A Path Order for Rewrite Systems that Compute Exponential Time Functions*

Martin Avanzini[1], Naohi Eguchi[2], and Georg Moser[1]

1 **Institute of Computer Science, University of Innsbruck, Austria**
{martin.avanzini,georg.moser}@uibk.ac.at

2 **School of Information Science, Japan Advanced Institute of Science and Technology, Japan**
n-eguchi@jaist.ac.jp

Abstract

In this paper we present a new path order for rewrite systems, the *exponential path order* $\mathsf{EPO}^\star$. Suppose a term rewrite system is compatible with $\mathsf{EPO}^\star$, then the runtime complexity of this rewrite system is bounded from above by an exponential function. Furthermore, the class of function computed by a rewrite system compatible with $\mathsf{EPO}^\star$ equals the class of functions computable in exponential time on a Turing machine.

1998 ACM Subject Classification F.2.2, F.4.1, F.4.2, D.2.4, D.2.8

Keywords and phrases Runtime Complexity, Exponential Time Functions, Implicit Computational Complexity

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

Category Regular Research Paper

1 Introduction

In this paper we are concerned with the complexity analysis of term rewrite systems (TRSs) and the ramifications of such an analysis in implicit computational complexity (ICC for short).

Several notions to assess the complexity of a terminating term rewrite system (TRS) have been proposed in the literature, compare [12, 20, 13, 18]. The conceptually simplest one was suggested by Hofbauer and Lautemann in [20]: the complexity of a given TRS is measured as the maximal length of derivation sequences. More precisely, the *derivational complexity function* with respect to a terminating TRS $\mathcal{R}$ relates the maximal derivation height to the size of the initial term. A more fine-grained approach is introduced in [12] (compare also [18]), where the derivational complexity function is refined so that in principle only argument normalised (aka basic) terms are considered. This notion, in the following referred to as the *runtime complexity* of TRSs, aims at capturing the complexity of the *functions computed* by the analysed TRS (see [13]).

* The first and third author are partly supported by FWF (Austrian Science Fund) projects P20133-N15. The second author is supported in part by the JSPS Institutional Program for Young Researcher Overseas Visits.

submitted to 22nd International Conference on Rewriting Techniques and Applications.
Editor: M. Schmidt-Schauß Editor; pp. 1–16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In recent years the field of complexity analysis of rewrite systems matured and some advances towards an automated complexity analysis of TRSs evolved (see [24] for an overview). The current focus of modern complexity analysis of rewrite systems is on techniques that yield *polynomial* derivational or runtime complexity. In this paper we study a complementary view. We establish following results:

- We present a new path order for rewrite systems, the *exponential path order* EPO*. Suppose a TRS $\mathcal{R}$ is compatible with EPO*. Then the runtime complexity of $\mathcal{R}$ is at most exponential.
- EPO* is sound, that is, any function computed by a TRS compatible with EPO* is computable on a Turing machine in exponential time.
- EPO* is complete, that is, any function computable in exponential time can be computed by a TRS that is compatible with EPO*.

Note that the first and second result relate two different notions of the complexity of a TRS $\mathcal{R}$: the *runtime complexity* with respect to $\mathcal{R}$ and the complexity of the function *computed* with $\mathcal{R}$. Furthermore, we have implemented the order EPO* so that our research yields a fully automatic complexity tool for exponential time functions. Our research is motivated by earlier successful order-theoretic characterisations of complexity classes. We mention the *light multiset path order* introduced by Marion [23]. Roughly speaking the light multiset path order is a tamed version of the multiset path order, characterising the functions computable in polytime (compare also [4]). In a similar spirit the here presented path order EPO* characterises the functions computable in exponential time.

The definition of EPO* makes use of *tiering* [8] and is strongly influenced by a recursion theoretic characterisation $\mathcal{N}$ of the class of functions computable in exponential time by Arai and the second author (see [1]), and a very recent term-rewriting characterisation of $\mathcal{N}$ by the second author (see [15]). We motivate our study through the following example.

▶ **Example 1.1.** Consider the following TRS $\mathcal{R}_{\mathsf{fib}}$ which is easily seen to represent the computation of the n^{th} Fibonacci number.

$$\mathsf{fib}(x) \to \mathsf{dfib}(x, \mathsf{0}) \qquad \mathsf{dfib}(\mathsf{0}, y) \to \mathsf{s}(y)$$
$$\mathsf{dfib}(\mathsf{s}(\mathsf{0}), y) \to \mathsf{s}(y) \qquad \mathsf{dfib}(\mathsf{s}(\mathsf{s}(x)), y) \to \mathsf{dfib}(\mathsf{s}(x), \mathsf{dfib}(x, y))$$

Then all rules in the TRS $\mathcal{R}_{\mathsf{fib}}$ can be oriented with EPO*, which allows us to (automatically) deduce that the runtime complexity of this system is exponential. Using the machinery of [5], exploiting graph rewriting, we can even show that any function computed by a TRS compatible with EPO* is computable in exponential time on a Turing machine. Conversely we show that any function f that can be computed in exponential time on a Turing machine can be computed by a TRS $\mathcal{R}(f)$ such that $\mathcal{R}(f)$ is compatible with EPO*. In total, we obtain an alternative, syntactic characterisation of the exponential time functions.

Related Work. With respect to rewriting we mention [16], where it is shown that *matrix interpretations* yield exponential derivational complexity, hence at most exponential runtime complexity. Our work is also directly related to work in ICC (see [7] for an overview). We want to mention [10, 22], were alternative characterisations of the class of functions computable in exponential time are given. For less directly related work we cite [9], where a complete characterisation of (imperative) programs that admit linear and polynomial runtime complexity is established. As these characterisations are decidable, we obtain a decision procedure for programs that admit a runtime complexity that is at most exponential.

The remainder of the paper is organised as follows. In Section 2 we recall definitions. The order EPO* is presented in Section 3. In Section 4 we introduce an intermediate order EPO critical for establishing our soundness result, and in Section 5 we prove that EPO* induces exponentially bounded runtime complexity. In Section 6 we present the aforementioned soundness and completeness result. Finally, we conclude in Section 7. Due to space limitations we omit some proofs of auxiliary lemmas. Missing proofs are available in a separate technical report [3].

2 Preliminaries

We assume familiarity with the basics of term rewriting, see [6, 26] and briefly review definitions and notations used. Let $\mathcal{V}$ denote a countably infinite set of variables and let $\mathcal{F}$ be a finite signature. The set of terms over $\mathcal{F}$ and $\mathcal{V}$ is written as $\mathcal{T}(\mathcal{F},\mathcal{V})$. We denote by $\vec{s},\vec{t},\dots$ sequences of terms, and for a set of terms T we write $\vec{t} \subseteq T$ to indicate that for each t_i appearing in $\vec{t}$, $t_i \in T$. We suppose that the signature $\mathcal{F}$ is partitioned into *defined symbols* $\mathcal{D}$ and *constructors* $\mathcal{C}$. The set of *basic* terms $\mathcal{B} \subseteq \mathcal{T}(\mathcal{F},\mathcal{V})$ is defined as $\mathcal{B} := \{f(t_1,\dots,t_n) \mid f \in \mathcal{D} \text{ and } t_i \in \mathcal{T}(\mathcal{C},\mathcal{V}) \text{ for } i \in \{1,\dots,n\}\}$.

We write $\trianglelefteq$ and $\trianglerighteq$ to denote the *subterm* and *superterm* relation, the strict part of $\trianglelefteq$ (respectively $\trianglerighteq$) is denoted by $\triangleleft$ (respectively $\triangleright$). We denote by $|t|$ and $\mathsf{dp}(t)$ the *size* and *depth* of the term t. The *root symbol* (denoted as $\mathsf{rt}(t)$) of a term t is either t itself, if $t \in \mathcal{V}$, or the symbol f, if $t = f(t_1,\dots,t_n)$.

A *preorder* is a reflexive and transitive binary relation. If $\gtrsim$ is a preorder, we write $\sim := \gtrsim \cap \lesssim$ and $> := \gtrsim \setminus \sim$ to denote the *equivalence* and *strict part* of $\gtrsim$ respectively. A *quasi-precedence* (or simply *precedence*) is a preorder $\gtrsim = > \uplus \sim$ on the signature $\mathcal{F}$ so that the strict part $>$ is well-founded. We lift the equivalence $\sim$ induced by the precedence $\gtrsim$ to terms in the obvious way: $s \sim t$ if and only if (i) $s = t$, or (ii) $s = f(s_1,\dots,s_n)$, $t = g(t_1,\dots,t_n)$, $f \sim g$ and $s_i \sim t_i$ for all $i \in \{1,\dots,n\}$. The precedence $\gtrsim$ induces a *rank* $\mathsf{rk}(f)$ for any $f \in \mathcal{F}$ as follows: $\mathsf{rk}(f) := \max\{1 + \mathsf{rk}(g) \mid g \in \mathcal{F} \text{ and } f > g\}$.

Let $\mathcal{R}$ be a TRS over $\mathcal{F}$. We write $\to_{\mathcal{R}}$ for the induced rewrite relation. A term $s \in \mathcal{T}(\mathcal{F},\mathcal{V})$ is called a *normal form* if there is no $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$ such that $s \to_{\mathcal{R}} t$. We use $\mathrm{NF}(\mathcal{R})$ to denote the set of normal-forms of $\mathcal{R}$. With $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ we denote the *innermost rewrite relation*. We write $s \to^{!}_{\mathcal{R}} t$ (respectively $s \xrightarrow{\mathrm{i}}^{!}_{\mathcal{R}} t$) if $s \to^{*}_{\mathcal{R}} t$ (respectively $s \xrightarrow{\mathrm{i}}^{*}_{\mathcal{R}} t$) and $t \in \mathrm{NF}(\mathcal{R})$. A TRS is a *constructor* TRS if left-hand sides are basic terms and it is *completely defined* if each defined symbol is completely defined. Here a symbol is completely defined if it does not occur in any normal form. A TRS $\mathcal{R}$ is called *terminating* if $\to_{\mathcal{R}}$ is well-founded, $\mathcal{R}$ is *confluent* if for all terms s, t_1, t_2 with $s \to^{*}_{\mathcal{R}} t_1$ and $s \to^{*}_{\mathcal{R}} t_2$, there exists u such that $t_1 \to^{*}_{\mathcal{R}} u$ and $t_2 \to^{*}_{\mathcal{R}} u$.

Let $\to$ be a finitely branching, well-founded binary relation on terms. The *derivation height* of a term t with respect to $\to$ is given by $\mathrm{dh}(t,\to) := \max\{n \mid \exists u.\ t \to^n u\}$. Here $\to^n$ denotes the n-fold application of $\to$. The (innermost) *runtime complexity* of a terminating TRS $\mathcal{R}$ is defined as $\mathrm{rc}^{(\mathrm{i})}_{\mathcal{R}}(n) := \max\{\mathrm{dh}(t,\to) \mid t \in \mathcal{B} \text{ and } |t| \leqslant n\}$, where $\to$ denotes $\to_{\mathcal{R}}$ or $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ respectively. We say the (innermost) runtime complexity is exponential to assert the existence of an exponential function that binds $\mathrm{rc}^{(\mathrm{i})}_{\mathcal{R}}$ from above.

Furthermore, we assume (at least nodding) acquaintance with complexity theory, compare [21]. We write $\mathbb{N}$ for the set of *natural numbers*. Let M be a *Turing machine* (TM for short) with alphabet Σ, and let $w \in \Sigma^*$. We say that M computes $v \in \Sigma^*$ on input w, if M accepts w and v is written on a dedicated output tape. Note that when M is nondeterministic, then v computed on input w may not be unique. We say that M computes a binary relation $R \subseteq \Sigma^* \times \Sigma^*$ if for all $w, v \in \Sigma^*$ with $w\ R\ v$, M computes v on input w. Note that if M is deterministic then R induces a partial function $f_R : \Sigma^* \to \Sigma^*$. In this case we say that M computes the function f_R. Let $S \colon \mathbb{N} \to \mathbb{N}$ denote a bounding function. We denote by $\mathsf{FTIME}(S(n))$ the class of functions computable by some TM M in time $S(n)$. Then $\mathsf{FEXP} := \bigcup_{k \in \mathbb{N}} \mathsf{FTIME}(2^{\mathsf{O}(n^k)})$ denotes the class of *exponential-time computable functions*.

3 Exponential Path Order EPO*

In this section we present the *exponential path order* (EPO* for short). Throughout the following, we fix $\gtrsim$ to denote an *admissible* quasi-precedence on $\mathcal{F}$. Here a precedence is called admissible if constructors are minimal, i.e., for all defined symbols f we have $f > c$ for all constructors c.

In addition to the precedence $\gtrsim$, an instance of EPO* is induced by a *safe mapping* $\mathsf{safe} \colon \mathcal{F} \to 2^{\mathbb{N}}$. This mapping associates with every n-ary function symbol f the set of *safe argument positions* $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$. Argument positions included in $\mathsf{safe}(f)$ are called *safe*, those not included are called *normal* and collected in $\mathsf{nrm}(f)$. For n-ary constructors c we require that all argument positions are safe, i.e., $\mathsf{safe}(c) = \{1, \ldots, n\}$. To simplify the presentation, we write $f(t_{i_1}, \ldots, t_{i_k}; t_{j_1}, \ldots, t_{j_l})$ for the term $f(t_1, \ldots, t_n)$ with $\mathsf{nrm}(f) = \{i_1, \ldots, i_k\}$ and $\mathsf{safe}(f) = \{j_1, \ldots, j_l\}$.

We restrict term equivalence $\sim$ in the definition of $\overset{s}{\sim}$ below so that the separation of arguments through safe is taken into account: We define $s \overset{s}{\sim} t$ if either (i) $s = t$, or (ii) $s = f(s_1, \ldots, s_l\,;\, s_{l+1}, \ldots, s_{l+m})$, $t = g(t_1, \ldots, t_l\,;\, t_{l+1}, \ldots, t_{l+m})$ where $f \sim g$ and $s_i \overset{s}{\sim} t_i$ for all $i \in \{1, \ldots, l+m\}$. Note that in particular $\mathsf{safe}(f) = \mathsf{safe}(g)$. The definition of an instance $>_{\mathsf{epo}\star}$ of EPO* is split into the following two definitions.

▶ **Definition 3.1.** Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that $s = f(s_1, \ldots, s_l\,;\, s_{l+1}, \ldots, s_{l+m})$. Then $s \sqsupset_{\mathsf{epo}\star} t$ if $s_i \sqsupseteq_{\mathsf{epo}\star} t$ for some $i \in \{1, \ldots, l+m\}$. Further, if $f \in \mathcal{D}$, then $i \in \mathsf{nrm}(f)$. Here we set $\sqsupseteq_{\mathsf{epo}\star} := \sqsupset_{\mathsf{epo}\star} \cup \overset{s}{\sim}$.

▶ **Definition 3.2.** Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ such that $s = f(s_1, \ldots, s_l\,;\, s_{l+1}, \ldots, s_{l+m})$. Then $s >_{\mathsf{epo}\star} t$ with respect to the admissible precedence $\gtrsim$ and safe mapping safe if either

1) $s_i \geqslant_{\mathsf{epo}\star} t$ for some $i \in \{1, \ldots, l+m\}$, or
2) $t = g(t_1, \ldots, t_k\,;\, t_{k+1}, \ldots, t_{k+n})$, $f > g$ and
   i) $s \sqsupset_{\mathsf{epo}\star} t_1, \ldots, s \sqsupset_{\mathsf{epo}\star} t_k$, and
   ii) $s >_{\mathsf{epo}\star} t_{k+1}, \ldots, s >_{\mathsf{epo}\star} t_{k+n}$, or
3) $t = g(t_1, \ldots, t_k\,;\, t_{k+1}, \ldots, t_{k+n})$, $f \sim g$ and for some $i \in \{1, \ldots, \min(l,k)\}$
   i) $s_1 \overset{s}{\sim} t_1, \ldots, s_{i-1} \overset{s}{\sim} t_{i-1}$, $s_i \sqsupset_{\mathsf{epo}\star} t_i$, $s \sqsupset_{\mathsf{epo}\star} t_{i+1}, \ldots, s \sqsupset_{\mathsf{epo}\star} t_k$, and
   ii) $s >_{\mathsf{epo}\star} t_{k+1}, \ldots, s >_{\mathsf{epo}\star} t_{k+n}$.

Here we set $\geqslant_{\mathsf{epo}\star} := >_{\mathsf{epo}\star} \cup \overset{s}{\sim}$.

We write $\rhd/_{\sim}$ for the *superterm relation modulo term equivalence* $\sim$, defined as follows: $f(s_1, \ldots, s_n) \rhd/_{\sim} t$ if $s_i \rhd/_{\sim} t$ or $s_i \sim t$ for some $i \in \{1, \ldots, m\}$. Further, we set $\unrhd/_{\sim} := \rhd/_{\sim} \cup \sim$. As immediate consequence of the definitions we obtain the following lemma.

▶ **Lemma 3.3.** *The inclusions $\sqsupset_{\mathsf{epo}\star} \subseteq \rhd/_{\sim} \subseteq >_{\mathsf{epo}\star}$ hold and further, if $s \in \mathcal{T}(\mathcal{C}, \mathcal{V})$ and $s >_{\mathsf{epo}\star} t$ then $t \in \mathcal{T}(\mathcal{C}, \mathcal{V})$.*

Note that the last property holds due to the restrictions imposed on precedence and safe mapping. The central theorem of this section states that EPO* induces exponential innermost runtime complexity.

▶ **Theorem 3.4.** *Suppose $\mathcal{R}$ is a constructor TRS compatible with $>_{\mathsf{epo}\star}$, i.e., $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$. Then the innermost runtime complexity $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}(n)$ is bounded by an exponential $2^{\mathsf{O}(n^k)}$ for some fixed $k \in \mathbb{N}$.*

The proof of this theorem needs further preparation: We introduce in Section 4 an auxiliary order EPO, akin to the order presented in [1]. Although this auxiliary order is admittedly technical, it is easier to reason about its induced complexity. In Section 5 we then use this order to measure the derivation height of terms with respect to $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$, proving Theorem 3.4.

▶ **Example 3.5.** [Example 1.1 continued]. Let safe be the safe mapping such that $\mathsf{safe}(\mathsf{fib}) = \varnothing$ and $\mathsf{safe}(\mathsf{dfib}) = \{2\}$. Further, let $\gtrsim$ be the admissible precedence with $\mathsf{fib} > \mathsf{dfib} > \mathsf{s} \sim \mathsf{0}$. It is easy to verify that $\mathcal{R}_{\mathsf{fib}} \subseteq >_{\mathsf{epo}\star}$ for the induced order $>_{\mathsf{epo}\star}$. By Theorem 3.4 we conclude that the innermost runtime complexity of $\mathcal{R}_{\mathsf{fib}}$ is exponentially bounded.

We emphasise that Theorem 3.4 *does not* hold for full rewriting.

▶ **Example 3.6.** Consider the TRS $\mathcal{R}_{\mathsf{d}}$ consisting of the rules

$$\mathsf{d}(;x) \to \mathsf{c}(;x,x) \qquad \mathsf{f}(\mathsf{0};y) \to y \qquad \mathsf{f}(\mathsf{s}(;x);y) \to \mathsf{f}(x;\mathsf{d}(;\mathsf{f}(x;y))) \ .$$

Then $\mathcal{R}_{\mathsf{d}} \subseteq >_{\mathsf{epo}\star}$ for the precedence $\mathsf{f} > \mathsf{d} > \mathsf{c}$ and safe mapping as indicated in the definition of $\mathcal{R}_{\mathsf{d}}$. Theorem 3.4 proves that the innermost runtime complexity of $\mathcal{R}_{\mathsf{d}}$ is exponentially bounded.

On the other hand, the runtime complexity of $\mathcal{R}_{\mathsf{d}}$ (with respect to full rewriting) grows strictly faster than any exponential: Consider for arbitrary $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$ the term $\mathsf{f}(\mathsf{s}^n(\mathsf{0}),t)$. We verify, for $n > 0$, $\mathrm{dh}(\mathsf{f}(\mathsf{s}^n(\mathsf{0}),t),\to_{\mathcal{R}}) \geqslant 2^{2^{n-1}} \cdot (1 + \mathrm{dh}(t,\to_{\mathcal{R}}))$ by induction on n. For $m \in \mathbb{N}$, set $\underline{m} := \mathsf{s}^m(\mathsf{0})$. Consider the base case $n = 1$. Observe that, unlike for innermost rewriting, $\mathsf{f}(\underline{1},t) \to^5_{\mathcal{R}} \mathsf{c}(t,t)$. Since $\mathrm{dh}(\mathsf{c}(t,t),\to_{\mathcal{R}}) = 2 \cdot \mathrm{dh}(t,\to_{\mathcal{R}})$, the claim is easy to establish for this case. For the inductive step, consider a maximal derivation $\mathsf{f}(\underline{n+1},t) \to_{\mathcal{R}} \mathsf{f}(\underline{n},\mathsf{d}(\mathsf{f}(\underline{n},t))) \to_{\mathcal{R}} \cdots$. Applying induction hypothesis twice we obtain

$$\begin{aligned}\mathrm{dh}(\mathsf{f}(\underline{n+1},t),\to_{\mathcal{R}}) > \mathrm{dh}(\mathsf{f}(\underline{n},\mathsf{d}(\mathsf{f}(\underline{n},t))),\to_{\mathcal{R}}) &> \mathrm{dh}(\mathsf{f}(\underline{n},\mathsf{f}(\underline{n},t)),\to_{\mathcal{R}}) \\ &> 2^{2^{n-1}} \cdot (2^{2^{n-1}} \cdot (1 + \mathrm{dh}(t,\to_{\mathcal{R}}))) \\ &= 2^{2^n} \cdot (1 + \mathrm{dh}(t,\to_{\mathcal{R}})) \ .\end{aligned}$$

4 Exponential Path Order EPO

In this section we introduce the aforementioned order EPO that is used in the proof of Theorem 3.4. We slightly extend the definitions and results originally presented by the second author in [15].

The path order EPO is defined over *sequences* of terms from $\mathcal{T}(\mathcal{F},\mathcal{V})$. To denote sequences, we use an auxiliary function symbol $\mathsf{list} \notin \mathcal{F}$. The function symbol list is variadic, i.e., the arity of list is finite, but arbitrary. We write $[t_1 \ \cdots \ t_n]$ instead of $\mathsf{list}(t_1,\dots,t_n)$. For sequences $[s_1 \ \cdots \ s_n]$ and $[t_1 \ \cdots \ t_m]$, we write $[s_1 \ \cdots \ s_n]\frown[t_1 \ \cdots \ t_m]$ to denote the concatenation $[s_1 \ \cdots \ s_n \ t_1 \ \cdots \ t_m]$. We write $\mathcal{T}^\star(\mathcal{F},\mathcal{V})$ for the set of finite sequences of terms from $\mathcal{T}(\mathcal{F},\mathcal{V})$, i.e. $\mathcal{T}^\star(\mathcal{F},\mathcal{V}) := \{[t_1 \ \cdots \ t_n] \mid n \in \mathbb{N} \text{ and } t_1,\dots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{V})\}$. Each term $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$ is identified with the single list $[t] = \mathsf{list}(t) \in \mathcal{T}^\star(\mathcal{F},\mathcal{V})$. This identification ensures $\mathcal{T}(\mathcal{F},\mathcal{V}) \subseteq \mathcal{T}^\star(\mathcal{F},\mathcal{V})$. We use $a,b,c,\dots$ to denote elements of $\mathcal{T}^\star(\mathcal{F},\mathcal{V})$, possibly extending them by subscripts.

▶ **Definition 4.1.** Let $a, b \in \mathcal{T}^\star(\mathcal{F}, \mathcal{V})$, and let $\ell \geqslant 1$. Below we assume $f, g \in \mathcal{F}$. We define $a >^\ell_{\mathsf{epo}} b$ with respect to the precedence $\gtrsim$ if either

1) $a = f(s_1, \dots, s_m)$ and $s_i \geqslant^\ell_{\mathsf{epo}} b$ for some $i \in \{1, \dots, m\}$, or
2) $a = f(s_1, \dots, s_m)$, $b = [t_1 \ \cdots \ t_n]$ with $n = 0$ or $2 \leqslant n \leqslant \ell$, f is a defined function symbol, and $a >^\ell_{\mathsf{epo}} t_j$ for all $j \in \{1, \dots, n\}$, or
3) $a = f(s_1, \dots, s_m)$, $b = g(t_1, \dots, t_n)$ with $n \leqslant \ell$, f is a defined function symbol with $f > g$, and a is a strict superterm (modulo $\sim$) of all t_j ($j \in \{1, \dots, n\}$), or
4) $a = [s_1 \ \cdots \ s_m]$, $b = b_1 \frown \cdots \frown b_m$, and for some $j \in \{1, \dots, m\}$,
   - $s_1 \sim b_1, \dots, s_{j-1} \sim b_{j-1}$,
   - $s_j >^\ell_{\mathsf{epo}} b_j$, and
   - $s_{j+1} \geqslant^\ell_{\mathsf{epo}} b_{j+1}, \dots, s_m \geqslant^\ell_{\mathsf{epo}} b_m$, or
5) $a = f(s_1, \dots, s_m)$, $b = g(t_1, \dots, t_n)$ with $n \leqslant \ell$, f and g are defined function symbols with $f \sim g$, and for some $j \in \{1, \dots, \min(m, n)\}$,
   - $s_1 \sim t_1, \dots, s_{j-1} \sim t_{j-1}$,
   - $s_j \rhd\!\!/_{\sim} t_j$, and
   - $a \rhd\!\!/_{\sim} t_{j+1}, \dots, a \rhd\!\!/_{\sim} t_n$.

Here we set $\geqslant^\ell_{\mathsf{epo}} := >^\ell_{\mathsf{epo}} \cup \sim$. Finally, we set $>_{\mathsf{epo}} := \bigcup_{k \geqslant 1} >^\ell_{\mathsf{epo}}$ and $\geqslant_{\mathsf{epo}} := \bigcup_{k \geqslant 1} \geqslant^\ell_{\mathsf{epo}}$.

We note that, by Definition 4.1.2 with $n = 0$, we have $f(s_1, \dots, s_m) >^\ell_{\mathsf{epo}} [\,]$ for all $\ell \geqslant 1$ if f is a defined function symbol. It is not difficult to see that $l \leqslant k$ implies $>^l_{\mathsf{epo}} \subseteq >^k_{\mathsf{epo}}$. Unfortunately EPO is not a restriction of lexicographic path orders, as the length of lists is not bounded globally. However, the critical Clause 4 amounts to a *lifting* from terms to sequences of terms in the sense of [17, Section 3]. Conclusively an application of the main result of [17, Section 3] gives well-foundedness of $>^\ell_{\mathsf{epo}}$.

▶ **Lemma 4.2.** *Let $a = a_1 \frown \cdots \frown a_{j-1} \frown a_j \frown a_{j+1} \cdots \frown a_m$. Suppose that $a_j >^\ell_{\mathsf{epo}} b$. Then $a >^\ell_{\mathsf{epo}} a_1 \frown \cdots \frown a_{j-1} \frown b \frown a_{j+1} \cdots \frown a_m$.*

Following Arai and the second author [2] we define G_ℓ that measures the $>^\ell_{\mathsf{epo}}$-descending lengths:

▶ **Definition 4.3.** We define $G_\ell : \mathcal{T}^\star(\mathcal{F}, \mathcal{V}) \to \mathbb{N}$ as

$$G_\ell(a) := \max\{G_\ell(b) + 1 \mid b \in \mathcal{T}^\star(\mathcal{F}, \mathcal{V}) \text{ and } a >^\ell_{\mathsf{epo}} b\} \ .$$

▶ **Lemma 4.4.** *For all $\ell \geqslant 1$ we have*

1) $\rhd\!\!/_{\sim} \subseteq >^\ell_{\mathsf{epo}}$,
2) if $t \in \mathcal{T}(\mathcal{C}, \mathcal{V})$ then $G_\ell(t) = \mathsf{dp}(t)$, and
3) $G_\ell([t_1 \ \cdots \ t_m]) = \sum_{i=1}^m G_\ell(t_i)$.

Proof. The Properties 1) and 2) can be shown by straight forward inductive arguments. We prove Property 3) for the non-trivial case $m \geqslant 2$. It is not difficult to check that $G_\ell([t_1 \ \cdots \ t_m]) \geqslant \sum_{i=1}^m G_k(t_i)$. We show that $G_\ell([t_1 \ \cdots \ t_m]) \leqslant \sum_{i=1}^m G_\ell(t_i)$ by induction on $G_\ell([t_1 \ \cdots \ t_m])$.

Let $a = [t_1 \ \cdots \ t_m]$. Then, it suffices to show that for any $b \in \mathcal{T}^\star(\mathcal{F}, \mathcal{V})$, $a >^\ell_{\mathsf{epo}} b$ implies $G_\ell(b) < \sum_{i=1}^m G_\ell(t_i)$. Fix $b \in \mathcal{T}^\star(\mathcal{F}, \mathcal{V})$ and suppose that $a >^\ell_{\mathsf{epo}} b$. Then, by Definition 4.1.4, there exist some $b_1, \dots, b_m \in \mathcal{T}^\star(\mathcal{F}, \mathcal{V})$ and $j \in \{1, \dots, m\}$ such that $b = b_1 \frown \cdots \frown b_m$, $t_i \geqslant^\ell_{\mathsf{epo}} b_i$ for each $i \in \{1, \dots m\}$, and $t_j >^\ell_{\mathsf{epo}} b_j$. By the definition of G_ℓ, we have that $G_\ell(t_i) \geqslant G_\ell(b_i)$ for each $i \in \{1, \dots m\}$, and $G_\ell(t_j) > G_\ell(b_j)$. Thus $\sum_{i=1}^m G_\ell(b_i) < \sum_{i=1}^m G_\ell(t_i)$ follows. Let $b_i = [u_{i,1} \cdots u_{i,n_i}]$ for each $i \in \{1, \dots, m\}$. Then, since $G_\ell(b) < G_\ell(a)$, induction hypothesis gives $G_\ell(b) \leqslant \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_\ell(u_{i,j})$. Recalling that $\sum_{j=1}^{n_i} G_\ell(u_{i,j}) \leqslant G_\ell(b_i)$ also holds for each $i \in \{1, \ldots, m\}$. Summing up, we obtain that $G_\ell(b) \leqslant \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_\ell(u_{i,j}) \leqslant \sum_{i=1}^{m} G_\ell(b_i) < \sum_{i=1}^{m} G_\ell(t_i)$. ◀

We finally arrive at the main theorem of this section.

▶ **Theorem 4.5.** *Suppose that $f \in \mathcal{F}$ with arity $n \leqslant \ell$ and $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. Let $N := \max\{G_\ell(t_i) \mid 1 \leqslant i \leqslant n\} + 1$. Then*

$$G_\ell(f(t_1, \ldots, t_n)) \leqslant (\ell + 1)^{N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell - i} G_\ell(t_i)} . \tag{1}$$

Proof. Let $t = f(t_1, \ldots, t_n)$. We prove the inequality (1) by induction on $G_\ell(t)$. In the base case, $G_\ell(t) = 0$, and hence the inequality (1) trivially holds. In the case $G_\ell(t) > 0$, it suffices to show that for any $b \in \mathcal{T}^\star(\mathcal{F}, \mathcal{V})$, $t >^\ell_{\mathsf{epo}} b$ implies $G_\ell(b) < (\ell + 1)^{N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell - i} G_\ell(t_i)}$. The induction case splits into four cases depending on which rule of Definition 4.1 concludes $t >^\ell_{\mathsf{epo}} b$. For the sake of convenience, we start with the case corresponding to Definition 4.1.2. Namely, we consider the case $b = [s_1 \ \cdots \ s_k]$ where $2 \leqslant k \leqslant \ell$ and $t >^\ell_{\mathsf{epo}} s_i$ for all $i \in \{1, \ldots, k\}$. We show that for all $i \in \{1, \ldots, k\}$,

$$G_\ell(s_i) \leqslant (\ell + 1)^{(N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell - i} G_\ell(t_i)) - 1} . \tag{2}$$

We prove the inequality (2) by case analysis according to the last rule that concludes $t >^\ell_{\mathsf{epo}} s_i$. Fix some element $u \in \{s_i \mid i \in \{1, \ldots, k\}\}$.

1) Case $t_j \geqslant^\ell_{\mathsf{epo}} u$ for some $j \in \{1, \ldots, n\}$: In this case we trivially see

$$G_\ell(u) \leqslant G_\ell(t_j) \leqslant (\ell + 1)^{(N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell - i} G_\ell(t_i)) - 1} . \tag{3}$$

2) Case $u = g(u_1, \ldots, u_m)$ where $m \leqslant \ell$, g is a defined symbol with $f > g$ and for all $i \in \{1, \ldots, m\}$, t is a strict superterm (modulo $\sim$) of u_i: Let $M := \max\{G_\ell(u_i) \mid 1 \leqslant i \leqslant m\} + 1$. Then, we have $M \leqslant N$ since t is a strict superterm (modulo $\sim$) of every u_i. We claim

$$M^\ell \cdot \mathsf{rk}(g) + \sum_{i=1}^{m} M^{\ell - i} G_\ell(u_i) < N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell - i} G_\ell(t_i) .$$

To see this, conceive left- and right-hand side as numbers represented in base M and respectively N of length ℓ (observe $G_\ell(u_i) < M$ and $G_\ell(t_i) < N$). From $\mathsf{rk}(g) < \mathsf{rk}(f)$ and $M \leqslant N$ the above inequality is obvious. This allows us to conclude

$$\begin{aligned} G_\ell(u) &\leqslant (\ell + 1)^{M^\ell \cdot \mathsf{rk}(g) + \sum_{i=1}^{m} M^{\ell - i} G_\ell(u_i)} \\ &\leqslant (\ell + 1)^{N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell - i} G_\ell(t_i) - 1} . \end{aligned} \tag{4}$$

Here the first inequality follows by induction hypothesis.

3) Case $u = g(u_1, \ldots, u_m)$ where $m \leqslant \ell$, g is a defined symbol with $f \sim g$ and there exists $j \in \{1, \ldots, \min(n, m)\}$ such that $t_i \sim u_i$ for all $i < j$, t_j is a strict superterm (modulo $\sim$) of u_j, and t is a strict superterm (modulo $\sim$) of u_i for all $i > j$: Let $M := \max\{G_\ell(u_i) \mid 1 \leqslant i \leqslant m\} + 1$ and consider the following claim:

▶ **Claim 4.6.** $\sum_{i=1}^{m} M^{\ell - i} G_\ell(u_i) < \sum_{i=1}^{n} N^{\ell - i} G_\ell(t_i)$.

To prove this claim, observe that the assumptions give $G_\ell(u_i) = G_\ell(t_i)$ for all $i < j$, $G_\ell(u_j) < G_\ell(t_j)$, and $G_\ell(u_i) < N$ for all $i > j$: This implies that $M \leqslant N$ and

$$\sum_{i=1}^{m} M^{\ell-i} G_\ell(u_i) \leqslant \sum_{i=1}^{j-1} N^{\ell-i} G_\ell(t_i) + N^{\ell-j}(G_\ell(t_j) - 1) + \sum_{i=j+1}^{n} N^{\ell-i}(N-1)$$
$$< \sum_{i=1}^{n} N^{\ell-i} G_\ell(t_i) \ .$$

As above, the claim together with induction hypothesis yields

$$G_\ell(u) \leqslant (\ell+1)^{N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell-i} G_\ell(t_i) - 1} \ . \qquad (5)$$

Summing up the inequality (3), (4) and (5) concludes inequality (2). Thus, having $G_\ell(b) = \sum_{i=1}^{k} G_\ell(s_i)$ by Lemma 4.4, and employing $k \leqslant \ell$, we see

$$G_\ell(b) \leqslant \ell \cdot (\ell+1)^{(N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell-i} G_\ell(t_i)) - 1} \qquad \text{(by the inequality (2))}$$
$$< (\ell+1)^{N^\ell \cdot \mathsf{rk}(f) + \sum_{i=1}^{n} N^{\ell-i} G_\ell(t_i)} \ .$$

This completes the case for Definition 4.1.2. The cases for Definition 4.1.1, 4.1.3 and 4.1.5 follow respectively from the inequality (3), (4) and (5). ◀

5 Embedding EPO* in EPO

In this section we define *predicative interpretations* $\mathcal{I}$ that embed innermost rewrite steps into $>^\ell_{\mathsf{epo}}$, i.e., if $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$, then $\mathcal{I}(s) >^\ell_{\mathsf{epo}} \mathcal{I}(t)$. The definition of $\mathcal{I}$ makes use of the mapping safe underlying the definition of $>_{\mathsf{epo}\star}$. Based on this embedding we then use Theorem 4.5 to prove that EPO* induces exponential (innermost) runtime complexity (Theorem 3.4).

Before we define predicative interpretations, we start with a simple observation. Let $\mathcal{R}$ be a TRS compatible with some instance $>_{\mathsf{epo}\star}$, i.e., $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$. For the moment, suppose $\mathcal{R}$ is completely defined. We replace this restriction by constructor TRS later on. Since $\mathcal{R}$ is completely defined, normal forms and constructor terms coincide, and thus $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$ if $s = C[l\sigma], t = C[r\sigma]$ for some rule $l \to r \in \mathcal{R}$ where additionally $l\sigma \in \mathcal{B}$. Let t be obtained by rewriting a basic term s. By the inclusion $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$, every normal argument t_i of t is irreducible, i.e., $t_i \in \mathcal{T}(\mathcal{C}, \mathcal{V})$. We capture this observation in the definition of $\mathcal{B}^{\rightarrow}$:

▶ **Definition 5.1.** The set $\mathcal{B}^{\rightarrow}$ is the least set of terms such that

1) $\mathcal{T}(\mathcal{C}, \mathcal{V}) \subseteq \mathcal{B}^{\rightarrow}$, and
2) if $f \in \mathcal{F}$, $\vec{s} \subseteq \mathcal{T}(\mathcal{C}, \mathcal{V})$ and $\vec{t} \subseteq \mathcal{B}^{\rightarrow}$ then $f(\vec{s}; \vec{t}) \in \mathcal{B}^{\rightarrow}$.

Note that $\mathcal{B} \subseteq \mathcal{B}^{\rightarrow}$. The verification of the next Lemma is straight forward:

▶ **Lemma 5.2.** *Let $\mathcal{R}$ be a completely defined TRS compatible with $>_{\mathsf{epo}\star}$. If $s \in \mathcal{B}^{\rightarrow}$ and $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$ then $t \in \mathcal{B}^{\rightarrow}$.*

We define predicative interpretation $\mathcal{I}$ as follows. Since we are only interested in the length of derivations starting from basic terms, Lemma 5.2 justifies that only terms from $\mathcal{B}^{\rightarrow}$ are considered. For each defined symbol f, let f_{n} be a fresh function symbol, and let $\mathcal{F}_{\mathsf{n}} = \{f_{\mathsf{n}} \mid f \in \mathcal{D}\} \cup \mathcal{C}$. Here the arity of f_{n} is k where $\mathsf{nrm}(f) = \{i_1, \ldots, i_k\}$, moreover f_{n} is still considered a defined function symbol when applying Definition 4.1. We extend the (admissible) precedence $\gtrsim$ to $\mathcal{F}_{\mathsf{n}}$ in the obvious way: $f_{\mathsf{n}} \sim g_{\mathsf{n}}$ if $f \sim g$ and $f_{\mathsf{n}} > g_{\mathsf{n}}$ if $f > g$.

▶ **Definition 5.3.** A *predicative interpretation* $\mathcal{I}$ is a mapping $\mathcal{I}\colon \mathcal{B}^{\rightarrow} \to \mathcal{T}^{\star}(\mathcal{F},\mathcal{V})$ defined as follows:

1) $\mathcal{I}(t) = [\,]$ if $t \in \mathcal{T}(\mathcal{C},\mathcal{V})$, and otherwise
2) $\mathcal{I}(t) = [f_{\mathsf{n}}(t_1,\dots,t_k)]\frown\mathcal{I}(t_{k+1})\frown\cdots\frown\mathcal{I}(t_{k+n})$ where $t = f(t_1,\dots,t_k \,;\, t_{k+1},\dots,t_{k+n})$.

The next lemma provides the embedding of root steps for completely defined, compatible, TRSs $\mathcal{R}$. Here we could simply define $\mathcal{I}(t) = f_{\mathsf{n}}(t_1,\dots,t_k)$ in Case 2. The complete definition becomes only essential when we look at closure under context in Lemma 5.5.

▶ **Lemma 5.4.** *Let $s \in \mathcal{B}$ and let $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{C},\mathcal{V})$ be a substitution. If $s >_{\mathsf{epo}\star} t$ then $\mathcal{I}(s\sigma) >_{\mathsf{epo}}^{|t|} \mathcal{I}(t\sigma)$.*

Proof. Let f denote the (defined) root symbol of s, and let $s_1,\dots,s_l$ denote the normal arguments of s. Thus $\mathcal{I}(s\sigma) = [f_{\mathsf{n}}(s_1\sigma,\dots,s_l\sigma)] = f_{\mathsf{n}}(s_1\sigma,\dots,s_l\sigma)$. If $t \in \mathcal{T}(\mathcal{C},\mathcal{V})$ then the lemma trivially follows as $\mathcal{I}(t\sigma) = [\,]$. Hence suppose $t \notin \mathcal{T}(\mathcal{C},\mathcal{V})$.

We continue by induction on the definition of $>_{\mathsf{epo}\star}$. Let $t = g(t_1,\dots,t_k \,;\, t_{k+1},\dots,t_{k+n})$ and so

$$\mathcal{I}(t\sigma) = [g_{\mathsf{n}}(t_1\sigma,\dots,t_k\sigma)]\frown\mathcal{I}(t_{k+1}\sigma)\frown\cdots\frown\mathcal{I}(t_{k+n}\sigma)\ .$$

Observe that $\mathcal{I}(x\sigma) = [\,]$ for all variables x in t. Using this we see that the length of the list $\mathcal{I}(t\sigma)$ is bound by $|t|$. Hence by Definition 4.1.2, it suffices to verify $\mathcal{I}(s\sigma) >_{\mathsf{epo}}^{|t|} \mathcal{I}(t_i\sigma)$ for all safe arguments t_i ($i \in \{k+1,\dots,k+n\}$), and further to show

$$f_{\mathsf{n}}(s_1\sigma,\dots,s_l\sigma) >_{\mathsf{epo}}^{|t|} g_{\mathsf{n}}(t_1\sigma,\dots,t_k\sigma)\ . \tag{6}$$

Since $s \in \mathcal{B}$ but $t \notin \mathcal{T}(\mathcal{C},\mathcal{V})$, a consequence of Lemma 3.3 is that $s >_{\mathsf{epo}\star} t$ follows either by Definition 3.2.2 or Definition 3.2.3. Let t_i be a safe argument. Then by definition $s >_{\mathsf{epo}\star} t_i$ and induction hypothesis yields $\mathcal{I}(s\sigma) >_{\mathsf{epo}}^{|t|} \mathcal{I}(t_i\sigma)$ (employing $|t_i| \leqslant |t|$). It thus remains to verify (6). We continue by case analysis.

1) Suppose $f > g$, i.e., Definition 3.2.2 applies. Then $f_{\mathsf{n}} > g_{\mathsf{n}}$ by definition. By Definition 4.1.3 it suffices to prove $f_{\mathsf{n}}(s_1\sigma,\dots,s_l\sigma) \rhd\!/\!_{\sim}\ t_i\sigma$ for all $i \in \{1,\dots,k\}$. Fix $i \in \{1,\dots,k\}$. According to Definition 3.2.2 $s \sqsupset_{\mathsf{epo}\star} t_i$ holds, and thus there exists $j \in \{1,\dots,l\}$ such that $s_j \sqsupseteq_{\mathsf{epo}\star} t_i$. Hence $s_j \unrhd\!/\!_{\sim}\ t_i$ by Lemma 3.3, from which we conclude $f_{\mathsf{n}}(s_1\sigma,\dots,s_l\sigma) \rhd\!/\!_{\sim}\ t_i\sigma$ since we suppose $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{C},\mathcal{V})$.
2) Suppose $f \sim g$, i.e., Definition 3.2.3 applies. Then $f_{\mathsf{n}} \sim g_{\mathsf{n}}$. By Definition 4.1.5 it suffices to prove (i) $s_1\sigma \sim t_1\sigma$, …, $s_{\ell-1}\sigma \sim t_{\ell-1}\sigma$, (ii) $s_\ell\sigma \rhd\!/\!_{\sim}\ t_\ell\sigma$, and further (iii) $f_{\mathsf{n}}(s_1\sigma,\dots,s_l\sigma) \rhd\!/\!_{\sim}\ t_{\ell+1}\sigma$, …, $f_{\mathsf{n}}(s_1\sigma,\dots,s_l\sigma) \rhd\!/\!_{\sim}\ t_k\sigma$ for some $\ell \in \{1,\dots,k\}$. The assumptions in Definition 3.2.3 yield $s_1 \overset{s}{\sim} t_1$, …, $s_{\ell-1} \overset{s}{\sim} t_{\ell-1}$ from which we conclude (i), further $s_\ell \sqsupset_{\mathsf{epo}\star} t_\ell$ from which we conclude (ii) with the help of Lemma 3.3 (using $s_\ell \in \mathcal{T}(\mathcal{C},\mathcal{V})$), and finally $s \sqsupset_{\mathsf{epo}\star} t_{\ell+1}$, …, $s \sqsupset_{\mathsf{epo}\star} t_k$ from which we obtain (iii) as in the case above.

◀

▶ **Lemma 5.5.** *Let $s,t \in \mathcal{B}^{\rightarrow}$ and let C be a context such that $C[s] \in \mathcal{B}^{\rightarrow}$. If $\mathcal{I}(s) >_{\mathsf{epo}}^{\ell} \mathcal{I}(t)$ then $\mathcal{I}(C[s]) >_{\mathsf{epo}}^{\ell} \mathcal{I}(C[t])$.*

Proof. We show the lemma by induction on C. It suffices to consider the step case. Observe that by the assumption $\mathcal{I}(s) >_{\mathsf{epo}}^{\ell} \mathcal{I}(t)$, $s \notin \mathcal{T}(\mathcal{C},\mathcal{V})$ since otherwise $\mathcal{I}(s) = [\,]$ is $>_{\mathsf{epo}}^{\ell}$-minimal. By definition of $\mathcal{B}^{\rightarrow}$ we can thus assume $C = f(s_1,\dots,s_k \,;\, s_{k+1},\dots,C'[\Box],\dots s_{k+l})$ for some context C'. Thus, for each $u \in \{s,t\}$,

$$\mathcal{I}(C[u]) = [f_{\mathsf{n}}(s_1,\dots,s_k)]\frown\mathcal{I}(s_{k+1})\frown\cdots\frown\mathcal{I}(C'[u])\frown\cdots\frown\mathcal{I}(s_{k+l})\ .$$

By induction hypothesis $\mathcal{I}(C'[s]) >_{\mathsf{epo}}^{\ell} \mathcal{I}(C'[t])$. We conclude using Lemma 4.2. ◀

Combining Lemma 5.4 and Lemma 5.5 completes the embedding.

▶ **Lemma 5.6.** *Let $\mathcal{R}$ be a completely defined TRS compatible with $>_{\mathsf{epo\star}}$. Set $\ell := \max\{|r| \mid l \to r \in \mathcal{R}\}$. If $s \in \mathcal{B}^{\to}$ and $s \xrightarrow{\mathrm{i}}_{\mathcal{R}} t$ then $\mathcal{I}(s) >^{\ell}_{\mathsf{epo}} \mathcal{I}(t)$.*

Proof. Suppose $s \xrightarrow{\mathrm{i}}_{\mathcal{R}} t$. Hence there exists a context C, substitution σ and rule $l \to r \in \mathcal{R}$ such that $s = C[l\sigma]$ and $t = C[r\sigma]$. By the assumption that $\mathcal{R}$ is completely defined, $l \in \mathcal{B}$ and $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{C}, \mathcal{V})$. Since $\mathcal{R} \subseteq >_{\mathsf{epo\star}}$, we obtain $\mathcal{I}(l\sigma) >^{\ell}_{\mathsf{epo}} \mathcal{I}(r\sigma)$ by Lemma 5.4 (additionally employing $>^{|r|}_{\mathsf{epo}} \subseteq >^{\ell}_{\mathsf{epo}}$). Lemma 5.5 then establishes $\mathcal{I}(s) >^{\ell}_{\mathsf{epo}} \mathcal{N}(t)$. ◀

We obtain Theorem 3.4 formulated for completely defined TRSs.

▶ **Theorem 5.7.** *Let $\mathcal{R}$ be a completely defined, possibly infinite, TRS compatible with $>_{\mathsf{epo\star}}$. Suppose $\max\{|r| \mid l \to r \in \mathcal{R}\}$ is well-defined. There exists $k \in \mathbb{N}$ such that $\mathrm{rc}^{\mathrm{i}}_{\mathcal{R}}(n) \leqslant 2^{\mathsf{O}(n^k)}$.*

Proof. Set $\ell := \max\{\max\{|r| \mid l \to r \in \mathcal{R}\}, \max\{\mathsf{ar}(f) \mid f \in \mathcal{F}\}\}$. Not that ℓ is well-defined as $\mathcal{F}$ is finite and non-variadic. We prove the existence of $c_1, c_2 \in \mathbb{N}$ so that for any $s \in \mathcal{B}$, $\mathrm{dh}(s, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \leqslant 2^{c_1 \cdot |s|^{c_2}}$. Consider a maximal derivation $s = t_0 \xrightarrow{\mathrm{i}}_{\mathcal{R}} t_1 \xrightarrow{\mathrm{i}}_{\mathcal{R}} \cdots \xrightarrow{\mathrm{i}}_{\mathcal{R}} t_n$ with $s \in \mathcal{B}$. Let $i \in \{0, \dots, n-1\}$. We observed $t_i \in \mathcal{B}^{\to}$ in Lemma 5.2, and thus $\mathcal{I}(t_i) >^{\ell}_{\mathsf{epo}} \mathcal{I}(t_{i+i})$ due to Lemma 5.6. So in particular $\mathrm{dh}(s, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \leqslant G_\ell(\mathcal{I}(s))$. It remains to estimate $G_\ell(\mathcal{I}(s))$ in terms of $|s|$: for this, suppose $s = f(s_1, \dots, s_k \,;\, s_{k+1}, \dots, s_{k+l})$ for some $f \in \mathcal{D}$ and $s_i \in \mathcal{T}(\mathcal{C}, \mathcal{V})$ $(i \in \{1, \dots, k+l\})$. By definition $\mathcal{I}(s) = f_{\mathsf{n}}(s_1, \dots, s_k)$. Set $N := \max\{G_\ell(s_i) \mid 1 \leqslant i \leqslant k\} + 1$, and verify

$$N \leqslant 1 + \sum_{i=1}^{k} G_\ell(s_i) \leqslant 1 + \sum_{i=1}^{k} \mathsf{dp}(s_i) \leqslant |s| \ . \tag{7}$$

For the second inequality we employ Lemma 4.4, which gives $G_\ell(s_i) = \mathsf{dp}(s_i)$ as $s_i \in \mathcal{T}(\mathcal{C}, \mathcal{V})$ for all $i \in \{1, \dots, k\}$. Applying Theorem 4.5 we see

$$\begin{aligned}
G_\ell(\mathcal{I}(s)) &= G_\ell(f_{\mathsf{n}}(s_1, \dots, s_k)) \\
&\leqslant (\ell+1)^{N^\ell \cdot \mathsf{rk}(f_{\mathsf{n}}) + \sum_{i=1}^{k} N^{\ell-i} \cdot G_\ell(s_i)} && \text{(by Theorem 4.5, using } k \leqslant \ell\text{)} \\
&\leqslant (\ell+1)^{|s|^\ell \cdot \mathsf{rk}(f_{\mathsf{n}}) + |s|^\ell \cdot \sum_{i=1}^{k} G_\ell(s_i)} && \text{(by Equation 7)} \\
&\leqslant (\ell+1)^{|s|^\ell \cdot \mathsf{rk}(f_{\mathsf{n}}) + |s|^\ell \cdot |s|} && \text{(by Equation 7)} \\
&\leqslant (\ell+1)^{(\mathsf{rk}(f_{\mathsf{n}})+1) \cdot |s|^{\ell+1}} \ .
\end{aligned}$$

Since ℓ depends only on $\mathcal{R}$ and $\mathcal{F}$, and $\mathsf{rk}(f_{\mathsf{n}})$ is bounded by some constant depending only on $\mathcal{F}$, simple arithmetical reasoning gives the constants c_1, c_2 such that $\mathrm{dh}(s, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \leqslant G_\ell(\mathcal{I}(s)) \leqslant 2^{c_1 \cdot |s|^{c_2}}$. This concludes the Theorem. ◀

We now lift the restriction that $\mathcal{R}$ is completely defined for constructor TRSs $\mathcal{R}$. The idea is to extend $\mathcal{R}$ with sufficiently many rules so that the resulting system is completely defined and Theorem 5.7 applicable.

▶ **Definition 5.8.** Let $\bot$ be a fresh constructor symbol. We define

$$\mathcal{S} := \{t \to \bot \mid t \in \mathcal{T}(\mathcal{F} \cup \{\bot\}, \mathcal{V}) \cap \mathrm{NF}(\mathcal{R}) \text{ and the root symbol of } t \text{ is defined}\} \ .$$

We extend the precedence $\gtrsim$ to $\mathcal{F} \cup \{\bot\}$ so that $\bot$ is minimal. Thus $\mathcal{S} \subseteq >_{\mathsf{epo\star}}$ follows by one application of Definition 3.2.2. Further, the *completely defined* TRS $\mathcal{R} \cup \mathcal{S}$ is able to simulate $\xrightarrow{\mathrm{i}}_{\mathcal{R}}$ derivations for constructor TRS $\mathcal{R}$:

▶ **Lemma 5.9.** *Suppose $\mathcal{R}$ is a constructor TRS. Then $\mathcal{R} \cup \mathcal{S}$ is completely defined. Further, if $s \xrightarrow{\mathsf{i}}{}^{\ell}_{\mathcal{R}} t$ then $s \xrightarrow{\mathsf{i}}{}^{\ell'}_{\mathcal{R}\cup\mathcal{S}} t'$ for some t' and $\ell' \geqslant \ell$.*

For the latter property we show that $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$ implies $s' \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{R}\cup\mathcal{S}} t'$ for s' and t' $\mathcal{S}$-normal forms of s and t. Here the key observation is that rewriting according to $\mathcal{S}$ does not interfere with pattern matching with respect to $\mathcal{R}$.

An immediate consequence of Lemma 5.9 is $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}(n) \leqslant \mathrm{rc}^{\mathsf{i}}_{\mathcal{R}\cup\mathcal{S}}(n)$, i.e., the innermost runtime complexity of $\mathcal{R}$ can be analysed through $\mathcal{R} \cup \mathcal{S}$. We arrive at the proof of our main theorem:

Proof of Theorem 3.4. Suppose $\mathcal{R}$ is a constructor TRS compatible with $>_{\mathsf{epo}\star}$. We verify that $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}(n) \leqslant 2^{\mathsf{O}(n^k)}$ for some fixed $k \in \mathbb{N}$: let $\mathcal{S}$ be defined according to Definition 5.8. By Lemma 5.9, $\mathcal{R} \cup \mathcal{S}$ is completely defined, and moreover, $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}(n) \leqslant \mathrm{rc}^{\mathsf{i}}_{\mathcal{R}\cup\mathcal{S}}(n)$. Note that $\max\{|r| \mid l \to r \in \mathcal{R} \cup \mathcal{S}\} = \max\{|r| \mid l \to r \in \mathcal{R}\}$ is well-defined. Further $(\mathcal{R} \cup \mathcal{S}) \subseteq >_{\mathsf{epo}\star}$ follows by assumption and definition of $\mathcal{S}$. Hence all assumptions of Theorem 5.7 are fulfilled, and we conclude $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}(n) \leqslant \mathrm{rc}^{\mathsf{i}}_{\mathcal{R}\cup\mathcal{S}}(n) \leqslant 2^{\mathsf{O}(n^k)}$ for some $k \in \mathbb{N}$. ◀

6 Characterising Exponential Time Computation

In this section we present one application of EPO* in the context of *implicit computational complexity (ICC)*. Following [19, 11, 5] we give semantics to TRSs $\mathcal{R}$ as follows:

▶ **Definition 6.1.** Let $\mathcal{V}\mathrm{al} := \mathcal{T}(\mathcal{C}, \mathcal{V})$ denote the set of *values*. Further, let $\mathcal{P} \subseteq \mathcal{V}\mathrm{al}$ be a finite set of *non-accepting patterns*. We call a term t *accepting* (with respect to $\mathcal{P}$) if there exists no $p \in \mathcal{P}$ such that $p\sigma = t$ for some substitution σ. We say that $\mathcal{R}$ *computes the relation* $R \subseteq \mathcal{V}\mathrm{al} \times \mathcal{V}\mathrm{al}$ with respect to $\mathcal{P}$ if there exists $\mathsf{f} \in \mathcal{D}$ such that for all $s, t \in \mathcal{V}\mathrm{al}$,

$$s \, R \, t \qquad \Leftrightarrow \qquad \mathsf{f}(s) \xrightarrow{\mathsf{i}}{}^{!}_{\mathcal{R}} t \text{ and } t \text{ is accepting} .$$

On the other hand, we say that a relation R is computed by $\mathcal{R}$ if R is defined by the above equations with respect to *some* set $\mathcal{P}$ of non-accepting patterns.

For the case that $\mathcal{R}$ is *confluent* we also say that $\mathcal{R}$ computes the (partial) *function* induced by the relation R. Note that the restriction to binary relations is a non-essential simplification. The assertion that for normal forms t, t is accepting aims to eliminate by-products of the computation that should not be considered as part of the computed relation R.

As a consequence of Theorem 3.4 we derive our soundness result. Following [14, 5] we employ *graph rewriting* (c.f. [25]) to efficiently compute normal forms.

▶ **Theorem 6.2** (Soundness). *Suppose $\mathcal{R}$ is a constructor TRS compatible with $>_{\mathsf{epo}\star}$. The relations computed by $\mathcal{R}$ are computable in nondeterministic time $2^{\mathsf{O}(n^k)}$ for some $k \in \mathbb{N}$. In particular, if $\mathcal{R}$ is confluent then $f \in$ FEXP for each function f computed by $\mathcal{R}$.*

Proof. We sketch the implementation of the relation R_f (function f) on a Turing machine M_f.

Single out the corresponding defined function symbol f, and consider some arbitrary input $v \in \mathcal{V}\mathrm{al}$. First writing $\mathsf{f}(v)$ on a dedicated working tape, the machine M_f iteratively rewrites $\mathsf{f}(v)$ to normal form in an innermost fashion. For non-confluent TRSs $\mathcal{R}$, the choice of the redex is performed nondeterministically, otherwise some innermost redex is computed deterministically.

By the assumption $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$, Theorem 3.4 provides an upper bound $2^{|\mathsf{f}(v)|^{c_1}}$ on the number of iterations for some $c_1 \in \mathbb{N}$, i.e., the machine performs at most exponentially many iterations in the size of the input v. Thus the theorem follows if we can prove that each iteration is computable in time exponential in $|v|$.

To investigate into the complexity of a single iteration, consider the i-th iteration with t_i written on the working tape (where $\mathsf{f}(v) \rightarrow^i_{\mathcal{R}} t_i$). We want to compute some t_{i+1} with $t_i \xrightarrow{\mathsf{i}}_{\mathcal{R}} t_{i+1}$. Observe that in the presence of duplicating rules, $|t_i|$ might be exponential in i (and $|v|$). As we can only assume $i \leqslant 2^{|\mathsf{f}(v)|^{c_1}}$, we cannot hope to construct t_{i+1} from t_i in time exponential in $|v|$ if we use a representation of terms that is linear in size in the number of symbols.

Instead, we employ the machinery of [5]. By taking sharing into account, [5] achieves an encoding of t_i that is bounded in size polynomially in $|v|$ and i. Hence in particular t_i is encoded in size $2^{|s|^{c_2}}$ for some $c_2 \in \mathbb{N}$ depending only on $\mathcal{R}$. Further, a single step is computable in polynomial time (in the encoding size). And so t_{i+1} is computable from t_i in time $2^{|s|^{c_3}}$ for some $c_3 \in \mathbb{N}$ depending only on $\mathcal{R}$. Overall, we conclude that normal forms are computable in time $2^{|v|^{c_1}} \cdot 2^{|v|^{c_3}} = 2^{\mathsf{O}(|v|^k)}$ for some $k \in \mathbb{N}$ worst case.

After the final iteration, the machine M_f checks whether the computed normal form t_l is accepting and either accepts or rejects the computation. Using the machinery of [5] pattern matching is polynomial the encoding size of t_l, by the above bound on encoding sizes the operation is exponential in $|v|$. As v was chosen arbitrarily and k depends only on $\mathcal{R}$, we conclude the theorem. ◀

▶ **Example 6.3.** [Example 6.3 continued]. Since $\mathcal{R}_{\mathsf{fib}} \subseteq >_{\mathsf{epo}\star}$, Theorem 6.2 yields that the function $f_{\mathsf{fib}} : \mathcal{T}(\{\mathsf{0}, \mathsf{s}\}, \mathcal{V}) \to \mathcal{T}(\{\mathsf{0}, \mathsf{s}\}, \mathcal{V})$ computed by $\mathcal{R}_{\mathsf{fib}}$ is computable in exponential time.

In correspondence to Theorem 6.2, EPO* is also complete in the sense that every exponential time function is computable by a TRS compatible with EPO*. To prove completeness we use the characterisation of the exponential time computable functions $\mathcal{N}$ given by Arai and the second author [1], or more precisely the resulting *term rewriting characterisation* $\mathcal{R}_{\mathcal{N}}$ presented in [15].

Similar to the definition of EPO*, the class $\mathcal{N}$ relies on a syntactic separation of argument positions into *normal* and *safe* ones. To highlight this separation, we again write $f(\vec{x}; \vec{y})$ instead of $f(\vec{x}, \vec{y})$ for normal arguments $\vec{x}$ and safe arguments $\vec{y}$. The class $\mathcal{N}$ is defined as the least class containing certain initial functions that is closed under the scheme of *weak safe composition* (*WSC* for short) and *safe nested recursion on notation* (*SNRN* for short). In [1] it has been shown that $\mathcal{N}$ coincides with the class of exponential time functions FEXP. Below we give a brief definition of the above mentioned term rewriting characterisation of $\mathcal{N}$. Essentially, all the equations defining the functions from $\mathcal{N}$ are oriented from left to right, resulting in an *infinite* set of rewrite rules $\mathcal{R}_{\mathcal{N}}$.

For $k, l \in \mathbb{N}$, the signature $\mathcal{F}$ underlying $\mathcal{R}_{\mathcal{N}}$ is partitioned into sets $\mathcal{F}^{k,l}$, collecting function symbols with k normal and l safe arguments. To express natural numbers, the constructor $\mathsf{0} \in \mathcal{F}^{0,0}$, and dyadic successors $\mathsf{S}_1, \mathsf{S}_2 \in \mathcal{F}^{0,1}$ are used. Terms formed from $\mathsf{0}, \mathsf{S}_1$ and S_2 are called *numerals*. A numeral u encodes the natural number $\bar{u}$ as follows: $\bar{\mathsf{0}} := 0$, and $\overline{\mathsf{S}_i(; u)} := 2 \cdot \bar{u} + i$.

▶ **Definition 6.4.** The system $\mathcal{R}_{\mathcal{N}}$ contains the following rewrite rules, encoding the initial functions of $\mathcal{N}$:

$$\begin{array}{lll} \mathsf{O}^{k,l}(\vec{x};\vec{y}) \to \mathsf{0} & \text{for } k,l \in \mathbb{N} & \mathsf{P}(;\mathsf{0}) \to \mathsf{0} \\ \mathsf{I}^{k,l}_r(\vec{x};\vec{y}) \to x_r & \text{for } k,l \in \mathbb{N} \text{ and } r \in \{1,\dots,k\} & \mathsf{P}(;\mathsf{S}_i(;x)) \to x \quad \text{for } i \in \{1,2\} \\ \mathsf{I}^{k,l}_r(\vec{x};\vec{y}) \to y_{r-k} & \text{for } k,l \in \mathbb{N} \text{ and} & \mathsf{C}(;\mathsf{0},y_1,y_2) \to y_1 \\ & \quad r \in \{k+1,\dots,l+k\} & \mathsf{C}(;\mathsf{S}_i(;x),y_1,y_2) \to y_i \quad \text{for } i \in \{1,2\} \end{array}$$

Here $\vec{x} = x_1,\dots,x_k$ and $\vec{y} = y_1,\dots,y_l$ are supposed to be distinct variables.

Suppose the mapping safe is defined according to the definition of the rules above. Then each rule is oriented by an instance of EPO* regardless of the precedence used.

The scheme WSC is captured in the following definition.

▶ **Definition 6.5.** Suppose $g \in \mathcal{F}^{m,n}$ and $\vec{h} = h_1,\dots,h_n \in \mathcal{F}^{k,l}$. Then for each sequence of indices $1 \leqslant i_1,\dots,i_m \leqslant k$, the signature contains a fresh function symbol $\mathsf{SUB}[g,i_1,\dots,i_m,h_1,\dots,h_n] \in \mathcal{F}^{k,l}$. This symbol denotes the composition of functions g and $\vec{h}$ according to the rule

$$\mathsf{SUB}[g,i_1,\dots,i_m,h_1,\dots,h_n](\vec{x};\vec{y}) \to g(x_{i_1},\dots,x_{i_m};\vec{h}(\vec{x};\vec{y})) \ .$$

Here we use $\vec{h}(\vec{x};\vec{y})$ to abbreviate $h_1(\vec{x};\vec{y}),\dots,h_n(\vec{x};\vec{y})$, and we use $\vec{x} = x_1,\dots,x_k$ and $\vec{y} = y_1,\dots,y_l$ for distinct variables.

Note that the above rule can be oriented by EPO*. For that we can employ any precedence that complies with $\mathsf{SUB}[g,i_1,\dots,i_m,h_1,\dots,h_n] > g,\vec{h}$. The scheme reflects that the class of exponential time functions is *not* closed under composition in general. However, we are allowed to substitute function calls $h_i(\vec{x};\vec{y})$ in safe argument positions of g.

It remains to define the rules for the scheme SNRN. For that we make use of the following restriction of the lexicographic order.

▶ **Definition 6.6.** Let $\vec{u} = u_1,\dots,u_n$ and $\vec{v} = v_1,\dots,v_n$ be vectors of (possibly non-ground) numerals. We define $\vec{u} >^n_{\mathsf{lex}'} \vec{v}$ if there exists $k \in \{1,\dots,n\}$ such that i) $u_1,\dots,u_{k-1} = v_1,\dots,v_{k-1}$, ii) u_k is a binary successor of v_k (i.e., $u_k = \mathsf{S}_i(;v_k)$ for some $i \in \{1,2\}$), and iii) for each $j \in \{k+1,\dots,n\}$ there exists $i \in \{1,\dots,n\}$ such that $u_i = v_j$ or u_i is a binary successor of v_j.

Clearly the predecessor with respect to $>^n_{\mathsf{lex}'}$ is not unique. To precisely explain the relationship between arguments of the function and arguments replaced in recursive calls, we introduce the notion of a $>^n_{\mathsf{lex}'}$*-function* p.

The function p computes a suitable $>^n_{\mathsf{lex}'}$-predecessor of the normal arguments $\vec{u}$. We make use of the *type* $\tau(\vec{u})$ of $\vec{u}$, which is a ternary string over $\Sigma := \{0,1,2\}$: for single numeral, we set $\tau(\mathsf{0}) := 0$, $\tau(\mathsf{S}_1(;u)) := 1$ and $\tau(\mathsf{S}_2(;u)) := 2$. We extend τ to sequences of numerals $\tau(u_1,\dots,u_n) := \tau(u_1)\cdots\tau(u_n)$.

Thus roughly, $\tau(\vec{u})$ corresponds to the vector of most significant bits of $\vec{u}$. Abusing notation, let $\mathsf{S}_0(;u)$ denote $\mathsf{0}$. Then $\vec{u}$ with type $\tau(\vec{u}) = w_1\cdots w_n = w$ is expressible as $\mathsf{S}_{w_1}(;v_1),\dots,\mathsf{S}_{w_n}(;v_n)$, or short $\mathsf{S}_w(;\vec{v})$, for some numerals $\vec{v} = v_1,\dots,v_n$. The use of $>^n_{\mathsf{lex}'}$-functions relies on the following projection-function: for $n \in \mathbb{N}$ and $j \in \{1,\dots,2n\}$

$$J^n_j(u_1,\dots,u_n) := \begin{cases} u_j & \text{if } 1 \leqslant j \leqslant n, \text{ and} \\ v & \text{if } n+1 \leqslant j \leqslant 2n \text{ and } u_{j-n} = \mathsf{S}_i(;v) \ (i \in \Sigma) \ . \end{cases}$$

Further, consider a function $p\colon \{1,\dots,n\} \times \Sigma^n \to \{1,\dots,2n\}$. Based on p we extend the above function J^n, returning sequences of arguments as follows:

$$J^n_p(u_1,\dots,u_n) := J^n_{p(1,\tau(\vec{u}))}(u_1,\dots,u_n),\dots,J^n_{p(n,\tau(\vec{u}))}(u_1,\dots,u_n)\ .$$

Finally, the next definition provides the notion of a $>^n_{\mathsf{lex}'}$*-function* p.

▶ **Definition 6.7.** A function $p\colon \{1,\dots,n\} \times \Sigma^n \to \{1,\dots,2n\}$ is called a $>^n_{\mathsf{lex}'}$*-function* if for all vectors of numerals $u_1,\dots,u_n \neq \mathsf{0},\dots,\mathsf{0}$ we have $u_1,\dots,u_n >_{\mathsf{lex}'} J^n_p(u_1,\dots,u_n)$.

We complete the definition of $\mathcal{R}_{\mathcal{N}}$. We abbreviate $\Sigma^k \setminus \{\mathsf{0}\cdots\mathsf{0}\}$ as Σ^k_0.

▶ **Definition 6.8.** Suppose $g \in \mathcal{F}^{k,l}$ and $r_w, \vec{s_w}, \vec{t_w} \in \mathcal{F}^{k+k',l+1}$ for each type $w \in \Sigma^k_0$. Then for each triple $\vec{p} = p_1,p_2,p_3$ of $>^k_{\mathsf{lex}'}$-functions, the signature contains a fresh function symbol

$$\mathsf{f} = \mathsf{SNRN}_{\vec{p}}\big[g,[r_w \mid w \in \Sigma^k_0],[\vec{s_w} \mid w \in \Sigma^k_0],[\vec{t_w} \mid w \in \Sigma^k_0]\big] \in \mathcal{F}^{k+k',l}\ ,$$

or more briefly $\mathsf{SNRN}_{\vec{p}}[g,r_w,\vec{s_w},\vec{t_w}(w \in \Sigma^k_0)]$. This symbol denotes the function defined by SNRN according to the following set of rules (here the second rule is present for all $w \in \Sigma^k_0$).

$$\begin{aligned}\mathsf{f}(\vec{\mathsf{0}},\vec{x};\vec{y}) &\to g(\vec{x};\vec{y})\\ \mathsf{f}(\mathsf{S}_w(;\vec{z}),\vec{x};\vec{y}) &\to r_w(\vec{v_1},\vec{x};\vec{y},\mathsf{f}(\vec{v_1},\vec{x};\vec{s}_w(\vec{v_2},\vec{x};\vec{y},\mathsf{f}(\vec{v_2},\vec{x};\vec{t}_w(\vec{v_3},\vec{x};\vec{y},\mathsf{f}(\vec{v_3},\vec{x};\vec{y}))))))\end{aligned}$$

Here, $\vec{x}$, $\vec{y}$ and $\vec{z}$ are distinct variables and $\vec{v_i} = J^k_{p_i}(\mathsf{S}_w(\vec{;z}))$ $(i \in \{1,2,3\})$ are the predecessors of normal arguments as given by p_i.

The system $\mathcal{R}_{\mathcal{N}}$ consists of all the rules mentioned in Definition 6.4, Definition 6.5 and Definition 6.8. It is not difficult to see that for each function $f \in \mathcal{N}$, there is a *finite* restriction $\mathcal{R}(f) \subseteq \mathcal{R}_{\mathcal{N}}$ that computes the function f, c.f. [15]. Hence to prove our completeness theorem, it suffices to orient each finite restriction of $\mathcal{R}_{\mathcal{N}}$ by an instance of EPO*.

In the proof below, we use the following auxiliary function $\mathsf{h}\colon \mathcal{F} \to \mathbb{N}$ that computes the height of the definition tree of functions in $\mathcal{N}$. For $\vec{f} = f_1,\dots,f_n$, we write $\max\{\mathsf{h}(\vec{f})\}$ instead of $\max\{\mathsf{h}(f_1),\dots,\mathsf{h}(f_n)\}$.

$$\mathsf{h}(f) := \begin{cases} 0 & \text{if } f \in \{\mathsf{O}^{k,l},\mathsf{I}^{k,l}_r,\mathsf{P},\mathsf{C},\mathsf{S}_1,\mathsf{S}_2,\mathsf{0}\},\\ 1+\max\{\mathsf{h}(g),\max\{\mathsf{h}(\vec{h})\}\} & \text{if } f = \mathsf{SUB}[g,i_1,\dots,i_m,\vec{h}],\\ 1+\max\big\{\mathsf{h}(g), & \text{if } f = \mathsf{SNRN}_{\vec{p}}[g,r_w,\vec{s_w},\vec{t_w}(w \in \Sigma^k_0)].\\ \quad \max\{\mathsf{h}(r_w),\max\{\mathsf{h}(\vec{s_w})\},\max\{\mathsf{h}(\vec{t_w})\} \mid w \in \Sigma^k_0\}\big\} & \end{cases}$$

▶ **Theorem 6.9** (Completeness)**.** *Suppose $f \in FEXP$. Then there exists a confluent, constructor TRS $\mathcal{R}(f)$ computing f that is compatible with some exponential path order $>_{\mathsf{epo}\star}$.*

Proof. Consider some arbitrary function $f \in \mathsf{FEXP}$ and the corresponding TRS $\mathcal{R}(f) \subseteq \mathcal{R}_{\mathcal{N}}$ computing f. Note that $\mathcal{R}(f)$ is a non-overlapping (hence confluent) constructor TRS. Let $\mathcal{F}(f)$ be the (finite) signature consisting of function symbols appearing in $\mathcal{R}(f)$.

For function symbols $g,h \in \mathcal{F}(f)$, we define the (admissible) precedence $>$ by setting $g > h$ if and only if $\mathsf{h}(g) > \mathsf{h}(h)$. Furthermore, define the safe mapping safe as indicated by the system $\mathcal{R}_{\mathcal{N}}$. We verify $\mathcal{R}(f) \subseteq >_{\mathsf{epo}\star}$ for $>_{\mathsf{epo}\star}$ induced by the precedence $>$ and the mapping safe. For brevity, we consider the most interesting case, the rules representing the schema SNRN, cf. Definition 6.8.

Abbreviate $\mathsf{SNRN}_{\vec{p}}[g, r_w, \vec{s_w}, \vec{t_w}(w \in \Sigma_0^k)]$ as f and fix some $w \in \Sigma_0^k$. Using Definition 3.2.2, employing $\mathsf{f} > g$, it is easy to check that $\mathsf{f}(\vec{0}, \vec{x}; \vec{y}) >_{\mathsf{epo}\star} g(\vec{x}; \vec{y})$ holds. We show

$$\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) >_{\mathsf{epo}\star} r_w(\vec{v_1}, \vec{x}; \vec{y}, \mathsf{f}(\vec{v_1}, \vec{x}; \vec{s_w}(\vec{v_2}, \vec{x}; \vec{y}, \mathsf{f}(\vec{v_2}, \vec{x}; \vec{t_w}(\vec{v_3}, \vec{x}; \vec{y}, \mathsf{f}(\vec{v_3}, \vec{x}; \vec{y})))))) \quad (8)$$

First, consider the recursion-parameters $\vec{v_i} = J^k_{p_i}(\mathsf{S}_w(;\vec{z}))$ with $i \in \{1,2,3\}$. Let $\mathsf{S}_w(;\vec{z}) = \mathsf{S}_{w_1}(;z_1), \dots, \mathsf{S}_{w_k}(;z_k)$ and let $\vec{v_i} = v_{i,1}, \dots, v_{i,k}$. According to Definition 6.7 we have $\mathsf{S}_w(;\vec{z}) >^k_{\mathsf{lex}'} \vec{v_i}$. That is, there exists index $m \in \{1, \dots, k\}$ such that $\mathsf{S}_{w_j}(;z_j) = v_{i,j}$ for $j \in \{1, \dots, m-1\}$, $\mathsf{S}_{w_m}(;z_m)$ is a binary successor of $v_{i,m}$, and for $j \in \{m+1, \dots, k\}$ there exists some $n \in \{1, \dots, k\}$ with $\mathsf{S}_{w_n}(;z_n)$ equal to or a binary successor of $v_{i,j}$. This immediately gives

1) $\mathsf{S}_{w_j}(;z_j) \gtrsim v_{i,j}$ for $j \in \{1, \dots, m-1\}$,
2) $\mathsf{S}_{w_m}(;z_1) \sqsupset_{\mathsf{epo}\star} v_{i,m}$, and
3) $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) \sqsupset_{\mathsf{epo}\star} v_{i,j}$ for all $j \in \{m+1, \dots, k\}$.

Clearly $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) \sqsupset_{\mathsf{epo}\star} x_j$ for all $x_j \in \vec{x}$ by Definition 3.1, and similar Definition 3.2.1 gives $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) >_{\mathsf{epo}\star} y_j$ for $y_j \in \vec{y}$. Using (1) – (3) with respect to $i = 3$, we conclude $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) >_{\mathsf{epo}\star} \mathsf{f}(\vec{v_3}, \vec{x}; \vec{y})$ through an application of Definition 3.2.3.

Consider an arbitrary function symbol $t_{w,j} \in \vec{t_w}$. By definition, $\mathsf{f} > t_{w,j}$ in the precedence. Note that the above observations (1) – (3) imply $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) \sqsupset_{\mathsf{epo}\star} v_{3,j}$ for all $v_{3,j} \in \vec{v_3}$. Further, using $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) >_{\mathsf{epo}\star} y_j$ (for all $y_j \in \vec{y}$) and the above established inequality $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) >_{\mathsf{epo}\star} \mathsf{f}(\vec{v_3}, \vec{x}; \vec{y})$ we see that $\mathsf{f}(\mathsf{S}_w(;\vec{z}), \vec{x}; \vec{y}) >_{\mathsf{epo}\star} \vec{t}_{w,j}(\vec{v_3}, \vec{x}; \vec{y}, \mathsf{f}(\vec{v_3}, \vec{x}; \vec{y}))$ follows by Definition 3.2.2.

By instantiating observations (1) – (3) with $i = 1, 2$, and repeated application of Definition 3.2.3 and Definition 3.2.2 exactly as above, it is tedious but straight forward to prove (8). ◀

7 Conclusion

In this paper we present the *exponential path order* EPO*. Suppose a term rewrite system $\mathcal{R}$ is compatible with EPO*, then the runtime complexity of $\mathcal{R}$ is bounded from above by an exponential function. Further, EPO* is sound and complete for the class of functions computable in exponential time on a Turing machine. We have implemented EPO* in the complexity tool TCT.[1] TCT can automatically prove exponential runtime complexity of our motivating example $\mathcal{R}_{\mathsf{fib}}$. Due to Theorem 6.2 we thus obtain through an automatic analysis that the computation of the Fibonacci number is exponential.

Based on our characterisation of the class of exponential time function through the order EPO*, it is a natural question whether this approach easily generalises to any super-exponential function 2_k, for $k \in \mathbb{N}$. We studied the possiblity for such generalisations, for instance to the class of double-exponential time functions to some extent. We soon realised that any sound generalisation of EPO* to this class quickly becomes technically much more involved, if possible at all.

References

1 T. Arai and N. Eguchi. A New Function Algebra of EXPTIME Functions by Safe Nested Recursion. *TOCL*, 10(4):24:1–24:19, 2009.

[1] See `http://cl-informatik.uibk.ac.at/software/tct/`, the experimental data for our implementation is available here: `http://cl-informatik.uibk.ac.at/software/tct/experiments/epostar`.

2 T. Arai and G. Moser. Proofs of Termination of Rewrite Systems for Polytime Functions. In *Proc. of 15th FSTTCS*, volume 3821 of *LNCS*, pages 529–540, 2005.
3 M. Avanzini, N. Eguchi, and G. Moser. A Path Order for Rewrite Systems that Compute Exponential Time Functions. *CoRR*, cs/CC/1010.1128, 2010. Available at `http://www.arxiv.org/`.
4 M. Avanzini and G. Moser. Complexity Analysis by Rewriting. In *Proc. of 9th FLOPS*, volume 4989 of *LNCS*, pages 130–146, 2008.
5 M. Avanzini and G. Moser. Closing the Gap Between Runtime Complexity and Polytime Computability. In *Proc. of 21st RTA*, volume 6 of *LIPIcs*, pages 33–48, 2010.
6 F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
7 P. Baillot, J.-Y. Marion, and S. R. D. Rocca. Guest editorial: Special Issue on Implicit Computational Complexity. *TOCL*, 10(4), 2009.
8 S. Bellantoni and S. Cook. A new Recursion-Theoretic Characterization of the Polytime Functions. *CC*, 2(2):97–110, 1992.
9 A. Ben-Amram, N. D. Jones, and L. Kristiansen. Linear, Polynomial or Exponential? Complexity Inference in Polynomial Time. In *Proc. of 4th CiE*, volume 5028 of *LNCS*, pages 67–76, 2008.
10 G. Bonfante, A. Cichon, J.-Y. Marion, and H. Touzet. Algorithms with Polynomial Interpretation Termination Proof. *JFP*, 11(1):33–53, 2001.
11 G. Bonfante, J.-Y. Marion, and J.-Y. Moyen. Quasi-interpretations: A Way to Control Resources. *TCS*, 2009. To appear.
12 C. Choppy, S. Kaplan, and M. Soria. Complexity Analysis of Term-Rewriting Systems. *TCS*, 67(2–3):261–282, 1989.
13 A. Cichon and P. Lescanne. Polynomial Interpretations and the Complexity of Algorithms. In *Proc. of 11th CADE*, volume 607 of *LNCS*, pages 139–147, 1992.
14 U. Dal Lago and S. Martini. On Constructor Rewrite Systems and the Lambda-Calculus. In *Proc. of 36th ICALP*, volume 5556 of *LNCS*, pages 163–174, 2009.
15 N. Eguchi. A Lexicographic Path Order with Slow Growing Derivation Bounds. *MLQ*, 55(2):212–224, 2009.
16 J. Endrullis, J. Waldmann, and H. Zantema. Matrix Interpretations for Proving Termination of Term Rewriting. *JAR*, 40(3):195–220, 2008.
17 M. F. Ferreira. *Termination of Term Rewriting. Well-foundedness, Totality and Transformations*. PhD thesis, University of Utrecht, Faculty for Computer Science, 1995.
18 N. Hirokawa and G. Moser. Automated Complexity Analysis Based on the Dependency Pair Method. In *Proc. of 4th IJCAR*, volume 5195 of *LNAI*, pages 364–380, 2008.
19 D. Hofbauer. *Termination Proofs and Derivation Lengths in Term Rewriting Systems*. PhD thesis, Technische Universität Berlin, Fachbereich 20, Informatik, 1992. Bericht 92–46.
20 D. Hofbauer and C. Lautemann. Termination Proofs and the Length of Derivations. In *Proc. of 3rd RTA*, volume 355 of *LNCS*, pages 167–177, 1989.
21 D. C. Kozen. *Theory of Computation*. Springer Verlag, first edition, 2006.
22 D. Leivant. Stratified functional programs and computational complexity. In *Proc. of 20th POPL*, pages 325–333, 1993.
23 J.-Y. Marion. Analysing the implicit complexity of programs. *IC*, 183:2–18, 2003.
24 G. Moser. Proof Theory at Work: Complexity Analysis of Term Rewrite Systems. *CoRR*, abs/0907.5527, 2009. Habilitation Thesis.
25 D. Plump. Essentials of Term Graph Rewriting. *ENTCS*, 51:277–289, 2001.
26 TeReSe. *Term Rewriting Systems*, volume 55 of *Cambridge Tracks in Theoretical Computer Science*. Cambridge University Press, 2003.