Polynomial Path Orders and the Rules of Predicative Recursion with Parameter Substitution*

Martin Avanzini
Institute of Computer Science
University of Innsbruck, Austria
martin.avanzini@uibk.ac.at

Georg Moser
Institute of Computer Science
University of Innsbruck, Austria
georg.moser@uibk.ac.at

Abstract

In earlier work we introduced a restriction of the multiset path order, called *polynomial path order*, that induces polynomial runtime complexity. In this note, we present an extension that accounts for *predicative recursion with parameter substitution*. As confirmed by our implementation, the analytical power of polynomial path orders is significantly increased.

1 Introduction

Bellantoni and Cook [6] characterise the polytime computable functions as the least class of functions containing certain initial functions and which is closed under the schemes of *predicative recursion* and *composition*. Unlike the classical recursion-theoretic characterisation given by Cobham [7], this alternative characterisation does not rely on any externally imposed resource bounds. Instead, to break the strength of primitive recursion, the predicative schemes make use of a syntactic separation of arguments into *safe* and *normal* ones. To highlight this separation, we write $f(\vec{x};\vec{y})$ instead of $f(\vec{x},\vec{y})$ for normal arguments $\vec{x}$ and safe arguments $\vec{y}$. For previously defined functions g, h_1 and h_2, a new function f is defined by predicative recursion (on notation) via the equations

$$\begin{aligned} f(0,\vec{x};\vec{y}) &= g(\vec{x};\vec{y}) \\ f(2z+i,\vec{x};\vec{y}) &= h_i(z,\vec{x};\vec{y},f(z,\vec{x};\vec{y})),\ i\in\{1,2\}\ . \end{aligned} \tag{1}$$

For previously defined functions h, $\vec{r}$ and $\vec{s}$, a function f is defined by predicative composition by

$$f(\vec{x};\vec{y}) = h(\vec{r}(\vec{x};);\vec{s}(\vec{x};\vec{y}))\ . \tag{2}$$

Note that recursion is performed on a normal argument, whereas the recursively computed result $f(z,\vec{x};\vec{y})$ is substituted into a safe argument position of the stepping function h_i. The composition scheme guarantees that safe arguments cannot influence normal ones. Effectively, recursion on recursively computed results is prevented. The scheme of *predicative recursion with parameter substitution* generalises the scheme of predicative recursion depicted in (1). Here, the safe arguments in the recursive call may be altered additionally. Formally, a new function f is defined by predicative recursion with parameter substitution via the equations

$$\begin{aligned} f(0,\vec{x};\vec{y}) &= g(\vec{x};\vec{y}) \\ f(2z+i,\vec{x};\vec{y}) &= h_i(z,\vec{x};\vec{y},f(z,\vec{x};p_1(z,\vec{x};\vec{y}),\dots,p_m(z,\vec{x};\vec{y}))),\ i\in\{1,2\} \end{aligned} \tag{3}$$

for previously defined functions g,h_1,h_2 and $p_1,\dots,p_m$. As Bellantoni has shown in his thesis [5], the polytime-computable functions are closed under predicative recursion with parameter substitution.

Based on the schemes (1) and (2), we present in [2] a restriction of multiset path orders (MPOs), called *polynomial path orders* (POP* for short), that induce *polynomial innermost runtime complexities*. More precisely, for a TRS R define the *innermost runtime complexity function* $\mathrm{rc}^{\mathsf{i}}_{\mathsf{R}}$ as the function that relates the maximal length of an innermost R-derivation to the size of the initial term, whenever the set of initial terms is restricted to *basic* terms. Here a term is called basic, if all subterms are *values*

*This research is partially supported by FWF (Austrian Science Fund) projects P20133.

formed from constructors. Compatibility of a TRS R with some polynomial path order certifies that $\mathsf{rc}^{\mathsf{i}}_{\mathsf{R}}$ is bounded polynomially. Noteworthy, it can be shown that the polytime computable functions exactly correspond to those functions expressed by (syntactically restricted, cf. Theorem 2) TRSs compatible with polynomial path orders.

However, although the order is complete in the above sense, its application is limited to TRSs where recursion follows the specific form of (1). In particular, functions defined by *tail recursion* are excluded. Consider the TRS $\mathsf{R}_{\mathsf{rev}}$ defining the reversal of lists in a tail recursive fashion:

$$\mathsf{rev}(xs) \to \mathsf{rev}_{\mathsf{tl}}(xs, []) \qquad \mathsf{rev}_{\mathsf{tl}}([], ys) \to ys \qquad \mathsf{rev}_{\mathsf{tl}}(x : xs, ys) \to \mathsf{rev}_{\mathsf{tl}}(xs, x : ys)\ .$$

Even MPOs fail on the above TRS due to the definition of $\mathsf{rev}_{\mathsf{tl}}$, thus any application of POP* is doomed to fail as well. On the other hand, notice that the definition of $\mathsf{rev}_{\mathsf{tl}}$ is strongly reminiscent of predicative recursion with parameter substitution (3).

In this note we introduce an extension of POP* that goes beyond MPO and captures the schema (3). On a conceptual level, this is related to an alternative characterisation of the primitive recursive functions given in [12], where a related extension of MPO is employed. The resulting order, dubbed *polynomial path order with parameter substitution* or $\mathrm{POP}^*_{\mathrm{PS}}$ for short, is introduced in the next section.

2 The Polynomial Path Order with Parameter Substitution

Throughout the following, we follow the notions of [2, 4]. We fix a finite but else arbitrary signature F, partitioned into defined symbols D and constructors C. We use $\succsim = \succ \uplus \approx$ to denote an *admissible* quasi-precedence, i.e. a precedence where constructors are minimal. The separation of safe and normal argument positions is taken into account by the notion of *safe mapping*. A safe mapping safe is a function that associates with every n-ary function symbol f the set of *safe argument positions*. For constructors $f \in \mathsf{C}$ we set all argument positions safe. The argument positions not included in $\mathsf{safe}(f)$ are called *normal* and denoted by $\mathsf{nrm}(f)$. We use $\approx_{\mathsf{s}}$ to denote term equivalence as induced by $\succsim$. (Moreover, we assume $\approx_{\mathsf{s}}$ respects the separation of argument positions, cf. [4].)

The *polynomial path order* $>_{\mathsf{pps*}}$ *with parameter substitution* is based on an auxiliary order $>_{\mathsf{pps}}$ inductively defined as follows: $s = f(s_1, \dots, s_n) >_{\mathsf{pps}} t$ if either

(i) $s_i \gtrsim_{\mathsf{pps}} t$ for some $i \in \{1, \dots, n\}$, and if $f \in \mathsf{D}$ then $i \in \mathsf{nrm}(f)$, or

(ii) $t = g(t_1, \dots, t_m)$ with $f \succ g$, $f \in \mathsf{D}$ and $s >_{\mathsf{pps}} t_j$ for all $j \in \{1, \dots, m\}$.

Here $\gtrsim_{\mathsf{pps}} := >_{\mathsf{pps}} \cup \approx_{\mathsf{s}}$. The split into two orders is necessary, as we must enforce the special shape of predicative composition (2) in the definition of $>_{\mathsf{pps*}}$ below. (Note that due to the restrictive definition of case (i), one can show $f(\vec{x};\vec{y}) >_{\mathsf{pps}} r_i(\vec{x};)$, but one cannot show $f(\vec{x};\vec{y}) >_{\mathsf{pps}} s_i(\vec{x};\vec{y})$.)

Based on $>_{\mathsf{pps}}$, the polynomial path order $>_{\mathsf{pps*}}$ with parameter substitution is defined as follows. Here we write $\mathsf{F}^{\prec f}$ for the restriction of F to symbols ranked below f in the precedence, i.e., $\mathsf{F}^{\prec f} := \{g \mid f \succ g \wedge f \in \mathsf{F}\}$. Further, we write $\{\!\{a_1, \dots, a_n\}\!\}$ for the multiset with elements $a_1, \dots, a_n$. We set $\gtrsim_{\mathsf{pps*}} := >_{\mathsf{pps*}} \cup \approx_{\mathsf{s}}$ and refer with $>^{\mathsf{mul}}_{\mathsf{pps*}}$ to the strict order contained in the multiset extension of $\gtrsim_{\mathsf{pps*}}$. We inductively define $>_{\mathsf{pps*}}$ such that: $s = f(s_1, \dots, s_n) >_{\mathsf{pps*}} t$ if

(i) $s_i \gtrsim_{\mathsf{pps*}} t$ for some $i \in \{1, \dots, n\}$, or

(ii) $t = g(t_1, \dots, t_m)$ with $f \succ g$, $f \in \mathsf{D}$, and

  (a) $s >_{\mathsf{pps}} t_j$ for all $j \in \mathsf{nrm}(g)$ and $s >_{\mathsf{pps*}} t_j$ for all $j \in \mathsf{safe}(g)$, and

  (b) there exists $j_0 \in \mathsf{safe}(g)$ such that $t_j \in \mathsf{T}(\mathsf{F}^{\prec f}, \mathsf{V})$ for all $j \neq j_0$, or

(iii) $t = g(t_1,\dots,t_m)$ with $f \approx g$ and

(a) $\{\{s_{i_1},\dots,s_{i_p}\}\} >^{\mathsf{mul}}_{\mathsf{pps*}} \{\{t_{j_1},\dots,t_{j_q}\}\}$ where $\mathsf{nrm}(f) = \{i_1,\dots,i_p\}$ and $\mathsf{nrm}(g) = \{j_1,\dots,j_q\}$, and

(b) both $s >_{\mathsf{pps*}} t_j$ and $t_j \in \mathsf{T}(\mathsf{F}^{\prec f},\mathsf{V})$ for all $j \in \mathsf{safe}(g)$.

Case (ii) basically accounts for predicative composition. The condition (ii.b) is used to guarantee that at most one recursively computed results is substituted into a safe argument position j_o. Case (iii) reflects the scheme of predicative recursion with parameter substitution (3). The additional condition (iii.b) is essential, it prohibits for instance the orientation of $\mathsf{f}(\mathsf{s}(;x)\,;y) \to \mathsf{f}(x;\mathsf{f}(x;y))$ that gives rise to exponentially long (innermost) derivations.

Opposed to *recursive path orders* like MPO, $>_{\mathsf{pps*}}$ is neither closed under contexts nor closed under substitutions. Still, following the pattern of the proof given in [2], we derive our main theorem.

Theorem 1. *Let* R *be a finite constructor TRS. If* R *is compatible with* $>_{\mathsf{pps*}}$*, i.e.,* $\mathsf{R} \subseteq >_{\mathsf{pps*}}$*, then the innermost runtime complexity* $\mathrm{rc}^{\mathsf{i}}_{\mathsf{R}}$ *induced is polynomially bounded.*

Notice that $>_{\mathsf{pps*}}$ is applicable to the TRS $\mathsf{R}_{\mathsf{rev}}$ from above: define $\mathsf{safe}(\mathsf{rev}) = \varnothing$ and $\mathsf{safe}(\mathsf{rev_{tl}}) = \{2\}$. Moreover, set $\mathsf{rev} \succ \mathsf{rev_{tl}} \succ (:) \succ []$ in the precedence. Compatibility $\mathsf{R}_{\mathsf{rev}} \subseteq >_{\mathsf{pps*}}$ is now straight forward to verify. By Theorem 1, we conclude that the innermost runtime complexity of $\mathsf{R}_{\mathsf{rev}}$ is polynomially bounded.

We stress that every TRS compatible with some instance of POP* is also compatible with some instance of $\mathrm{POP}^*_{\mathsf{PS}}$. As demonstrated by the TRS $\mathsf{R}_{\mathsf{rev}}$, the reverse direction does not hold. Consequently, Theorem 1 is strictly more powerful than the main theorem given in [2].

In [2] we have shown that POP* gives rise to an alternative characterisation of the polytime computable functions. The same observation carries over to $\mathrm{POP}^*_{\mathsf{PS}}$. The next theorem establishes that $\mathrm{POP}^*_{\mathsf{PS}}$ induces polytime computability of the function described through the analysed TRS. The proof of the theorem follows exactly the proof of the corresponding theorem given in [3].

Theorem 2. *Suppose* R *is a finite, orthogonal and sorted constructor TRS based on a simple signature. If* R *is compatible with* $>_{\mathsf{pps*}}$ *then the functions computed by* R *are polytime computable and vice verse, each polytime computable function is computable by such a TRS that is compatible with* $>_{\mathsf{pps*}}$.

Here *simple* signature [11] essentially means that the size of any constructor term depends only polynomially on its depth. The restriction is responsible for the introduction of sorts, compare [2, 11]. Simple signatures allow the definition of enumerated and inductive datatypes like lists and words, but prohibit for instance the definition of tree structures.

3 Experimental Results

We have implemented the here described technique in the experimental version of the *Tyrolean Complexity Tool* TCT, an open source complexity analyser[1] All experiments were conducted on a machine that is identical to the official competition server (8 AMD Opteron® 885 dual-core processors with 2.8GHz, 8x8 GB memory). As timeout we use 5 seconds. Experiments were performed on a subset of the 1394 examples from the Termination Problem Database Version 5.0.2 that were used in the runtime-complexity category of the termination competition 2008[2], where we filtered out all non-constructor TRSs. The restricted testbed consists of 638 TRSs.

[1]The stable version is available at `http://cl-informatik.uibk.ac.at/software/tct`. A preliminary build and sources of the experimental version can be found at `http://cl-informatik.uibk.ac.at/~zini/wst09`.

[2]See `http://termcomp.uibk.ac.at`.

Table 1 compares the application of $\mathrm{POP}^*_{\mathrm{PS}}$ to the application of POP^* from [2], where we highlight the total on yes-, maybe- and timeout-instances. Furthermore, we annotate average times in seconds.[3] To check compatibility we encode the constraints on precedence and so forth in *propositional logic* (cf. [1] for details), employing MiniSat [8] for finding satisfying assignments. Table 1 reflects that the here proposed extension significantly increases the analytical power of polynomial path orders.

Table 1: Experimental Results

	Yes	Maybe	Timeout
POP^*	40 / 0.03	598 / 0.05	0 / 0.00
$\mathrm{POP}^*_{\mathrm{PS}}$	51 / 0.05	585 / 0.14	2 / 5.02

4 Conclusion

In this paper we study the runtime complexity of rewrite systems. We extend polynomial path orders with the scheme of predicative recursion with parameter substitution, resulting in a strictly more powerful technique. For constructor TRSs compatible with some instance of $\mathrm{POP}^*_{\mathrm{PS}}$, we conclude a *polynomial* bound on the *innermost runtime complexity* of the studied term rewrite system. Moreover, we obtain an alternative characterisation of the *polytime computable functions*. We have implemented the technique and experimental evidence clearly indicates the power and in particular the efficiency of the new method. Although not presented, following [4] we can lift the restriction that the TRS under consideration is a constructor TRS. Also worthy of note, the here described technique allows an integration into the *dependency pair framework for complexity analysis* as put forward in [9, 10], compare [4].

5 Appendix

Below we present the missing proofs of Theorem **??** and Theorem **??** respectively.

Following [**?**, Section 6.5], we briefly recall *typed rewriting*. Let S be a finite set representing the set of *types* or *sorts*. An *S-sorted set A* is a family of sets $\{A_s \mid s \in S\}$ such that all sets A_s are pairwise disjoint. In the following, we suppose that V denotes an S-sorted set of variables. An *S-sorted signature* F is like a signature, but the *arity* of $f \in \mathsf{F}$ is defined by $\mathsf{ar}(f) = (s_1,\ldots,s_n)$ for $s_1,\ldots,s_n \in S$. Additionally, each symbol $f \in \mathsf{F}$ is associated with a sort $s \in S$, called the *type of f* and denoted by $\mathsf{st}(f)$. We adopt the usual notion and write $f\colon (s_1,\ldots,s_n) \to s$ when $\mathsf{ar}(f) = (s_1,\ldots,s_n)$ and $\mathsf{st}(f) = s$. The *S-sorted set of terms* $\mathsf{T}(\mathsf{F},\mathsf{V})_S$ consists of the sets $\mathsf{T}(\mathsf{F},\mathsf{V})_s$ for $s \in S$, where $\mathsf{T}(\mathsf{F},\mathsf{V})_s$ is inductively defined by (i) $\mathsf{V}_s \subseteq \mathsf{T}(\mathsf{F},\mathsf{V})_s$, and (ii) $f(t_1,\ldots,t_n) \in \mathsf{T}(\mathsf{F},\mathsf{V})_s$ for all function symbols $f \in \mathsf{F}$, $f\colon (s_1,\ldots,s_n) \to s$ and terms $t_i \in \mathsf{T}(\mathsf{F},\mathsf{V})_{s_i}$ for $i \in \{1,\ldots,n\}$. We say that a term t is *well-typed* if $t \in \mathsf{T}(\mathsf{F},\mathsf{V})_s$ for some sort s. An S-sorted term rewrite system R is a TRS such that for $l \to r \in \mathsf{R}$, it holds that $l, r \in \mathsf{T}(\mathsf{F},\mathsf{V})_s$ for some sort $s \in S$. As a consequence, for $s \in \mathsf{T}(\mathsf{F},\mathsf{V})_s$ and $s \to_{\mathsf{R}} t$, we have that $t \in \mathsf{T}(\mathsf{F},\mathsf{V})_s$.

Example 1. *Let $S = \{\mathsf{Bool},\mathsf{List},\mathsf{Nat},\mathsf{Pair}\}$. The S-sorted rewrite system $\mathsf{R}_{\mathsf{Lst}}$ is given by the following rules:*

$$\begin{aligned} \mathsf{f}(\mathsf{s}(x)) &\to \mathsf{cons}(\mathsf{pair}(x,\mathsf{g}(x)),\mathsf{f}(x)) & \mathsf{g}(\mathsf{s}(x)) &\to \mathsf{g}(x)\\ \mathsf{f}(0) &\to \mathsf{nil} & \mathsf{g}(0) &\to \mathsf{tt} \end{aligned}$$

Here we assign arities and sorts as follows: for the constructors we set $0\colon \mathsf{Nat}$, $\mathsf{s}\colon \mathsf{Nat} \to \mathsf{Nat}$, $\mathsf{pair}\colon (\mathsf{Nat},\mathsf{Bool}) \to \mathsf{Pair}$, $\mathsf{tt}\colon \mathsf{Bool}$, $\mathsf{nil}\colon \mathsf{List}$, $\mathsf{cons}\colon (\mathsf{Pair},\mathsf{List}) \to \mathsf{List}$*; for the defined symbols we set* $\mathsf{f}\colon \mathsf{Nat} \to \mathsf{List}$ *and* $\mathsf{g}\colon \mathsf{Nat} \to \mathsf{Bool}$.

[3] See `http://cl-informatik.uibk.ac.at/~zini/wst09` for extended results.

A *simple* signature [11] is a sorted signature such that each sort has a finite *rank* r in the following sense: the sort s has rank r if for every constructor $c : (s_1,\dots,s_n) \to s$, the rank of each sort s_i is less than the rank of s, except for at most one sort which can be of rank r. Simple signatures allow the definition of enumerated datatypes and inductive datatypes like words and lists but prohibit for instance the definition of tree structures. Observe that the signature underlying $\mathsf{R}_{\mathsf{Lst}}$ from Example 1 is simple. A crucial insight is that sizes of values formed from a simple signature can be estimated polynomially in their depth. The easy proof of the following proposition can be found in [11, Proposition 17].

Proposition 1. *Let* C *be a set of constructors from a simple signature* F. *There exists a constant* $d \in \mathbb{N}$ *such that for each term* $t \in \mathsf{T}(\mathsf{C},\mathsf{V})_S$ *whose rank is* r, $|t| \leqslant d^r \cdot \mathsf{dp}(t)^{r+1}$.

In order to give a polytime algorithm for the functions computed by a TRS, it is essential that sizes of reducts do not exceed a polynomial bound with respect to the size of the start term. Recall that approximations $\blacktriangleright_k$ tightly control the size growth of terms. For simple signatures, we can exploit this property for a space-complexity analysis. Although predicative interpretations remove values, by the above proposition sizes of those can be estimated based on the Buchholz-norm record in N. And so we derive the following Lemma, essential for the proof of Theorem **??**.

Lemma 1. *Let* F *be a simple signature. There exists a (monotone) polynomial* p *depending only on* F *such that for each well-typed term* $t \in \mathsf{T}(\mathsf{F},\mathsf{V})_s$, $|t| \leqslant p(\mathsf{G}_k(\mathsf{N}^{\mathsf{s}}(t)))$.

Proof. The Lemma follows as: (i) for all sequences $s \in \mathsf{Seq}$, $|s| \leqslant \mathsf{G}_k(s)+1$, and (ii) for all terms $t \in \mathsf{T}(\mathsf{F},\mathsf{V})_s$, $|t| \leqslant c \cdot |\mathsf{N}^{\mathsf{s}}(t)|^d$ for some uniform constants $0 < c, d \in \mathbb{N}$. These properties are simple to verify: property (i) follows from induction on s where we employ for the inductive step that $f(s_1,\dots,s_n) \blacktriangleright_k [s_1 \cdots s_n]$ and $\mathsf{G}_k([s_1 \cdots s_n]) = \sum_{i=1}^{n} \mathsf{G}_k(s_i) + n$. For property (ii), set $d = r+2$ where r is the maximal rank of a symbol in C, and set $c = e^r$ where e is as given from Proposition 1. First one shows by a straight forward induction on t that $|t| \leqslant c \cdot (|\mathsf{S}(t)| \cdot \|t\|^{r+1})$ (employing Proposition 1 and $\mathsf{dp}(t) \leqslant \|t\|$). As $|\mathsf{S}(t)| < |\mathsf{N}(t)|$ and $\|t\| < |\mathsf{N}(t)|$, we derive $|t| < c \cdot |\mathsf{N}(t)|^d$. By induction on the definition of N^{s} we finally obtain property (ii). □ □

Let R be a (not necessarily S-sorted) TRS that is innermost terminating. In the sequel, we keep R fixed. In order to exploit Lemma 1 for an analysis by means of weak innermost dependency pairs, we introduce the notion of *type preserving weak innermost dependency pairs*.

Definition 3. *If* $l \to r \in \mathsf{R}$ *and* $r = C\langle u_1,\dots,u_n\rangle_{\mathsf{D}}$ *then* $l^\sharp \to \mathsf{c}(u_1^\sharp,\dots,u_n^\sharp)$ *is called a* type preserving weak innermost dependency pair *of* R. *Here, the* compound symbol c *is supposed to be fresh. We set* $\mathsf{repr}(\mathsf{c}) := C$ *and say that* c represents *the context* C. *The set of all type preserving weak innermost dependency pairs is denoted by* $\mathsf{WIDP}(\mathsf{R})$.

We collect all compound symbols appearing in $\mathsf{TPWIDP}(\mathsf{R})$ in the set C_{com}.

Example 2 (Example 1 continued)**.** *Reconsider the rewrite system* $\mathsf{R}_{\mathsf{Lst}}$ *given in Example 1. The set* $\mathsf{TPWIDP}(\mathsf{R}_{\mathsf{Lst}})$ *is given by*

$$\mathsf{f}^\sharp(\mathsf{s}(x)) \to \mathsf{c}_1(\mathsf{g}^\sharp(x),\mathsf{f}^\sharp(x)) \qquad \mathsf{g}^\sharp(\mathsf{s}(x)) \to \mathsf{c}_3(\mathsf{g}^\sharp(x))$$
$$\mathsf{f}^\sharp(0) \to \mathsf{c}_2 \qquad \mathsf{g}^\sharp(0) \to \mathsf{c}_4$$

The constant c_3 *represents for instance the empty context, and the constant* c_1 *represents the context* $\mathsf{repr}(\mathsf{c}_1) = \mathsf{cons}(\mathsf{pair}(x,\Box),\Box)$.

Lemma 2. *Let* R *be an S-sorted TRS such that the underlying signature* F *is simple. Then* $\mathsf{TPWIDP}(\mathsf{R}) \cup \mathsf{U}(\mathsf{WIDP}(\mathsf{R}))$ *is an S-sorted TRS, and the underlying signature* $\mathsf{F}^\sharp \cup \mathsf{C}_{com}$ *a simple signature.*

Proof. To conclude the claim, it suffices to type the marked and compound symbols appropriately. For each rule $f^\sharp(l_1,\dots,l_n) \to \mathsf{c}(r_1^\sharp,\dots,r_n^\sharp) \in \mathsf{TPWIDP}(\mathsf{R})$ we proceed as follows: we set $\mathsf{ar}(f^\sharp) := \mathsf{ar}(f)$ and $\mathsf{st}(f^\sharp) := \mathsf{st}(f)$. Moreover, we set $\mathsf{ar}(\mathsf{c}) := (\mathsf{st}(r_1),\dots,\mathsf{st}(r_m))$ and $\mathsf{st}(\mathsf{c}) := \mathsf{st}(f)$. It is easy to see that since R is S-sorted, $\mathsf{TPWIDP}(\mathsf{R}) \cup \mathsf{U}(\mathsf{TPWIDP}(\mathsf{R}))$ is S-sorted too. □ □

Notice that the above lemma fails for weak innermost dependency pairs: consider the rule $\mathsf{f}(x) \to \mathsf{d}(\mathsf{g}(x))$, where f and g are defined symbols and d is a constructor. Moreover, suppose $\mathsf{f}\colon \mathsf{s}_2 \to \mathsf{s}_1$, $\mathsf{g}\colon \mathsf{s}_2 \to \mathsf{s}_3$ and $\mathsf{d}\colon \mathsf{s}_3 \to \mathsf{s}_1$. Then we cannot type the corresponding weak innermost dependency pair $\mathsf{f}^\sharp(x) \to \mathsf{g}^\sharp(x)$ as above because (return-)types of $\mathsf{f}^\sharp$ and $\mathsf{g}^\sharp$ differ.

As for practical all termination techniques, compatibility of weak innermost dependency pairs with polynomial path orders also yield compatibility of type preserving weak innermost dependency pairs. Moreover, from the definition we immediately see that $\mathrm{dl}(t^\sharp, \xrightarrow{\mathsf{i}}_{\mathsf{TPWIDP}(\mathsf{R})/\mathsf{U}}) = \mathrm{dl}(t^\sharp, \xrightarrow{\mathsf{i}}_{\mathsf{WIDP}(\mathsf{R})/\mathsf{U}})$ with $\mathsf{U} = \mathsf{U}(\mathsf{WIDP}(\mathsf{R}))$ and basic term t. And so it is clear that in order to proof Theorem **??** and Theorem **??**, $\mathsf{WIDP}(\mathsf{R})$ can safely be replaced by $\mathsf{TPWIDP}(\mathsf{R})$. We continue with the proof of Theorem **??**.

5.1 A Proof of Theorem ??

As mentioned in Section **??**, we now introduce an *extended predicative interpretation* whose purpose is to interpret compound symbols as sequences, and their arguments via the interpretation N.

Definition 4. *The* extended predicative interpretation N^{s} *from terms* $\mathsf{T}(\mathsf{F},\mathsf{V})$ *to sequences* $\mathsf{Seq}(\mathsf{F}_\pi^{\mathsf{n}} \cup \{\mathsf{s}\},\mathsf{V})$ *is defined as follows: if* $t = \mathsf{c}(t_1,\dots,t_n)$ *and* $\mathsf{c} \in \mathsf{C}_{com}$ *then* $\mathsf{N}^{\mathsf{s}}(t) := [\mathsf{N}^{\mathsf{s}}(t_1)\ \cdots\ \mathsf{N}^{\mathsf{s}}(t_n)]$*, and otherwise* $\mathsf{N}^{\mathsf{s}}(t) := [\mathsf{N}(t)]$.

Let ComCtx abbreviate the set of contexts $\mathsf{T}(\mathsf{C}_{com} \cup \{\Box\},\mathsf{V})$ build from compound symbols. Set $\mathsf{P} = \mathsf{TPWIDP}(\mathsf{R})$ and $\mathsf{U} = \mathsf{U}(\mathsf{WIDP}(\mathsf{R}))$. In order to highlight the correspondence between $\xrightarrow{\mathsf{i}}_{\mathsf{R}}$ and $\xrightarrow{\mathsf{i}}_{\mathsf{P}/\mathsf{U}}$, we extend the notion of *representatives*.

Definition 5. *Let* $C \in \mathsf{ComCtx}$*. We define* $\mathsf{reprs}(C)$ *as the least set of (ground) contexts such that (i) if* $C = \Box$ *then* $\Box \in \mathsf{reprs}(C)$*, and (ii) if* $C = \mathsf{c}(C_1,\dots,C_n)$*,* $C_i' \in \mathsf{reprs}(C_i)$ *and* σ *is a substitution from all variables in* $\mathsf{repr}(\mathsf{c})$ *to ground normal forms of* R *then* $(\mathsf{repr}(\mathsf{c})\sigma)[C_1',\dots,C_n'] \in \mathsf{reprs}(C)$.

Example 3 (Example 2 continued)**.** *Reconsider the TRS* $\mathsf{R}_{\mathsf{Lst}}$ *from Example 1, together with* $\mathsf{TPWIDP}(\mathsf{R}_{\mathsf{Lst}})$ *as given in Example 2. Consider the step* $\mathsf{f}(\mathsf{s}(0)) \to_{\mathsf{R}_{\mathsf{Lst}}} \mathsf{cons}(\mathsf{pair}(0,\mathsf{g}(0)),\mathsf{f}(0))$ *and the corresponding dependency pair step* $\mathsf{f}^\sharp(\mathsf{s}(0)) \to_{\mathsf{TPWIDP}(\mathsf{R}_{\mathsf{Lst}})} \mathsf{c}_1(\mathsf{g}^\sharp(0),\mathsf{f}^\sharp(0))$*. Let* $C = \mathsf{c}_1(\Box,\Box)$*, remember that* $\mathsf{repr}(\mathsf{c}_1) = \mathsf{cons}(\mathsf{pair}(x,\Box),\Box)$, $\mathsf{reprs}(\Box) = \Box$ *and observe that* $C' = \mathsf{cons}(\mathsf{pair}(0,\Box),\Box) \in \mathsf{reprs}(C)$ *by taking the substitution* $\sigma = \{x \mapsto 0\}$*. And hence we can reformulate the above two steps as* $\mathsf{f}(\mathsf{s}(0)) \to_{\mathsf{R}_{\mathsf{Lst}}} C'[\mathsf{g}(0),\mathsf{f}(0)]$ *and likewise* $\mathsf{f}^\sharp(\mathsf{s}(0)) \to_{\mathsf{TPWIDP}(\mathsf{R}_{\mathsf{Lst}})} C[\mathsf{g}^\sharp(0),\mathsf{f}^\sharp(0)]$.

We manifest the above observation in the following lemma.

Lemma 3. *Let* $s \in \mathsf{T}_{\mathsf{b}}$ *be a ground and basic term. Suppose* $s \xrightarrow{\mathsf{i}}{}^*_{\mathsf{R}}\, t$*. Let* $\mathsf{P} = \mathsf{TPWIDP}(\mathsf{R})$ *and let* $\mathsf{U} = \mathsf{U}(\mathsf{WIDP}(\mathsf{R}))$*. Then there exists contexts* $C' \in \mathsf{ComCtx}$*,* $C \in \mathsf{reprs}(C')$ *and terms* $t_1,\dots,t_n$ *such that* $t = C[t_1,\dots,t_n]$ *and moreover,* $s^\sharp \xrightarrow{\mathsf{i}}{}^*_{\mathsf{P}\cup\mathsf{U}}\, C'[t_1^\sharp,\dots,t_n^\sharp]$.

Proof. We proof the lemma by induction on the length of the rewrite sequence $s \xrightarrow{\mathsf{i}}{}^n_{\mathsf{R}}\, t$. The base case $n = 0$ is trivial, we set $C = C' = \Box$. So suppose $s \xrightarrow{\mathsf{i}}{}^n_{\mathsf{R}}\, t \xrightarrow{\mathsf{i}}_{\mathsf{R}} u$ and the property holds for n. And thus we can find contexts $C_t' \in \mathsf{ComCtx}$, $C_t \in \mathsf{reprs}(C_t')$ and terms $t_1,\dots,t_n$ such that $t = C_t[t_1,\dots,t_n]$ and moreover, $s^\sharp \xrightarrow{\mathsf{i}}{}^*_{\mathsf{P}\cup\mathsf{U}}\, C_t'[t_1^\sharp,\dots,t_n^\sharp]$. Without loss of generality we can assume $u = C_t[t_1,\dots,u_i,\dots,t_n]$ with $t_i \xrightarrow{\mathsf{i}}_{\mathsf{R}} u_i$, as the context C_t is solely build from constructors and normal forms of R.

First, suppose $t_i \xrightarrow{\mathsf{i}}^{\varepsilon}_{\mathsf{R}} u_i$, and hence $t_i = l\sigma$ for $l \to r \in \mathsf{R}$ and substitution $\sigma \colon \mathsf{V} \to \mathsf{NF}(\mathsf{R}) \cap \mathsf{T}(\mathsf{F})$. Moreover $l^\sharp \to \mathsf{c}(r_1^\sharp,\dots,r_m^\sharp) \in \mathsf{P}$ such that $u_i = (\mathsf{repr}(\mathsf{c})\sigma)[r_1\sigma,\dots,r_m\sigma]$. We set C' as the context obtained from replacing the i-th hole of C'_t by $\mathsf{c}(\Box,\dots,\Box)$, likewise we set C as the context obtained from replacing the i-th hole of C_t by $\mathsf{repr}(\mathsf{c})\sigma$. Notice that $C \in \mathsf{reprs}(C')$. We conclude $s^\sharp \xrightarrow{\mathsf{i}}^{*}_{\mathsf{P}\cup\mathsf{U}} C'[t_1^\sharp,\dots,r_1^\sharp\sigma,\dots,r_m^\sharp\sigma,\dots,t_n^\sharp]$ and $u = C[t_1,\dots,r_1\sigma,\dots,r_m\sigma,\dots,t_n]$ which establishes the lemma for this case.

Now suppose $t_i \xrightarrow{\mathsf{i}}_{\mathsf{R}} u_i$ is a step below the root. Thus we have also $t_i^\sharp \xrightarrow{\mathsf{i}}_{\mathsf{R}} u_i^\sharp$. As shown in [9, Lemma 16], the latter can be strengthened to $t_i^\sharp \xrightarrow{\mathsf{P}\cup\mathsf{U}}_{\mathsf{U}} u_i^\sharp$. We conclude $s^\sharp \xrightarrow{\mathsf{i}}^{*}_{\mathsf{P}\cup\mathsf{U}} C'_t[t_1^\sharp,\dots,u_i^\sharp,\dots,t_n^\sharp]$, and the lemma follows by setting $C' = C'_t$ and $C = C_t$. □ □

Suppose $\mathsf{WIDP}(\mathsf{R})$ contains non-nullary compound symbols. In order to establish an embedding in the sense of Lemma **??** for that case, by the above lemma we see that it suffices to consider only terms of shape $s = C[s_1^\sharp,\dots,s_n^\sharp]$ with $C \in \mathsf{ComCtx}$. With this insight, we adjust Lemma **??** as below. Observe that due to the definition of N^{s}, we cannot simply apply Lemma **??** together with closure under context of $\blacktriangleright_k$ here.

Lemma 4. *Let $s = C[s_1^\sharp,\dots,s_n^\sharp]$ for $C \in \mathsf{ComCtx}$ and $s_1,\dots,s_n \in \mathsf{T}(\mathsf{F},\mathsf{V})$. Let $\mathsf{P} = \mathsf{TPWIDP}(\mathsf{R})$ and $\mathsf{U} = \mathsf{U}(\mathsf{WIDP}(\mathsf{R}))$. There exists a uniform constant $k \in \mathbb{N}$ depending only on R such that if $\mathsf{P} \subseteq >^{\pi}_{\mathsf{pop}*}$ holds then $s \xrightarrow{\mathsf{v}}_{\mathsf{P}} t$ implies $\mathsf{N}^{\mathsf{s}}(s) \blacktriangleright_k \mathsf{N}^{\mathsf{s}}(t)$. Moreover, if $\mathsf{U} \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ holds then $s \xrightarrow{\mathsf{v}}_{\mathsf{U}} t$ implies $\mathsf{N}^{\mathsf{s}}(s) \mathrel{\underset{\sim}{\blacktriangleright}}_k \mathsf{N}^{\mathsf{s}}(t)$.*

Proof. We proof the lemma for $k := \max\{3 \cdot \|r\| \mid l \to r \in \mathsf{P} \cup \mathsf{U}\}$. Suppose $s \xrightarrow{\mathsf{v}}_{\mathsf{P}} t$ or $s \xrightarrow{\mathsf{v}}_{\mathsf{U}} t$ respectively, and thus $t = C[s_1^\sharp,\dots,t_i,\dots,s_n^\sharp]$ for some term t_i. There exists a context C' (over sequences) such that $\mathsf{N}^{\mathsf{s}}(s) = C'[\mathsf{N}^{\mathsf{s}}(s_i^\sharp)]$ and $\mathsf{N}^{\mathsf{s}}(t) = C'[\mathsf{N}^{\mathsf{s}}(t_i^\sharp)]$. First assume $s_i^\sharp \xrightarrow{\mathsf{v}}_{\mathsf{P}} t_i$, and thus $\mathsf{N}^{\mathsf{s}}(s_i^\sharp) = [\mathsf{N}(l^\sharp\sigma)]$ and $\mathsf{N}^{\mathsf{s}}(t_i) = [[\mathsf{N}(r_1^\sharp\sigma)],\dots,[\mathsf{N}(r_m^\sharp\sigma)]]$ for $l \to \mathsf{c}(r_1^\sharp,\dots,r_m^\sharp) \in \mathsf{P}$. To verify $\mathsf{N}^{\mathsf{s}}(s) \blacktriangleright_k \mathsf{N}^{\mathsf{s}}(t)$, by Definition **??**.**??** and Definition **??**.**??** it suffices to verify $\mathsf{N}(l^\sharp\sigma) \blacktriangleright_{k-1} \mathsf{N}(r_j^\sharp\sigma)$ for all $j \in \{1,\dots,m\}$. The latter is an easy consequence of Lemma **??**, where we employ that (i) $l^\sharp >^{\pi}_{\mathsf{pop}*} r_j^\sharp$ follows from the assumption $\mathsf{P} \subseteq >^{\pi}_{\mathsf{pop}*}$, and (ii) $\|\pi(r)\| > \|\pi(r_j)\|$. Both properties are straight forward to verify since π is safe. For $s_i^\sharp \xrightarrow{\mathsf{v}}_{\mathsf{U}} t$ we have $\mathsf{N}^{\mathsf{s}}(s_i^\sharp) = [\mathsf{N}(s_i^\sharp)]$ and $\mathsf{N}^{\mathsf{s}}(t_i^\sharp) = [\mathsf{N}(t_i^\sharp)]$ for $l \to r \in \mathsf{U}$. From Lemma **??** we obtain $\mathsf{N}(s_i^\sharp) \mathrel{\underset{\sim}{\blacktriangleright}}_k \mathsf{N}(t_i^\sharp)$ which establishes the lemma. □ □

The proof of Theorem **??** is now easily obtained by incorporating the above lemma into Theorem **??**.

Theorem. *Let R be a constructor TRS, and let P denote the set of weak innermost dependency pairs. Assume P is non-duplicating, and suppose $\mathsf{U}(\mathsf{P}) \subseteq >_{\mathsf{A}}$ for some SLI A. Let π be a safe argument filtering. If $\mathsf{P} \subseteq >^{\pi}_{\mathsf{pop}*}$ and $\mathsf{U}(\mathsf{P}) \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ then $\mathrm{rc}^{\mathsf{i}}_{\mathsf{R}}$ is polynomially bounded.*

Proof. According to Proposition **??** we need to find a polynomial p such that $\mathrm{dl}(t^\sharp, \xrightarrow{\mathsf{i}}_{\mathsf{WIDP}(\mathsf{R})/\mathsf{U}(\mathsf{WIDP}(\mathsf{R}))}) \leqslant p(|t^\sharp|)$. We set $\mathsf{P} = \mathsf{TPWIDP}(\mathsf{R})$ and likewise $\mathsf{U} = \mathsf{U}(\mathsf{WIDP}(\mathsf{R}))$. Clearly, it suffices to show $\mathrm{dl}(t^\sharp, \xrightarrow{\mathsf{i}}_{\mathsf{TPWIDP}(\mathsf{R})/\mathsf{U}(\mathsf{WIDP}(\mathsf{R}))}) \leqslant p(|t^\sharp|)$ for that. Consider a sequence

$$t^\sharp = t_0 \xrightarrow{\mathsf{i}}_{\mathsf{P}/\mathsf{U}} t_1 \xrightarrow{\mathsf{i}}_{\mathsf{P}/\mathsf{U}} \cdots \xrightarrow{\mathsf{i}}_{\mathsf{P}/\mathsf{U}} t_\ell\,,$$

and pick a relative step $t_i \xrightarrow{\mathsf{i}}_{\mathsf{P}/\mathsf{U}} t_{i+1}$. Define $\mathsf{U}' = \mathsf{U} \cup \mathsf{V}(\mathsf{P} \cup \mathsf{U})$ and $\phi(t) = \phi_{\mathsf{P}\cup\mathsf{U}}(t)$. Clearly Lemma **??** can be extended to account for steps of P below the root, and thus $\phi(t_i) \xrightarrow{\mathsf{v}}_{\mathsf{P}/\mathsf{U}'} \phi(t_{i+1})$ follows. Hence for some terms u and v, $\phi(t_i) \xrightarrow{\mathsf{v}}^{*}_{\mathsf{U}'} u \xrightarrow{\mathsf{v}}_{\mathsf{P}} v \xrightarrow{\mathsf{v}}^{*}_{\mathsf{U}'} \phi(t_{i+1})$. As shown in Lemma 3, all involved terms in the above sequence have the shape $C[s_1^\sharp,\dots,s_n^\sharp]$, $C \in \mathsf{ComCtx}$. As $\mathsf{WIDP}(\mathsf{R}) \subseteq >^{\pi}_{\mathsf{pop}*}$, and since π is safe, it is easy to infer that $\mathsf{P} \subseteq >^{\pi}_{\mathsf{pop}*}$ holds (we just set every compound symbol from P minimal in the precedence). And hence Lemma 4 translates the above relative step to $\mathsf{N}^{\mathsf{s}}(\phi(s)) \blacktriangleright_k^{+} \mathsf{N}^{\mathsf{s}}(\phi(t))$ for some uniform constant k. As a consequence, $\mathrm{dl}(t, \xrightarrow{\mathrm{i}}_{\mathsf{WIDP(R)/U(WIDP(R))}}) \leqslant \mathsf{G}_k(\mathsf{N^s}(\phi(t)))$ for all terms t. Fix some reducible and basic term $t \in \mathsf{T_b}$. Observe $\mathsf{N^s}(\phi(t^\sharp)) = [\mathsf{N}(t^\sharp)]$ and so from Lemma **??** we see that $\mathsf{G}_k(\mathsf{N^s}(\phi(t^\sharp)))$ is bounded polynomially in the size of t. The polynomial depends only on k. We conclude the theorem. ☐ ☐

5.2 A Proof of Theorem ??

We now proceed with the proof Theorem **??**, which is essentially an extension to Theorem **??**.

We first precisely state what it means that a TRS *computes* some function. For this, let $\ulcorner\cdot\urcorner\colon \Sigma^* \to \mathsf{T}(\mathsf{C})$ denote an *encoding function* that represents words over the alphabet Σ as ground values. We call an encoding $\ulcorner\cdot\urcorner$ *reasonable* if it is bijective and there exists a constant c such that $|u| \leqslant |\ulcorner u \urcorner| \leqslant c \cdot |u|$ for every $u \in \Sigma^*$. Let $\ulcorner\cdot\urcorner$ denote a reasonable encoding function, and let R be a completely defined, orthogonal and terminating TRS. We say that an n-ary function $f\colon (\Sigma^*)^n \to \Sigma^*$ is *computable* by R if there exists a defined function symbol f such that for all $w_1,\dots,w_n,v \in \Sigma^*$ $\mathsf{f}(\ulcorner w_1\urcorner,\dots,\ulcorner w_n\urcorner) \to^! \ulcorner v \urcorner \iff f(w_1,\dots,w_n) = v$. On the other hand the TRS R *computes* f, if the function $f\colon (\Sigma^*)^n \to \Sigma^*$ is defined by the above equation.

Below we abbreviate Q_π as Q for predicative interpretation $\mathsf{Q} \in \{\mathsf{S},\mathsf{N},\mathsf{N^s}\}$ and the particular argument filtering π that induces the identity function on terms. Consider the following lemma.

Lemma 5. *Let* R *be an S-sorted and completely defined constructor TRS such that the underlying signature is simple. If* $\mathsf{TPWIDP(R)} \cup \mathsf{U(WIDP(R))} \subseteq \gtrsim_{\mathsf{pop*}}$ *then there exists a polynomial p such that for all ground and well-typed basic terms* $t \in \mathsf{T_b}$, $t^\sharp \xrightarrow{\mathrm{i}}{}^*_{\mathsf{TPWIDP(R)\cup U(WIDP(R))}} s$ *implies* $|s| \leqslant p(|t|)$.

Proof. Let $\mathsf{S} = \mathsf{TPWIDP(R)} \cup \mathsf{U(WIDP(R))}$. Suppose $t^\sharp \xrightarrow{\mathrm{i}}{}^*_{\mathsf{S}} s$, or equivalently $t^\sharp \xrightarrow{\mathrm{v}}{}^*_{\mathsf{S}} s$ since R is completely defined. By Lemma 4 we derive $\mathsf{N^s}(t^\sharp) \blacktriangleright^*_{\sim k} \mathsf{N^s}(s)$ for some uniform $k \in \mathbb{N}$. And thus $\mathsf{G}_k(\mathsf{N^s}(s)) \leqslant \mathsf{G}_k(\mathsf{N^s}(t^\sharp))$. As $\mathsf{G}_k(\mathsf{N^s}(t^\sharp)) = \mathsf{G}_k([\mathsf{N}(t^\sharp)])$ is bounded polynomially in the size of t according to Lemma **??**, we see that there exists a polynomial p such that $\mathsf{G}_k(\mathsf{N^s}(s)) \leqslant \mathsf{G}_k(\mathsf{N^s}(t^\sharp)) \leqslant p(|t|)$. Since R is and S-sorted TRS over a simple signature, the same holds for S due to Lemma 2. And thus since $t^\sharp$ is well-typed and $t^\sharp \xrightarrow{\mathrm{i}}{}^*_{\mathsf{S}} s$ holds, also s is well-typed. Let q be the polynomial as given from Lemma 1 with $|s| \leqslant q(\mathsf{G}_k(\mathsf{N^s}(s)))$. Summing up, we derive $|s| \leqslant q(\mathsf{G}_k(\mathsf{N^s}(s))) \leqslant q(p(|t|))$ as desired. ☐ ☐

The above lemma has established that sizes of reducts with respect to the relation $\xrightarrow{\mathrm{i}}_{\mathsf{TPWIDP(R)\cup U(WIDP(R))}}$ are bounded polynomially in the size of the start term, provided we can orient dependency pairs and usable rules. It remains to verify that this is indeed sufficient to appropriately estimate sizes of reducts with respect to $\xrightarrow{\mathrm{i}}_{\mathsf{R}}$. The fact is established in the final Theorem.

Theorem. *Let* R *be an orthogonal S-sorted and completely defined constructor TRS such that the underlying signature is simple. Let* P *denote the set of weak innermost dependency pairs. Assume* P *is non-duplicating, and suppose* $\mathsf{U(P)} \subseteq >_{\mathsf{A}}$ *for some SLI* A. *If* $\mathsf{P} \subseteq >^{\pi}_{\mathsf{pop*}}$ *and* $\mathsf{U(P)} \subseteq \gtrsim^{\pi}_{\mathsf{pop*}}$ *then the functions computed by* R *are computable in polynomial-time.*

Proof. We single out one of the defined symbols $\mathsf{f} \in \mathsf{D}$ and consider the corresponding function $f\colon (\Sigma^*)^n \to \Sigma^*$ computed by R. Under the assumptions, R is terminating, but moreover $\mathrm{rc}^{\mathrm{i}}_{\mathsf{R}}$ is polynomially bounded according to Theorem **??**. Additionally, from orthogonality (and hence confluence) of R, normal forms are unique and so the function f is well-defined. Suppose $\mathsf{f}(\ulcorner w_1\urcorner,\dots,\ulcorner w_n\urcorner) \to^!_{\mathsf{R}} \ulcorner v \urcorner$ for words $w_1,\dots,w_n,v$. In particular, from confluence we see that

$$\mathsf{f}(\ulcorner w_1\urcorner,\dots,\ulcorner w_n\urcorner) \xrightarrow{\mathrm{i}}_{\mathsf{R}} t_1 \xrightarrow{\mathrm{i}}_{\mathsf{R}} \cdots \xrightarrow{\mathrm{i}}_{\mathsf{R}} t_\ell = \ulcorner v \urcorner\,.$$

It is folklore that there exists a polytime algorithm performing one rewrite step. Hence to conclude the existence of a polytime algorithm for f it suffices to bound the size of terms t_i for $1 \leqslant i \leqslant \ell$ polynomially in $\sum_i |w_i|$. And as we suppose that the encoding $\ulcorner\cdot\urcorner$ is reasonable, it thus suffice to bound the sizes of t_i for $i \in \{1,\dots,\ell\}$ polynomially in the size of $t_0 = \mathsf{f}(\ulcorner w_1 \urcorner, \dots, \ulcorner w_n \urcorner)$.

Consider a term t_i. Without loss of generality, we can assume t_i is ground. According to Lemma 3 there exists contexts $C'_i \in \mathsf{ComCtx}$, $C_i \in \mathsf{reprs}(C'_i)$ and terms $u_1,\dots,u_n$ such that $t_i = C_i[u_1,\dots,u_n]$ and moreover, $t_0^\sharp \xrightarrow{\mathsf{i}}^*_{\mathsf{P}\cup\mathsf{U}} C'_i[u_1^\sharp,\dots,u_n^\sharp]$ for all $i \in \{1,\dots,\ell\}$. From the assumption $\mathsf{WIDP}(\mathsf{R}) \subseteq >_{\mathsf{pop}*}$ we see $\mathsf{TPWIDP}(\mathsf{R}) \subseteq >_{\mathsf{pop}*}$. Thus by Lemma 5 there exists a polynomial p such that $|C'_i[u_1^\sharp,\dots,u_n^\sharp]| \leqslant p(|t_0|)$. And so, clearly $\sum_{j=0}^n |u_j| \leqslant p(|t_0|)$. It remains to bound the sizes of contexts C_i polynomially in $|t_0|$.

Recall Definition 5, and recall that $C_i \in \mathsf{reprs}(C'_i)$. Thus C_i is a context build from constructors and variables, where the latter are replaced by normal forms of R. Since R is completely defined, $\mathsf{NF}(\mathsf{R})$ coincides with values. We conclude that $C_i \in \mathsf{T}(\mathsf{C} \cup \{\Box_s \mid s \in S\})$. Here $\Box_s$ denotes the hole of sort s. Moreover since R is S-sorted, and $t_0 \xrightarrow{\mathsf{i}}^*_{\mathsf{R}} C_i[u_1,\dots,u_n]$, we see that C_i is well-typed. We define $\triangle_{\mathsf{R}} = \max\{\mathsf{dp}(r) \mid l \to r \in \mathsf{R}\}$. By a straight forward induction it follows that $\mathsf{dp}(t_i) \leqslant \mathsf{dp}(t_0) + \triangle_{\mathsf{R}} \cdot i \leqslant |t_0| + \triangle_{\mathsf{R}} \cdot \mathrm{dl}(t_0, \xrightarrow{\mathsf{i}}_{\mathsf{R}})$. As a consequence, $\mathsf{dp}(C_i) \leqslant |t_0| + \triangle_{\mathsf{R}} \cdot \mathrm{dl}(t_0, \xrightarrow{\mathsf{i}}_{\mathsf{R}})$, and thus by Proposition 1 there exists constants $c, d \in \mathbb{N}$ such that $|C_i| \leqslant c \cdot \mathsf{dp}(C_i)^d \leqslant c \cdot (|t_0| + \triangle_{\mathsf{R}} \cdot \mathrm{dl}(t_0, \xrightarrow{\mathsf{i}}_{\mathsf{R}}))^d$. As we have that $\mathrm{dl}(t_0, \xrightarrow{\mathsf{i}}_{\mathsf{R}})$ is polynomially bounded in the size of t_0, it follows that $|C_i| \leqslant q(|t_0|)$ for some polynomial q.

Summing up, we conclude that for all $i \in \{1,\dots,\ell\}$, $|t_i| \leqslant p(|t_0|) + q(|t_0|)$ for the polynomials p and q from above. This concludes the theorem. □ □

References

[1] Martin Avanzini. Automation of polynomial path orders. Master's thesis, University of Innsbruck, Faculty for Computer Science., 2009. Available at `http://cl-informatik.uibk.ac.at/~zini/MT.pdf`.

[2] Martin Avanzini and Georg Moser. Complexity analysis by rewriting. In *Proc. 9th FLOPS*, volume 4989 of *LNCS*, pages 130–146, 2008.

[3] Martin Avanzini and Georg Moser. Complexity analysis by rewriting. Technical report, University of Innsbruck, Faculty for Computer Science, 2008.

[4] Martin Avanzini and Georg Moser. Dependency pairs and polynomial path orders. To appear. In *Proc. 20th RTA*, LNCS, 2009.

[5] Stephen Bellantoni. *Predicative recursion and computational complexity*. PhD thesis, University of Torronto, Faculty for Computer Science, 1992.

[6] Stephen Bellantoni and Stephen Cook. A new recursion-theoretic characterization of the polytime functions. *CC*, 2(2):97–110, 1992.

[7] Alan Cobham. The intrinsic computational difficulty of functions. In *Proc. 1964 LMPS*, pages 24–30, 1964.

[8] Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Proc. 6th SAT*, volume 2919 of *LNCS*, pages 502–518, 2003.

[9] Nao Hirokawa and Georg Moser. Automated complexity analysis based on the dependency pair method. In *Proc. 4th IJCAR*, volume 5195 of *LNCS*, pages 364–380, 2008.

[10] Nao Hirokawa and Georg Moser. Complexity, graphs, and the dependency pair method. In *Proc. 15th LPAR*, volume 5330 of *LNCS*, pages 667–681, 2008.

[11] Jean-Yves Marion. Analysing the implicit complexity of programs. *IC*, 183:2–18, 2003.

[12] Andreas Weiermann. A termination ordering for primitive recursive schemata. Preprint.