline{x})}$ and $g_\omega > 0$, the term $g_i x_i$ disappears as well if x_i is contained in $\omega^{g'(\overline{x})}$ and $g_\omega > 0$, and $f_i x_i$ disappears if x_i occurs in $\omega^{f'(\overline{x})}$ and $f_\omega > 0$, or if x_i occurs in $\omega^{g'(\overline{x})}$ and $g_\omega > 0$. Hence for any variable x_i the sum of $x_i f_i$, $x_i \hat{f}_i$, and $x_i g_i$ can be overapproximated by $x_i(f_i s_i t_i + \hat{f}_i t_i + g_i t_i)$. Therefore $[\alpha](f(\overline{x}) + g(\overline{x})) \leqslant [\alpha](f +_\nu g)(\overline{x})$.
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 $\omega^{g'(\overline{x})}$ and $g_\omega > 0$, the term $g_i x_i$ disappears as well if x_i is contained in $\omega^{g'(\overline{x})}$ and $g_\omega > 0$. 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). ◀

All (approximations of) interpretations are weakly monotone. It is easy to encode a criterion for an interpretation to be simple:

$$\mathrm{simple}(f(\overline{x})) = \bigwedge_{1 \leqslant i \leqslant n} \mathrm{con}(x_i, f(\overline{x}))$$

Thus, given a TRS $\mathcal{R}$ over a signature $\mathcal{F}$, we assign every $f \in \mathcal{F}$ an abstract ROE $f_\mathcal{O}$ of some depth d. Compatibility of $\mathcal{R}$ with a simple algebra $\mathcal{O}$ is then expressed by

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

4.2 Implementation

We implemented ordinal interpretations in the termination tool $\mathsf{T_TT_2}$ [18]. In version 1.09, which is available from the tool's website,[4] ordinal interpretations can be used by executing `./ttt2 -s HYDRA <file>`. Furthermore, the web interface has been updated accordingly. In this section we discuss crucial issues for a successful implementation. Section 4.2.1 shows how to ensure that the lexicographic combination of partial proofs preserves weak monotonicity. Section 4.2.2 deals with the problem of a compatible variable order and Section 4.2.3 is dedicated to efficiency considerations.

4.2.1 Lexicographic Combination of Interpretations

The termination proof of the TRS $\mathcal{G}$ (Theorem 7) 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 interpretation and (2) a linear (polynomial) interpretation and (3) a matrix interpretation of dimension 2. Regarding automation one can either encode

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

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 20.** 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 $[\mathsf{f}(\mathsf{b})]_{\mathcal{O}\times\mathcal{N}} = (\omega, 1) \ngeqslant_{\mathsf{lex}} (\omega, 2) = [\mathsf{f}(\mathsf{a})]_{\mathcal{O}\times\mathcal{N}}$.

However, we can recover weak monotonicity by interpreting the second component by a constant whenever the first component is only weakly—but not strictly—monotone, i.e., $f_\mathcal{O}(x, y) = (x + \omega, c)$ for some $c \in \mathbb{N}$. To achieve this goal in the implementation we consider relative rewriting and add a rule $f' \to f(x_1, \dots, x_n)$ in the relative part whenever $f_\mathcal{O}$ is not strictly monotone. Here f' is a fresh constant. In the presence of a rule $f' \to f(x_1, \dots, x_n)$, compatible interpretations satisfy $f_\mathcal{A}(x_1, \dots, x_n) = c$ for some c in the domain of $\mathcal{A}$. The idea is demonstrated by the following example.

▶ **Example 21** (Example 20 revisited)**.** Consider the TRS $\mathcal{R}$ from Example 20. 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.

However, adding rules $f' \to f(x_1, \dots, x_n)$ is likely to disable the orientation of rules whose left-hand sides are rooted by f (to satisfy $[\alpha]_\mathcal{A}(f') \geqslant [\alpha]_\mathcal{A}(f(x_1, \dots, x_n))$ the interpretation of f may not depend on its arguments) 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 $f' \to f(x_1, \dots, x_n)$ 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 $f(x_1, \dots, x_n) \to x_i$ (for $1 \leqslant i \leqslant n$) into the relative component of the TRS. This then ensures $[\alpha]_\mathcal{A}(f(x_1, \dots, x_n)) \geqslant [\alpha]_\mathcal{A}(x_i)$ for each $1 \leqslant i \leqslant n$.

Hence for a TRS $\mathcal{R}$ over a signature $\mathcal{F}$ the procedure amounts to the following three steps:

1. $\mathcal{S} := \{f(x_1, \dots, x_n) \to x_i \mid 1 \leqslant i \leqslant n, f \in \mathcal{F}\}$.
2. Find an algebra $\mathcal{A}$ satisfying $\mathcal{R} \cup \mathcal{S} \subseteq \geqslant_\mathcal{A}$ and $\mathcal{R} \cap >_\mathcal{A} \neq \varnothing$.
   $\mathsf{Nmon}_\mathcal{A}(\mathcal{R}) := \{f' \to f(x_1, \dots, x_n) \mid 1 \leqslant i \leqslant n, f \in \mathcal{F}, f_\mathcal{A} \text{ is not strictly monotone}\}$.
   $\mathcal{R} := \mathcal{R} \setminus >_\mathcal{A}$ and $\mathcal{S} := (\mathcal{S} \setminus >_\mathcal{A}) \cup \mathsf{Nmon}_\mathcal{A}(\mathcal{R})$.
3. If $\mathcal{R} = \varnothing$ then output *terminating* else go to step (2).

Instead of proving termination of $\mathcal{R}$ we try to establish termination of $\mathcal{R}$ relative to $\mathcal{S}$ (cf. step (1)). This pre-processing step ensures that the algebras in step (2) are simple. Step (2) employs SMT to find appropriate ROEs and matrix interpretations (of different dimensions), respectively. Note that this step may fail and cause the procedure to abort. Adding $\mathsf{Nmon}_\mathcal{A}(\mathcal{R})$ in the relative part ensures that the lexicographic combination of the used algebras is weakly monotone.

4.2.2 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 the encoding $[x+y+1 > y+x]$ is satisfiable. The same effect also happens in arithmetic operations, e.g. the overapproximation of $+$ in Lemma 19(d). Taking $\mathsf{f}_{\mathcal{O}}(x,y) = \mathsf{g}_{\mathcal{O}}(x,y) = x+y$, and $\alpha(x) = 1$, $\alpha(y) = \omega$, 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 add a constraint expressing that two ordinal expressions have *compatible* variable orders (in the standard addition part). 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', j' with $1 \leqslant i' < j' \leqslant n$ such that $x_i = y'_j$ and $x_j = y'_i$. 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 $e_1 = x_1 \cdot 1 + x_2 \cdot 1$, $e_2 = x_2 \cdot 1 + x_1 \cdot 1$, and $e_3 = x_2 \cdot 1 + x_1 \cdot 0$. Then e_1 and e_2 do not have compatible variable orders while e_1 and e_3 do have.

4.2.3 Efficiency

While the implementation fixes some initial depth d for the interpretation of function symbols, this depth increases when evaluating terms (when composing $f(g_1(\overline{x}),\dots,g_n(\overline{x}))$). It turned out that 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.

For the automatic termination proof of the TRS $\mathcal{G}$ in $\mathsf{T_TT_2}$ we (lexicographically) combine ordinal interpretations with matrix interpretations [9]. Then, $\mathsf{T_TT_2}$ manages $\mathcal{G}$ within six 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 86.000 variables and 217.000 clauses.

5 Hydra Battles

In their influential paper [17], Kirby and Paris 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 [10] 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 [4, 6–8, 20, 28]. Not all of these TRSs faithfully model the battle, and termination of some of them are not independent of Peano arithmetic.

▶ **Example 22.** Touzet [28] 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 a so-called *standard strategy*:

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

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 $f(x,y) = y + \omega^{x+1}$ [28]:

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

Compared to $\mathcal{G}$, $\mathsf{T_TT_2}$ requires more resources (initial depth 2, intermediate depth 3, 12 seconds, 117.000 variables, 300.000 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 $H_{\omega^{\omega^{\omega}}}$.

In [2], Beklemishev presents two infinite and one finite TRS $\mathcal{W}$ describing the Worm battle (corresponding to a one-dimensional version of Buchholz' Hydra game [3], first introduced by Hamano and Okada [13]). The finite system $\mathcal{W}$ consists of the following rules:

$$
\begin{aligned}
&(x \cdot y) \cdot z \to x \cdot (y \cdot z) & &\mathsf{a}(\mathsf{f}(x)) \to \mathsf{f}(\mathsf{a}(x)) & &\mathsf{a}(x \cdot y) \to \mathsf{a}(x) \cdot y\\
&\mathsf{a}(\mathsf{b}_1(x)) \to \mathsf{b}_1(\mathsf{a}(x)) & &\mathsf{f}(\mathsf{b}(x)) \to \mathsf{b}(\mathsf{f}(x)) & &\mathsf{b}(x) \cdot y \to \mathsf{b}(x \cdot y)\\
&\mathsf{a}(\mathsf{f}(\mathsf{0} \cdot x)) \to \mathsf{b}_1((\mathsf{f}(\mathsf{0} \cdot x)) \cdot (\mathsf{0} \cdot \mathsf{f}(x))) & &\mathsf{a}(\mathsf{f}(\mathsf{0})) \to \mathsf{b}_1(\mathsf{f}(\mathsf{0}) \cdot \mathsf{0}) & &\mathsf{b}_1(\mathsf{b}(x)) \to \mathsf{b}(\mathsf{b}(x))\\
&\mathsf{f}(\mathsf{0} \cdot x) \to \mathsf{b}(\mathsf{0} \cdot \mathsf{f}(x)) & &\mathsf{f}(\mathsf{0}) \to \mathsf{b}(\mathsf{0}) & &\mathsf{c}(\mathsf{b}(x)) \to \mathsf{c}(\mathsf{a}(x))\\
&\mathsf{a}(\mathsf{b}(x)) \to \mathsf{b}(\mathsf{a}(x))
\end{aligned}
$$

Termination of $\mathcal{W}$ is proved in [2] by relating it to another, infinite TRS. We have not been able to prove termination using ordinal interpretations, a goal we mention as a future aim.

Needless to say, there will always be TRSs whose termination is out of reach of automatic tools. With our implementation of ordinal interpretations, one cannot prove termination of TRSs whose derivational complexity goes beyond ϵ_0. Lepper [20] presented an infinite sequence $(\mathcal{R}_k)_{k \geqslant 1}$ of TRSs that simulate Hydra battles. The derivational complexity Δ_k of $\mathcal{R}_k$ approaches the small Veblen ordinal $\vartheta(\Omega^\omega)$ when k tends to infinity.

Furthermore, as our implementation is based on Theorem 5 it cannot cope with TRSs that are non-simply terminating. Hence the very first encoding of the Hydra battle in [7] defeats $\mathsf{T_TT_2}$. A (difficult) termination proof of this TRS can be found in Moser [24]. While dependency pairs [1] go beyond simple termination they do not solve the intrinsic problems of this work (exceeding multiple recursion, establishing weak monotonicity) and have hence been discarded for ease of presentation.

6 Conclusion

6.1 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 (see footnote 1 on page 1) 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 paper, automation of lexicographic combinations of termination proofs with respect to Theorem 5 is more challenging than in the standard setting where strictly monotone algebras are employed.

6.2 Future Work

Concerning future work we want to improve the approximations of our term interpretation encodings. Here we discuss scalar multiplication. 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 19(a), instead of the current underapproximation $(f \cdot_\mu a)(\overline{x})$ we could also use (if $a > 0$):

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

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 is in a suitable ratio with the increasing difficulty of the resulting SMT problems.

Furthermore we stress that (efficient) approximations (similar to the ones presented) will also be necessary for a successful implementation of e.g. elementary functions as proposed by Lescanne [21]. Despite the recent efforts of Lucas [22], to our knowledge an implementation of such interpretations is still lacking. We anticipate that automation of elementary functions might give new automatic termination proofs for problems from TPDB.

Acknowledgments: We thank Georg Moser for his comments on Hydra battles and ordinals, René Thiemann for communicating Example 20, and the anonymous reviewers for many helpful remarks.

References

1 T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *TCS*, 236(1-2):133–178, 2000.
2 L. Beklemishev. Representing Worms as term rewriting systems. In *Proc. Mini-Workshop: Logic, Combinatorics and Independence Results*, volume 3(4) of *Oberwolfach Reports*, pages 3093–3095. European Mathematical Society, 2006.
3 W. Buchholz. An independence result for $(\pi^1_1 - \mathrm{CA}) + \mathrm{BI}$. *Ann. Pure Appl. Logic*, 33:131–155, 1987.
4 W. Buchholz. Another rewrite system for the standard Hydra battle. In *Proc. Mini-Workshop: Logic, Combinatorics and Independence Results*, volume 3(4) of *Oberwolfach Reports*, pages 3099–3102. European Mathematical Society, 2006.
5 E.A. Cichon. A short proof of two recently discovered independence results using recursion theoretic methods. *Proc. Am. Math. Soc.*, 87(4):704–706, 1983.
6 N. Dershowitz. Trees, ordinals, and termination. In *TAPSOFT*, volume 668 of *LNCS*, pages 243–250, 1993.
7 N. Dershowitz and J.-P. Jouannaud. *Rewrite Systems. Handbook of Theoretical Computer Science*, volume B: Formal Models and Sematics, pages 243–320. Elsevier, 1990.
8 N. Dershowitz and G. Moser. The Hydra battle revisited. In *Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday*, volume 4600 of *LNCS*, pages 1–27, 2007.

9 J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. *JAR*, 40(2-3):195–220, 2008.
10 R. Fleischer. Die another day. *Theory of Computing Systems*, 44(2):205–214, 2009.
11 G. Gentzen. Die widerspruchsfreiheit der reinen zahlentheorie. *Mathematische Annalen*, 122:493–565, 1936.
12 R. L. Goodstein. On the restricted ordinal theorem. *J. Symb. Log.*, 9(2):33–41, 1944.
13 M. Hamano and M. Okada. A relationship among Gentzen's proof-reduction, Kirby-Paris' Hydra game and Buchholz's Hydra game. *Math. Logic Quart.*, 43:103–120, 1997.
14 D. Hofbauer and C. Lautemann. Termination proofs and the length of derivations (preliminary version). In *RTA*, volume 355 of *LNCS*, pages 167–177, 1989.
15 H. Hong and D. Jakuš. Testing positiveness of polynomials. *JAR*, 21(1):23–38, 1998.
16 W. Just and M. Weese. *Discovering Modern Set Theory. I The Basics*. American Mathematical Society, 1996.
17 L. Kirby and J. Paris. Accessible independency results for Peano arithmetic. *Bulletin of the London Mathematical Society*, 14:285–325, 1982.
18 M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean Termination Tool 2. In *RTA*, volume 5595 of *LNCS*, pages 295–304, 2009.
19 I. Lepper. Derivation lengths and order types of Knuth-Bendix orders. *TCS*, 269(1-2):433–450, 2001.
20 I. Lepper. Simply terminating rewrite systems with long derivations. *Archive for Mathematical Logic*, 43(1):1–18, 2004.
21 P. Lescanne. Termination of rewrite systems by elementary interpretations. *Formal Asp. Comp.*, 7(1):77–90, 1995.
22 S. Lucas. Automatic proofs of termination with elementary interpretations. *ENTCS*, 258(1):41–61, 2009.
23 P. Manolios and D. Vroon. Ordinal arithmetic: Algorithms and mechanization. *JAR*, 34(4):387–423, 2005.
24 G. Moser. The Hydra battle and Cichon's principle. *AAECC*, 20(2):133–158, 2009.
25 G. Moser and A. Schnabl. The derivational complexity induced by the dependency pair method. *LMCS*, 7(3), 2011.
26 A. Schnabl. *Derivational Complexity Analysis Revisited*. PhD thesis, University of Innsbruck, 2012.
27 TeReSe. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
28 H. Touzet. Encoding the Hydra battle as a rewrite system. In *MFCS*, volume 1450 of *LNCS*, pages 267–276, 1998.
29 A. Turing. Checking a large routine. In *Report of a Conference on High Speed Automatic Calculating Machines*, pages 67–68, 1949.
30 A. Weiermann. Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths. *TCS*, 139(1-2):355–362, 1995.
31 S. Winkler, H. Zankl, and A. Middeldorp. Ordinals and Knuth-Bendix orders. In *LPAR*, volume 7180 of *LNCS (ARCoSS)*, pages 420–434, 2012.
32 H. Zankl and A. Middeldorp. Satisfiability of non-linear (ir)rational arithmetic. In *LPAR*, volume 6355 of *LNCS (LNAI)*, pages 481–500, 2010.
33 H. Zankl, S. Winkler, and A. Middeldorp. Automating ordinal interpretations. In *WST*, pages 94–98, 2012.
34 H. Zantema. The termination hierarchy for term rewriting. *AAECC*, 12(1-2):3–19, 2001.