Dependency Pairs and Polynomial Path Orders*

Martin Avanzini and Georg Moser

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

Abstract. We show how polynomial path orders can be employed efficiently in conjunction with weak innermost dependency pairs to automatically certify the polynomial runtime complexity of term rewrite systems and the polytime computability of those functions computed by these rewrite systems. The established techniques have been implemented and we provide ample experimental data to assess the new method.

1 Introduction

In order to measure the complexity of a (terminating) term rewrite system (TRS for short) it is natural to look at the maximal length of derivation sequences—the *derivation length*—as suggested by Hofbauer and Lautemann in [1]. More precisely, the *runtime complexity function* with respect to a (finite and terminating) TRS $\mathcal{R}$ relates the maximal derivation length to the size of the initial term, whenever the set of initial terms is restricted to constructor based terms, also called *basic* terms. The restriction to basic terms allows us to accurately express the runtime complexity of programs through the runtime complexity of TRSs. In this paper we study and combine recent efforts for the *automatic* analysis of runtime complexities of TRSs. In [2] we introduced a restriction of the multiset path order, called *polynomial path order* (POP^* for short) that induces polynomial runtime complexity if restricted to innermost rewriting. The definition of POP^* employs the idea of *tiered recursion* [3]. Syntactically this amounts to a separation of arguments into *normal* and *safe* arguments, cf. [4]. Furthermore, Hirokawa and the second author introduced a variant of dependency pairs, dubbed *weak dependency pairs*, that makes the dependency pair method applicable in the context of complexity analysis, cf. [5,6].

We show how weak innermost dependency pairs can be successfully applied in conjunction with POP^*. The following example (see [7]) motivates this study. Consider the TRS $\mathcal{R}_{\mathsf{bits}}$ encoding the function $\lambda x.\lceil \log(x+1) \rceil$ for natural numbers given as tally sequences:

$$
\begin{array}{llll}
1: & \mathsf{half}(\mathsf{0}) \to \mathsf{0} & 4: & \mathsf{bits}(\mathsf{0}) \to \mathsf{0} \\
2: & \mathsf{half}(\mathsf{s}(\mathsf{0})) \to \mathsf{0} & 5: & \mathsf{bits}(\mathsf{s}(\mathsf{0})) \to \mathsf{s}(\mathsf{0}) \\
3: & \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)) & 6: & \mathsf{bits}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{bits}(\mathsf{s}(\mathsf{half}(x))))
\end{array}
$$

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

It is easy to see that the TRS $\mathcal{R}_{\mathsf{bin}}$ is not compatible with POP*, even if we allow quasi-precedences, see Section 4. On the other hand, employing (weak innermost) dependency pairs, argument filtering, and the usable rules criteria in conjunction with POP*, polynomial innermost runtime complexity of $\mathcal{R}_{\mathsf{bin}}$ can be shown fully automatically.

The combination of dependency pairs and polynomial path orders turns out to be technically involved. One of the first obstacles one encounters is that the pair $(\gtrsim_{\mathsf{pop}*}, >_{\mathsf{pop}*})$ cannot be used as a reduction pair, as $\gtrsim_{\mathsf{pop}*}$ fails to be closed under contexts. (This holds a fortiori for *safe* reduction pairs, as studied in [5].) Conclusively, we start from scratch and study polynomial path orders in the context of *relative rewriting* [8]. Based on this study an incorporation of argument filterings becomes possible so that we can employ the pair $(\gtrsim^{\pi}_{\mathsf{pop}*}, >^{\pi}_{\mathsf{pop}*})$ in conjunction with dependency pairs successfully. Here, $>^{\pi}_{\mathsf{pop}*}$ refers to the order obtained by combining $>_{\mathsf{pop}*}$ with the argument filtering π as expected, and $\gtrsim^{\pi}_{\mathsf{pop}*}$ denotes the extension of $>^{\pi}_{\mathsf{pop}*}$ by term equivalence, preserving the separation of safe and normal argument positions. Note that for polynomial path orders, the integration of argument filterings is not only non-trivial, but indeed a challenging task. This is mainly due to the embodiment of tiered recursion in POP*. Thus we establish a combination of two syntactic techniques for automatic runtime complexity analysis. The experimental evidence given below indicates the power and in particular the efficiency of the provided results.

Our next contribution is concerned with *implicit complexity theory*, see for example [9]. A careful analysis of our main result shows that polynomial path orders in conjunction with (weak innermost) dependency pairs even induce polytime computability of the functions defined by the TRS studied. This result fits well with recent results by Marion and Péchoux on the use of restricted forms of the dependency pair method to characterise complexity classes like PTIME or PSPACE, cf. [10]. Both results allow to conclude, based on different restrictions, polytime computability of the functions defined by constructor TRSs, whose termination can be shown by the dependency pair method. Note that the results in [10] also capture programs that admit infeasible runtime complexities, but define functions that are computable in polytime, if suitable (and non-trivial) program transformations are used. Such programs are outside the scope of our results. Thus it seems that our results more directly assess the complexity of the given programs. Note that our tool provides (for the first time) a fully automatic application of the dependency pair method in the context of implicit complexity theory. Here we only want to mention that for a variant of the TRS $\mathcal{R}_{\mathsf{bin}}$, as studied in [10], our tool easily verifies polytime computability fully automatically.

The rest of the paper is organised as follows. In Section 2 we present basic notions and recall (briefly) the *path order for FP* from [11]. We then briefly recall dependency pairs in the context of complexity analysis from [5,6], cf. Section 3. In Section 4 we present polynomial path orders over quasi-precedences. Our main results are presented in Section 5. We continue with experimental results in Section 6, and conclude in Section 7.

2 The Polynomial Path Order on Sequences

We assume familiarity with the basics of term rewriting, see [12,13]. Let $\mathcal{V}$ denote a countably infinite set of variables and $\mathcal{F}$ a signature, containing at least one constant. The set of terms over $\mathcal{F}$ and $\mathcal{V}$ is denoted as $\mathcal{T}(\mathcal{F},\mathcal{V})$ and the set of ground terms as $\mathcal{T}(\mathcal{F})$. We write $\mathsf{Fun}(t)$ and $\mathsf{Var}(t)$ for the set of function symbols and variables appearing in t, respectively. The root symbol $\mathsf{rt}(t)$ of a term t is defined as usual and the (proper) subterm relation is denoted as $\trianglelefteq$ ($\triangleleft$). We write $s|_p$ for the *subterm* of s at position p. The *size* $|t|$ of a term t is defined as usual and the *width* of t is defined as $\mathsf{width}(t) := \max\{n, \mathsf{width}(t_1), \dots, \mathsf{width}(t_n)\}$ if $t = f(t_1,\dots,t_n)$ and $n > 0$ or $\mathsf{width}(t) = 1$ else. Let $\succsim$ be a preorder on the signature $\mathcal{F}$, called *quasi-precedence* or simply *precedence*. Based on $\succsim$ we define an equivalence $\approx$ on terms: $s \approx t$ if either (i) $s = t$ or (ii) $s = f(s_1,\dots,s_n)$, $t = g(t_1,\dots,t_n)$, $f \approx g$ and there exists a permutation π such that $s_i \approx t_{\pi(i)}$. For a preorder $\succsim$, we use $\succsim^{\mathsf{mul}}$ for the multiset extension of $\succsim$, which is again a preorder. The proper order (equivalence) induced by $\succsim^{\mathsf{mul}}$ is written as $\succ^{\mathsf{mul}}$ ($\approx^{\mathsf{mul}}$).

A *term rewrite system* (*TRS* for short) $\mathcal{R}$ over $\mathcal{T}(\mathcal{F},\mathcal{V})$ is a *finite* set of rewrite rules $l \to r$, such that $l \notin \mathcal{V}$ and $\mathsf{Var}(l) \supseteq \mathsf{Var}(r)$. We write $\to_{\mathcal{R}}$ ($\xrightarrow{\mathsf{i}}_{\mathcal{R}}$) for the induced (innermost) rewrite relation. The set of defined function symbols is denoted as $\mathcal{D}$, while the constructor symbols are collected in $\mathcal{C}$, clearly $\mathcal{F} = \mathcal{D} \cup \mathcal{C}$. We use $\mathsf{NF}(\mathcal{R})$ to denote the set of normal forms of $\mathcal{R}$ and set $\mathsf{Val} := \mathcal{T}(\mathcal{C},\mathcal{V})$. The elements of Val are called *values*. A TRS is called *completely defined* if normal forms coincide with values. We define $\mathcal{T}_{\mathsf{b}} := \{f(v_1,\dots,v_n) \mid f \in \mathcal{D} \text{ and } v_i \in \mathsf{Val}\}$ as the set of *basic terms*. A TRS $\mathcal{R}$ is a *constructor TRS* if $l \in \mathcal{T}_{\mathsf{b}}$ for all $l \to r \in \mathcal{R}$. Let $\mathcal{Q}$ denote a TRS. The *generalised restricted rewrite relation* $\xrightarrow{\mathcal{Q}}_{\mathcal{R}}$ is the restriction of $\to_{\mathcal{R}}$ where all arguments of the redex are in normal form with respect to the TRS $\mathcal{Q}$ (compare [14]). We define the (innermost) relative rewriting relation (denoted as $\xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}}$) as follows:

$$\xrightarrow{\mathsf{i}}_{\mathcal{R}/\mathcal{S}} := \xrightarrow{\mathcal{R}\cup\mathcal{S}}{}^{*}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{R}\cup\mathcal{S}}_{\mathcal{R}} \cdot \xrightarrow{\mathcal{R}\cup\mathcal{S}}{}^{*}_{\mathcal{S}} ,$$

Similarly, we set $\xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}} := \xrightarrow{\mathcal{R}\cup\mathcal{S}}{}^{*}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{R}\cup\mathcal{S}}{}^{\varepsilon}_{\mathcal{R}} \cdot \xrightarrow{\mathcal{R}\cup\mathcal{S}}{}^{*}_{\mathcal{S}}$, to define an *(innermost) relative root-step*.

A *polynomial interpretation* is a well-founded and monotone algebra $(\mathcal{A}, >)$ with carrier $\mathbb{N}$ such that $>$ is the usual order on natural numbers and all interpretation functions $f_{\mathcal{A}}$ are polynomials. Let $\alpha\colon \mathcal{V} \to \mathcal{A}$ denote an *assignment*, then we write $[\alpha]_{\mathcal{A}}(t)$ for the evaluation of term t with respect to $\mathcal{A}$ and α. A polynomial interpretation is called a *strongly linear interpretation* (*SLI* for short) if all function symbols are interpreted by *weight functions* $f_{\mathcal{A}}(x_1,\dots,x_n) = \sum_{i=1}^{n} x_i + c$ with $c \in \mathbb{N}$. The *derivation length* of a terminating term s with respect to $\to$ is defined as $\mathrm{dl}(s,\to) := \max\{n \mid \exists t.\ s \to^n t\}$, where $\to^n$ denotes the n-fold application of $\to$. The *innermost runtime complexity function* $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}$ with respect to a TRS $\mathcal{R}$ is defined as $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}(n) := \max\{\mathrm{dl}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}}) \mid t \in \mathcal{T}_{\mathsf{b}} \text{ and } |t| \leqslant n\}$. If no confusion can arise $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}$ is simply called *runtime complexity function*.

We recall the bare essentials of the polynomial path order $\blacktriangleright$ on sequences (POP for short) as put forward in [11]. We kindly refer the reader to [11,2] for motivation and examples. We recall the definition of *finite approximations* $\blacktriangleright_k^l$ of $\blacktriangleright$. The latter is conceived as the *limit* of these approximations. The domain of this order are so-called *sequences* $\mathcal{S}\mathrm{eq}(\mathcal{F},\mathcal{V}) := \mathcal{T}(\mathcal{F} \cup \{\circ\},\mathcal{V})$. Here $\mathcal{F}$ is a finite signature and $\circ \notin \mathcal{F}$ a fresh variadic function symbol, used to form sequences. We denote sequences $\circ(s_1,\dots,s_n)$ by $[s_1 \cdots s_n]$ and write $a :: [b_1 \cdots b_n]$ for the sequence $[a\ b_1 \cdots b_n]$.

Let $\succsim$ denote a precedence. The order $\blacktriangleright_k^l$ is based on an auxiliary order $>\!\!\!>_k^l$. Below we set $\gtrsim_k^l := >\!\!\!>_k^l \cup \approx$, where $\approx$ denotes the equivalence on terms defined above. We write $\{\!\{t_1,\dots,t_n\}\!\}$ to denote multisets and $\uplus$ for the multiset sum.

Definition 1. *Let $k, l \geqslant 1$. The order $>\!\!\!>_k^l$ induced by $\succsim$ is inductively defined as follows: $s >\!\!\!>_k^l t$ for $s = f(s_1,\dots,s_n)$ or $s = [s_1 \cdots s_n]$ if either*

(1) $s_i \gtrsim_k^l t$ for some $i \in \{1,\dots,n\}$, or
(2) $s = f(s_1,\dots,s_n)$, $t = g(t_1,\dots,t_m)$ with $f \succ g$ or $t = [t_1 \cdots t_m]$, $s >\!\!\!>_k^{l-1} t_j$ for all $j \in \{1,\dots,m\}$, and $m < k + \mathsf{width}(s)$,
(3) $s = [s_1 \cdots s_n]$, $t = [t_1 \cdots t_m]$ and the following properties hold:
  - *$\{\!\{t_1,\dots,t_m\}\!\} = N_1 \uplus \cdots \uplus N_n$ for some multisets $N_1,\dots,N_n$, and*
  - *there exists $i \in \{1,\dots,n\}$ such that $\{\!\{s_i\}\!\} \not\approx^{\mathsf{mul}} N_i$, and*
  - *for all $1 \leqslant i \leqslant n$ such that $\{\!\{s_i\}\!\} \not\approx^{\mathsf{mul}} N_i$ we have $s_i >\!\!\!>_k^l r$ for all $r \in N_i$, and $m < k + \mathsf{width}(s)$.*

Definition 2. *Let $k, l \geqslant 1$. The* approximation $\blacktriangleright_k^l$ of the polynomial path order on sequences *induced by $\succsim$ is inductively defined as follows: $s \blacktriangleright_k^l t$ for $s = f(s_1,\dots,s_n)$ or $s = [s_1 \cdots s_n]$ if either $s >\!\!\!>_k^l t$ or*

(1) $s_i \underset{\sim}{\blacktriangleright}_k^l t$ for some $i \in \{1,\dots,n\}$,
(2) $s = f(s_1,\dots,s_n)$, $t = [t_1 \cdots t_m]$, and the following properties hold:
  - *$s \blacktriangleright_k^{l-1} t_{j_0}$ for some $j_0 \in \{1,\dots,m\}$,*
  - *$s >\!\!\!>_k^{l-1} t_j$ for all $j \neq j_0$, and $m < k + \mathsf{width}(s)$,*

(3) $s = f(s_1,\dots,s_n)$, $t = g(t_1,\dots,t_m)$, $f \sim g$ and $[s_1 \cdots s_n] \blacktriangleright_k^l [t_1 \cdots t_m]$, or
(4) $s = [s_1 \cdots s_n]$, $t = [t_1 \cdots t_m]$ and the following properties hold:
  - *$\{\!\{t_1,\dots,t_m\}\!\} = N_1 \uplus \cdots \uplus N_n$ for some multisets $N_1,\dots,N_n$, and*
  - *there exists $i \in \{1,\dots,n\}$ such that $\{\!\{s_i\}\!\} \not\approx^{\mathsf{mul}} N_i$, and*
  - *for all $1 \leqslant i \leqslant n$ such that $\{\!\{s_i\}\!\} \not\approx^{\mathsf{mul}} N_i$ we have $s_i \blacktriangleright_k^l r$ for all $r \in N_i$, and $m < k + \mathsf{width}(s)$.*

Above we set $\underset{\sim}{\blacktriangleright}_k^l := \blacktriangleright_k^l \cup \approx$ and abbreviate $\blacktriangleright_k^k$ as $\blacktriangleright_k$ in the following. Note that the empty sequence $[\,]$ is minimal with respect to both orders.

It is easy to see that for $k \leqslant l$, we have $>\!\!\!>_k \subseteq >\!\!\!>_l$ and $\blacktriangleright_k \subseteq \blacktriangleright_l$ and that $s \blacktriangleright_k t$ implies that $\mathsf{width}(t) < \mathsf{width}(s) + k$. For a fixed approximation $\blacktriangleright_k$, we define the length of its longest decent as: $\mathsf{G}_k(t) := \max\{n \mid t = t_0 \blacktriangleright_k t_1 \blacktriangleright_k \cdots \blacktriangleright_k t_n\}$. The following proposition is a reformulation of Lemma 6 in [11].

Proposition 3. *Let $k \in \mathbb{N}$. There exists a polynomial interpretation $\mathcal{A}$ such that $\mathsf{G}_k(t) \leqslant [\alpha]_{\mathcal{A}}(t)$ for all assignments $\alpha\colon \mathcal{V} \to \mathbb{N}$. As a consequence, for all terms $f(t_1,\dots,t_n)$ with $[\alpha]_{\mathcal{A}}(t_i) = \mathsf{O}(|t_i|)$, $\mathsf{G}_k(f(t_1,\dots,t_n))$ is bounded by a polynomial p in the size of t, where p depends on k only.*

Note that the polynomial interpretation $\mathcal{A}$ employed in the proposition fulfils: $\circ_{\mathcal{A}}(m_1,\dots,m_n) = \sum_{i=1}^n m_i + n$. In particular, we have $[\alpha]_{\mathcal{A}}([\,]) = 0$.

3 Complexity Analysis Based on the Dependency Pair Method

In this section, we briefly recall the central definitions and results established in [5,6]. We kindly refer the reader to [5,6] for further examples and underlying intuitions. Let $\mathcal{X}$ be a set of symbols. We write $C\langle t_1, \ldots, t_n \rangle_{\mathcal{X}}$ to denote $C[t_1, \ldots, t_n]$, whenever $\mathsf{rt}(t_i) \in \mathcal{X}$ for all $i \in \{1, \ldots, n\}$ and C is a n-hole context containing no symbols from $\mathcal{X}$. We set $\mathcal{D}^\sharp := \mathcal{D} \cup \{f^\sharp \mid f \in \mathcal{D}\}$ with each $f^\sharp$ a fresh function symbol. Further, for $t = f(t_1, \ldots, t_n)$ with $f \in \mathcal{D}$, we set $t^\sharp := f^\sharp(t_1, \ldots, t_n)$.

Definition 4. *Let $\mathcal{R}$ be a TRS. If $l \to r \in \mathcal{R}$ and $r = C\langle u_1, \ldots, u_n \rangle_{\mathcal{D}}$ then $l^\sharp \to \mathrm{COM}(u_1^\sharp, \ldots, u_n^\sharp)$ is called a* weak innermost dependency pair *of $\mathcal{R}$. Here $\mathrm{COM}(t) = t$ and $\mathrm{COM}(t_1, \ldots, t_n) = \mathsf{c}(t_1, \ldots, t_n)$, $n \neq 1$, for a fresh constructor symbol c, the* compound symbol*. The set of all weak innermost dependency pairs is denoted by $\mathsf{WIDP}(\mathcal{R})$.*

Example 5. Reconsider the example $\mathcal{R}_{\mathsf{bits}}$ from the introduction. The set of weak innermost dependency pairs $\mathsf{WIDP}(\mathcal{R}_{\mathsf{bits}})$ is given by

$$\begin{array}{rlrl}
7\colon & \mathsf{half}^\sharp(\mathsf{0}) \to \mathsf{c}_1 & 10\colon & \mathsf{bits}^\sharp(\mathsf{0}) \to \mathsf{c}_3 \\
8\colon & \mathsf{half}^\sharp(\mathsf{s}(\mathsf{0})) \to \mathsf{c}_2 & 11\colon & \mathsf{bits}^\sharp(\mathsf{s}(\mathsf{0})) \to \mathsf{c}_4 \\
9\colon & \mathsf{half}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{half}^\sharp(x) & 12\colon & \mathsf{bits}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{bits}^\sharp(\mathsf{s}(\mathsf{half}(x)))
\end{array}$$

We write $f \rhd_{\mathrm{d}} g$ if there exists a rewrite rule $l \to r \in \mathcal{R}$ such that $f = \mathsf{rt}(l)$ and g is a defined symbol in $\mathsf{Fun}(r)$. For a set $\mathcal{G}$ of defined symbols we denote by $\mathcal{R}{\restriction}\mathcal{G}$ the set of rewrite rules $l \to r \in \mathcal{R}$ with $\mathsf{rt}(l) \in \mathcal{G}$. The set $\mathcal{U}(t)$ of usable rules of a term t is defined as $\mathcal{R}{\restriction}\{g \mid f \rhd_{\mathrm{d}}^* g \text{ for some } f \in \mathsf{Fun}(t)\}$. Finally, we define $\mathcal{U}(\mathcal{P}) = \bigcup_{l \to r \in \mathcal{P}} \mathcal{U}(r)$.

Example 6 (Example 5 continued). The usable rules of $\mathsf{WIDP}(\mathcal{R}_{\mathsf{bits}})$ consist of the following rules: $1\colon \mathsf{half}(\mathsf{0}) \to \mathsf{0}$, $2\colon \mathsf{half}(\mathsf{s}(\mathsf{0})) \to \mathsf{0}$, and $3\colon \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{half}(x)$.

The following proposition allows the analysis of the (innermost) runtime complexity through the study of (innermost) relative rewriting, see [5] for the proof.

Proposition 7. *Let $\mathcal{R}$ be a TRS, let t be a basic terminating term, and let $\mathcal{P} = \mathsf{WIDP}(\mathcal{R})$. Then $\mathrm{dl}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \leqslant \mathrm{dl}(t^\sharp, \xrightarrow{\mathrm{i}}_{\mathcal{U}(\mathcal{P}) \cup \mathcal{P}})$. Moreover, suppose $\mathcal{P}$ is non-duplicating and $\mathcal{U}(\mathcal{P}) \subseteq >_{\mathcal{A}}$ for some SLI $\mathcal{A}$. Then there exist constants $K, L \geqslant 0$ (depending on $\mathcal{P}$ and $\mathcal{A}$ only) such that $\mathrm{dl}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \leqslant K \cdot \mathrm{dl}(t^\sharp, \xrightarrow{\mathrm{i}}_{\mathcal{P}/\mathcal{U}(\mathcal{P})}) + L \cdot |t^\sharp|$.*

This approach admits also an integration of *dependency graphs* [15] in the context of complexity analysis. The nodes of the *weak innermost dependency graph* $\mathsf{WIDG}(\mathcal{R})$ are the elements of $\mathcal{P}$ and there is an arrow from $s \to t$ to $u \to v$ if there exist a context C and substitutions σ, τ such that $t\sigma \xrightarrow{\mathrm{i}}_{\mathcal{R}}^* C[u\tau]$. Let $\mathcal{G} = \mathsf{WIDG}(\mathcal{R})$; a *strongly connected component* (*SCC* for short) in $\mathcal{G}$ is a maximal *strongly connected subgraph*. We write $\mathcal{G}/_{\equiv}$ for the *congruence graph*, where $\equiv$ is the equivalence relation induced by SCCs.

Example 8 (Example 5 continued). $\mathcal{G} = \mathsf{WIDG}(\mathcal{R}_{\mathsf{bits}})$ consists of the nodes (7)–(12) as mentioned in Example 5 and has the following shape:

$$\begin{tikzcd} 7 & 9 \arrow[l] \arrow[r] \arrow[loop above] & 8 & 10 & 12 \arrow[r] \arrow[loop above] & 11 \end{tikzcd}$$

The only non-trivial SCCs in $\mathcal{G}$ are $\{9\}$ and $\{12\}$. Hence $\mathcal{G}/_{\equiv}$ consists of the nodes $[7]_{\equiv}$–$[12]_{\equiv}$, and edges $([a]_{\equiv}, [b]_{\equiv})$ for edges (a, b) in $\mathcal{G}$. Here $[a]_{\equiv}$ denotes the equivalence class of a.

We set $\mathsf{L}(t) := \max\{\mathrm{dl}(t, \xrightarrow{\mathsf{i}}_{\mathcal{P}_m/\mathcal{S}}) \mid (\mathcal{P}_1, \ldots, \mathcal{P}_m) \text{ a path in } \mathcal{G}/_{\equiv}, \mathcal{P}_1 \in \mathsf{Src}\}$, where Src denotes the set of source nodes from $\mathcal{G}/_{\equiv}$ and $\mathcal{S} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_{m-1} \cup \mathcal{U}(\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_m)$. The proposition allows the use of different techniques to analyse polynomial runtime complexity on separate paths, cf. [6].

Proposition 9. *Let $\mathcal{R}$, $\mathcal{P}$, and t be as above. Then there exists a polynomial p (depending only on $\mathcal{R}$) such that $\mathrm{dl}(t^\sharp, \xrightarrow{\mathsf{i}}_{\mathcal{P}/\mathcal{U}(\mathcal{P})}) \leqslant p(\mathsf{L}(t^\sharp))$.*

4 The Polynomial Path Order over Quasi-Precedences

In this section, we briefly recall the central definitions and results established in [2,16] on the *polynomial path order*. We employ the variant of POP* based on quasi-precendences, cf. [16].

As mentioned in the introduction, POP* relies on tiered recursion, which is captured 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*. If $f \in \mathcal{D}$ then $\mathsf{safe}(f) \subseteq \{1, \ldots, n\}$, for $f \in \mathcal{C}$ we fix $\mathsf{safe}(f) = \{1, \ldots, n\}$. The argument positions not included in $\mathsf{safe}(f)$ are called *normal* and denoted by $\mathsf{nrm}(f)$. We extend safe to terms $t \notin \mathcal{V}$. We define $\mathsf{safe}(f(t_1, \ldots, t_n)) := \{t_{i_1}, \ldots, t_{i_p}\}$ where $\mathsf{safe}(f) = \{i_1, \ldots, i_p\}$, and we define $\mathsf{nrm}(f(t_1, \ldots, t_n)) := \{t_{j_1}, \ldots, t_{j_q}\}$ where $\mathsf{nrm}(f) = \{j_1, \ldots, j_q\}$. Not every precedence is suitable for $>_{\mathsf{pop}*}$, in particular we need to assert that constructors are minimal.

We say that a precedence $\succsim$ is *admissible* for POP* if the following is satisfied: (i) $f \succ g$ with $g \in \mathcal{D}$ implies $f \in \mathcal{D}$, and (ii) if $f \approx g$ then $f \in \mathcal{D}$ if and only if $g \in \mathcal{D}$. In the sequel we assume any precedence is admissible. We extend the equivalence $\approx$ to the context of safe mappings: $s \overset{\mathsf{safe}}{\approx} t$, if (i) $s = t$, or (ii) $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_n)$, $f \approx g$ and there exists a permutation π so that $s_i \overset{\mathsf{safe}}{\approx} t_{\pi(i)}$, where $i \in \mathsf{safe}(f)$ if and only if $\pi(i) \in \mathsf{safe}(g)$ for all $i \in \{1, \ldots, n\}$. Similar to POP, the definition of the polynomial path order $>_{\mathsf{pop}*}$ makes use of an auxiliary order $>_{\mathsf{pop}}$.

Definition 10. *The auxiliary order $>_{\mathsf{pop}}$ induced by $\succsim$ and safe is inductively defined as follows: $s = f(s_1, \ldots, s_n) >_{\mathsf{pop}} t$ if either*

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

(2) $t = g(t_1, \ldots, t_m)$, $f \succ g$, $f \in \mathcal{D}$ and $s >_{\mathsf{pop}} t_j$ for all $j \in \{1, \ldots, m\}$.

Definition 11. *The* polynomial path order $>_{\mathsf{pop*}}$ *induced by* $\succsim$ *and* safe *is inductively defined as follows:* $s = f(s_1, \ldots, s_n) >_{\mathsf{pop*}} t$ *if either* $s >_{\mathsf{pop}} t$ *or*

(1) $s_i \gtrsim_{\mathsf{pop*}} t$ *for some* $i \in \{1, \ldots, n\}$*, or*
(2) $t = g(t_1, \ldots, t_m)$*,* $f \succ g$*,* $f \in \mathcal{D}$*, and*
   – $s >_{\mathsf{pop*}} t_{j_0}$ *for some* $j_0 \in \mathsf{safe}(g)$*, and*
   – *for all* $j \neq j_0$ *either* $s >_{\mathsf{pop}} t_j$*, or* $s \rhd t_j$ *and* $j \in \mathsf{safe}(g)$*, or*
(3) $t = g(t_1, \ldots, t_m)$*,* $f \approx g$*,* $\mathsf{nrm}(s) >^{\mathsf{mul}}_{\mathsf{pop*}} \mathsf{nrm}(t)$ *and* $\mathsf{safe}(s) \gtrsim^{\mathsf{mul}}_{\mathsf{pop*}} \mathsf{safe}(t)$*.*

Above we set $\gtrsim_{\mathsf{pop}} := >_{\mathsf{pop}} \cup \overset{\mathsf{safe}}{\approx}$ and $\gtrsim_{\mathsf{pop*}} := >_{\mathsf{pop*}} \cup \overset{\mathsf{safe}}{\approx}$. Here $>^{\mathsf{mul}}_{\mathsf{pop*}}$ and $\gtrsim^{\mathsf{mul}}_{\mathsf{pop*}}$ refer to the strict and weak multiset extension of $\gtrsim_{\mathsf{pop*}}$ respectively.

The intuition of $>_{\mathsf{pop}}$ is to deny any recursive call, whereas $>_{\mathsf{pop*}}$ allows predicative recursion: by the restrictions imposed by the mapping safe, recursion needs to be performed on normal arguments, while a recursively computed result must only be used in a safe argument position, compare [4]. Note that the alternative $s \rhd t_j$ for $j \in \mathsf{safe}(g)$ in Definition 11(2) guarantees that POP* characterises the class of polytime computable functions. The proof of the next theorem follows the pattern of the proof of the Main Theorem in [2], but the result is stronger due to the extension to quasi-precedences.

Theorem 12. *Let* $\mathcal{R}$ *be a constructor TRS. If* $\mathcal{R}$ *is compatible with* $>_{\mathsf{pop*}}$*, i.e.,* $\mathcal{R} \subseteq >_{\mathsf{pop*}}$*, then the innermost runtime complexity* $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}$ *induced is polynomially bounded.*

Note that Theorem 12 is too weak to handle the TRS $\mathcal{R}_{\mathsf{bits}}$ as the (necessary) restriction to an admissible precedence is too strong. To rectify this, we analyse POP* in Section 5 in the context of relative rewriting.

An argument filtering (for a signature $\mathcal{F}$) is a mapping π that assigns to every n-ary function symbol f an argument position $i \in \{1, \ldots, n\}$ or a (possibly empty) list $\{k_1, \ldots, k_m\}$ of argument positions with $1 \leqslant k_1 < \cdots < k_m \leqslant n$. The signature $\mathcal{F}_\pi$ consists of all function symbols f such that $\pi(f)$ is some list $\{k_1, \ldots, k_m\}$, where in $\mathcal{F}_\pi$ the arity of f is m. Every argument filtering π induces a mapping from $\mathcal{T}(\mathcal{F}, \mathcal{V})$ to $\mathcal{T}(\mathcal{F}_\pi, \mathcal{V})$, also denoted by π:

$$\pi(t) = \begin{cases} t & \text{if } t \text{ is a variable} \\ \pi(t_i) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } \pi(f) = i \\ f(\pi(t_{k_1}), \ldots, \pi(t_{k_m})) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } \pi(f) = \{k_1, \ldots, k_m\} \ . \end{cases}$$

Definition 13. *Let* π *denote an argument filtering, and* $>_{\mathsf{pop*}}$ *a polynomial path order. We define* $s >^{\pi}_{\mathsf{pop*}} t$ *if and only if* $\pi(s) >_{\mathsf{pop*}} \pi(t)$*, and likewise* $s \gtrsim^{\pi}_{\mathsf{pop*}} t$ *if and only if* $\pi(s) \gtrsim_{\mathsf{pop*}} \pi(t)$*.*

Example 14 (Example 5 continued). Let π be defined as follows: $\pi(\mathsf{half}) = 1$ and $\pi(f) = \{1, \ldots, n\}$ for each n-ary function symbol other than half. Compatibility of $\mathsf{WIDP}(\mathcal{R}_{\mathsf{bits}})$ with $>^{\pi}_{\mathsf{pop*}}$ amounts to the following set of order constraints:

$$\begin{array}{ccc} \mathsf{half}^\sharp(0) >_{\mathsf{pop*}} \mathsf{c}_1 & \mathsf{bits}^\sharp(0) >_{\mathsf{pop*}} \mathsf{c}_3 & \mathsf{half}^\sharp(\mathsf{s}(\mathsf{s}(x))) >_{\mathsf{pop*}} \mathsf{half}^\sharp(x) \\ \mathsf{half}^\sharp(\mathsf{s}(0)) >_{\mathsf{pop*}} \mathsf{c}_2 & \mathsf{bits}^\sharp(\mathsf{s}(0)) >_{\mathsf{pop*}} \mathsf{c}_4 & \mathsf{bits}^\sharp(\mathsf{s}(\mathsf{s}(x))) >_{\mathsf{pop*}} \mathsf{bits}^\sharp(\mathsf{s}(x)) \end{array}$$

In order to define a POP* instance $>_{\mathsf{pop}*}$, we set $\mathsf{safe}(\mathsf{bits}^\sharp) = \mathsf{safe}(\mathsf{half}) = \mathsf{safe}(\mathsf{half}^\sharp) = \varnothing$ and $\mathsf{safe}(\mathsf{s}) = \{1\}$. Furthermore, we define an (admissible) precedence: $0 \approx \mathsf{c}_1 \approx \mathsf{c}_2 \approx \mathsf{c}_3 \approx \mathsf{c}_4$. The easy verification of $\mathsf{WIDP}(\mathcal{R}_{\mathsf{bits}}) \subseteq >^{\pi}_{\mathsf{pop}*}$ is left to the reader.

5 Dependency Pairs and Polynomial Path Orders

Motivated by Example 14, we show in this section that the pair $(\gtrsim^{\pi}_{\mathsf{pop}*}, >^{\pi}_{\mathsf{pop}*})$ can play the role of a *safe* reduction pair, cf. [5,6]. Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ that is innermost terminating. In the sequel $\mathcal{R}$ is kept fixed. Moreover, we fix some safe mapping safe, an admissible precedence $\succsim$, and an argument filtering π. We refer to the induced POP* instance by $>^{\pi}_{\mathsf{pop}*}$.

We adapt safe to $\mathcal{F}_\pi$ in the obvious way: for each $f_\pi \in \mathcal{F}_\pi$ with corresponding $f \in \mathcal{F}$, we define $\mathsf{safe}(f_\pi) := \mathsf{safe}(f) \cap \pi(f)$, and likewise $\mathsf{nrm}(f_\pi) := \mathsf{nrm}(f) \cap \pi(f)$. Set $\mathsf{Val}_\pi := \mathcal{T}(\mathcal{C}_\pi, \mathcal{V})$. Based on $\mathcal{F}_\pi$ we define the *normalised signature* $\mathcal{F}^{\mathsf{n}}_\pi := \{f^{\mathsf{n}} \mid f \in \mathcal{F}_\pi\}$ where the arity of f^{n} is $|\mathsf{nrm}(f)|$. We extend $\succsim$ to $\mathcal{F}^{\mathsf{n}}_\pi$ by $f^{\mathsf{n}} \succsim g^{\mathsf{n}}$ if and only if $f \succsim g$. Let s be a fresh constant that is minimal with respect to $\succsim$. We introduce the *Buchholz norm* of t (denoted as $\|t\|$) as a term complexity measure that fits well with the definition of POP*. Set $\|t\| := 1 + \max\{n, \|t_1\|, \ldots, \|t_n\|\}$ for $t = f(t_1, \ldots, t_n)$ and $\|t\| := 1$, otherwise.

In the following we define an embedding from the relative rewriting relation $\xrightarrow{\mathsf{i}}{}^{\epsilon}_{\mathcal{R}/\mathcal{S}}$ into $\blacktriangleright_k$, such that k depends only on TRSs $\mathcal{R}$ and $\mathcal{S}$. This embedding provides the technical tool to measure the number of root steps in a given derivation through the number of descents in $\blacktriangleright_k$. Hence Proposition 3 becomes applicable to establishing our main result. This intuition is cast into the next definition.

Definition 15. *A* predicative interpretation *is a pair of mappings* $(\mathsf{S}_\pi, \mathsf{N}_\pi)$ *from terms to sequences* $\mathcal{S}\mathrm{eq}(\mathcal{F}^{\mathsf{n}}_\pi \cup \{\mathsf{s}\}, \mathcal{V})$ *defined as follows. We assume* $\pi(t) = f(\pi(t_1), \ldots, \pi(t_n))$*,* $\mathsf{safe}(f) = \{i_1, \ldots, i_p\}$*, and* $\mathsf{nrm}(f) = \{j_1, \ldots, j_q\}$*.*

$$\mathsf{S}_\pi(t) := \begin{cases} [\,] & \text{if } \pi(t) \in \mathsf{Val}_\pi, \\ [f^{\mathsf{n}}(\mathsf{N}_\pi(t_{j_1}), \ldots, \mathsf{N}_\pi(t_{j_q}))\ \mathsf{S}_\pi(t_{i_1})\ \cdots\ \mathsf{S}_\pi(t_{i_p})] & \text{if } \pi(t) \notin \mathsf{Val}_\pi. \end{cases}$$

$$\mathsf{N}_\pi(t) := \mathsf{S}_\pi(t) :: \mathsf{BN}_\pi(t)$$

Here the function BN_π *maps a term* t *to the sequence* $[\mathsf{s}\cdots\mathsf{s}]$ *with* $\|\pi(t)\|$ *occurrences of the constant* s*.*

As a direct consequence of the definitions we have $\mathsf{width}(\mathsf{N}_\pi(t)) = \|\pi(t)\| + 1$ for all terms t.

Lemma 16. *There exists a polynomial p such that* $\mathsf{G}_k(\mathsf{N}_\pi(t)) \leqslant p(|t|)$ *for every basic term t. The polynomial p depends only on k.*

Proof. Suppose $t = f(v_1, \ldots, v_n)$ is a basic term with $\mathsf{safe}(f) = \{i_1, \ldots, i_p\}$ and $\mathsf{nrm}(f) = \{j_1, \ldots, j_q\}$. The only non-trivial case is when $\pi(t) \notin \mathsf{Val}_\pi$. Then $\mathsf{N}_\pi(t) = [u\ \mathsf{S}_\pi(v_{i_1}) \cdots \mathsf{S}_\pi(v_{i_p})] :: \mathsf{BN}_\pi(t)$ where $u = f^{\mathsf{n}}(\mathsf{N}_\pi(v_{j_1}), \ldots, \mathsf{N}_\pi(v_{j_q}))$.

Note that $\mathsf{S}_\pi(v_i) = [\,]$ for $i \in \{i_1, \dots, i_q\}$. Let $\mathcal{A}$ denote a polynomial interpretation fulfilling Proposition 3. Using the assumption $\circ_{\mathcal{A}}(m_1, \dots, m_n) = \sum_{i=1}^{n} m_i + n$, it is easy to see that $\mathsf{G}_k(\mathsf{N}_\pi(t))$ is bounded linear in $\|\pi(t)\| \leqslant |t|$ and $[\alpha]_{\mathcal{A}}(u)$. As $\mathsf{N}_\pi(v_j) = [[\,]\ \mathsf{s} \cdots \mathsf{s}]$ with $\|\pi(v_j)\| \leqslant |t|$ occurrences of s, $\mathsf{G}_k(\mathsf{N}_\pi(v_j))$ is linear in $|t|$. Hence from Proposition 3 we conclude that $\mathsf{G}_k(\mathsf{N}_\pi(t))$ is polynomially bounded in $|t|$. ☐

The next sequence of lemmas shows that the relative rewriting relation $\xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}}$ is embeddable into $\blacktriangleright_k$.

Lemma 17. *Suppose $s >^{\pi}_{\mathsf{pop*}} t$ such that $\pi(s\sigma) \in \mathsf{Val}_\pi$. Then $\mathsf{S}_\pi(s\sigma) = [\,] = \mathsf{S}_\pi(t\sigma)$ and $\mathsf{N}_\pi(s\sigma) \blacktriangleright_1 \mathsf{N}_\pi(t\sigma)$.*

Proof. Let $\pi(s\sigma) \in \mathsf{Val}_\pi$, and suppose $s >^{\pi}_{\mathsf{pop*}} t$, i.e., $\pi(s) >_{\mathsf{pop*}} \pi(t)$ holds. Since $\pi(s) \in \mathsf{Val}_\pi$ (and due to our assumptions on safe mappings) only clause (1) from the definition of $>_{\mathsf{pop*}}$ (or respectively $>_{\mathsf{pop}}$) is applicable. Thus $\pi(t)$ is a subterm of $\pi(s)$ modulo the equivalence $\approx$. We conclude $\pi(t\sigma) \in \mathsf{Val}_\pi$, and hence $\mathsf{S}_\pi(s\sigma) = [\,] = \mathsf{S}_\pi(t\sigma)$. Finally, note that $\|\pi(s\sigma)\| > \|\pi(t\sigma)\|$ as $\pi(t\sigma)$ is a subterm of $\pi(s\sigma)$. Thus $\mathsf{N}_\pi(s\sigma) \blacktriangleright_1 \mathsf{N}_\pi(t\sigma)$ follows as well. ☐

To improve the clarity of the exposition, we concentrate on the crucial cases in the proofs of the following lemmas. The interested reader is kindly referred to [17] for the full proof.

Lemma 18. *Suppose $s >^{\pi}_{\mathsf{pop}} t$ such that $\pi(s\sigma) = f(\pi(s_1\sigma), \dots, \pi(s_n\sigma))$ with $\pi(s_i\sigma) \in \mathsf{Val}_\pi$ for $i \in \{1, \dots, n\}$. Moreover suppose $\mathsf{nrm}(f) = \{j_1, \dots, j_q\}$. Then $f^{\mathsf{n}}(\mathsf{N}_\pi(s_{j_1}\sigma), \dots, \mathsf{N}_\pi(s_{j_q}\sigma)) \gtrdot_{3 \cdot \|\pi(t)\|} \mathsf{N}_\pi(t\sigma)$ holds.*

Proof. Note that the assumption implies that the argument filtering π does not collapse f. We show the lemma by induction on $>^{\pi}_{\mathsf{pop}}$. We consider the subcase that $s >^{\pi}_{\mathsf{pop}} t$ follows as $t = g(t_1, \dots, t_m)$, π does not collapse on g, $f \succ g$, and $s >^{\pi}_{\mathsf{pop}} t_j$ for all $j \in \pi(g)$, cf. Definition 10(2). We set $u := f^{\mathsf{n}}(\mathsf{N}_\pi(s_{j_1}\sigma), \dots, \mathsf{N}_\pi(s_{j_q}\sigma))$ and $k := 3 \cdot \|\pi(t)\|$ and first prove $u \gtrdot_{k-1} \mathsf{S}_\pi(t\sigma)$.

If $\pi(t\sigma) \in \mathsf{Val}_\pi$, then $\mathsf{S}_\pi(t\sigma) = [\,]$ is minimal with respect to $\gtrdot_{k-1}$. Thus we are done. Hence suppose $\mathsf{nrm}(g) = \{j'_1, \dots, j'_q\}$, $\mathsf{safe}(g) = \{i'_1, \dots, i'_p\}$ and let

$$\mathsf{S}_\pi(t\sigma) = [g^{\mathsf{n}}(\mathsf{N}_\pi(t_{j'_1}\sigma), \dots, \mathsf{N}_\pi(t_{j'_q}\sigma))\ \mathsf{S}_\pi(t_{i'_1}\sigma) \cdots \mathsf{S}_\pi(t_{i'_p}\sigma)]\ .$$

We set $v := g^{\mathsf{n}}(\mathsf{N}_\pi(t_{j'_1}\sigma), \dots, \mathsf{N}_\pi(t_{j'_q}\sigma))$. It suffices to show $u \gtrdot_{k-2} v$ and $u \gtrdot_{k-2} \mathsf{S}_\pi(t_j\sigma)$ for $j \in \mathsf{safe}(g)$. Both assertions follow from the induction hypothesis.

Now consider $\mathsf{N}_\pi(t\sigma) = [\mathsf{S}_\pi(t\sigma)\ \mathsf{s} \cdots \mathsf{s}]$ with $\|\pi(t\sigma)\|$ occurrences of the constant s. Recall that $\mathsf{width}(\mathsf{N}_\pi(t\sigma)) = \|\pi(t\sigma)\| + 1$. Observe that $f^{\mathsf{n}} \succ \mathsf{s}$. Hence to prove $u \gtrdot_k \mathsf{S}_\pi(t\sigma)$ it suffices to observe that $\mathsf{width}(u) + k > \|\pi(t\sigma)\| + 1$ holds. For that note that $\|\pi(t\sigma)\|$ is either $\|\pi(t_j\sigma)\| + 1$ for some $j \in \pi(g)$ or less than k. In the latter case, we are done. Otherwise $\|\pi(t\sigma)\| = \|\pi(t_j\sigma)\| + 1$. Then from the definition of $\gtrdot_k$ and the induction hypothesis $u \gtrdot_{3 \cdot \|\pi(t_j)\|} \mathsf{N}_\pi(t_j\sigma)$ we can conclude $\mathsf{width}(u) + 3 \cdot \|\pi(t_j)\| > \mathsf{width}(\mathsf{N}_\pi(t_j\sigma)) = \|\pi(t_j\sigma)\| + 1$. Since $k \geqslant 3 \cdot (\|\pi(t_j)\| + 1)$, $\mathsf{width}(u) + k > \|\pi(t\sigma)\| + 1$ follows. ☐

Lemma 19. *Suppose $s >^{\pi}_{\mathsf{pop}*} t$ such that $\pi(s\sigma) = f(\pi(s_1\sigma), \dots, \pi(s_n\sigma))$ with $\pi(s_i\sigma) \in \mathsf{Val}_\pi$ for $i \in \{1, \dots, n\}$. Then for $\mathsf{nrm}(f) = \{j_1, \dots, j_q\}$,*

(1) $f^{\mathsf{n}}(\mathsf{N}_\pi(s_{j_1}\sigma), \dots, \mathsf{N}_\pi(s_{j_q}\sigma)) \blacktriangleright_{3\cdot\|\pi(t)\|} \mathsf{S}_\pi(t\sigma)$, and
(2) $f^{\mathsf{n}}(\mathsf{N}_\pi(s_{j_1}\sigma), \dots, \mathsf{N}_\pi(s_{j_q}\sigma)) :: \mathsf{BN}_\pi(s\sigma) \blacktriangleright_{3\cdot\|\pi(t)\|} \mathsf{N}_\pi(t\sigma)$.

Proof. The lemma is shown by induction on the definition of $>^{\pi}_{\mathsf{pop}*}$. Set $u = f^{\mathsf{n}}(\mathsf{N}_\pi(s_{j_1}\sigma), \dots, \mathsf{N}_\pi(s_{j_q}\sigma))$. Suppose $s >^{\pi}_{\mathsf{pop}*} t$ follows due to Definition 11(2). We set $k := 3 \cdot \|\pi(t)\|$. Let $\mathsf{nrm}(g) = \{j'_1, \dots, j'_q\}$ and let $\mathsf{safe}(g) = \{i'_1, \dots, i'_p\}$.

Property (1) is immediate for $\pi(t\sigma) \in \mathsf{Val}_\pi$, so assume otherwise. We see that $s >^{\pi}_{\mathsf{pop}} t_j$ for all $j \in \mathsf{nrm}(g)$ and obtain $u \gtrdot_{k-1} g^{\mathsf{n}}(\mathsf{N}_\pi(t_{j'_1}\sigma), \dots, \mathsf{N}_\pi(t_{j'_q}\sigma))$ as in Lemma 18. Furthermore, $s >^{\pi}_{\mathsf{pop}*} t_{j_0}$ for some $j_0 \in \mathsf{safe}(g)$ and by induction hypothesis: $u \blacktriangleright_{k-1} \mathsf{S}_\pi(t_{j_0}\sigma)$. To conclude property (1), it remains to verify $u \gtrdot_{k-1} \mathsf{S}_\pi(t_j\sigma)$ for the remaining $j \in \mathsf{safe}(g)$. We either have $s >^{\pi}_{\mathsf{pop}} t_j$ or $\pi(s_i) \trianglerighteq \pi(t_j)$ (for some i). In the former subcase we proceed as in the claim, and for the latter we observe $\pi(t_j\sigma) \in \mathsf{Val}_\pi$, and thus $\mathsf{S}_\pi(t_j\sigma) = [\,]$ follows. This establishes property (1).

To conclude property (2), it suffices to show $\mathsf{width}(u :: \mathsf{BN}_\pi(s\sigma)) + k > \mathsf{width}(\mathsf{N}_\pi(t\sigma))$, or equivalently $\|\pi(s\sigma)\| + 1 + k > \|\pi(t\sigma)\|$. The latter can be shown, if we proceed similar as in the claim. □

Recall the definition of $\xrightarrow{\mathcal{Q}}_{\mathcal{R}}$ from Section 2 and define $\mathcal{Q} := \{f(x_1, \dots, x_n) \to \bot \mid f \in \mathcal{D}\}$, and set $\xrightarrow{\mathsf{v}}_{\mathcal{R}} := \xrightarrow{\mathcal{Q}}_{\mathcal{R}}$. We suppose $\bot \in \mathcal{F}$ is a constructor symbol not occurring in $\mathcal{R}$. As the normal forms of $\mathcal{Q}$ coincide with Val, $\xrightarrow{\mathsf{v}}_{\mathcal{R}}$ is the restriction of $\xrightarrow{\mathsf{i}}_{\mathcal{R}}$, where arguments need to be values instead of normal forms of $\mathcal{R}$. From Lemma 17 and 19 we derive an embedding of root steps $\xrightarrow{\mathsf{v}}^{\varepsilon}_{\mathcal{R}}$.

Now, suppose the step $s \xrightarrow{\mathsf{v}}_{\mathcal{R}} t$ takes place below the root. Observe that $\pi(s) \neq \pi(t)$ need not hold in general. Thus we cannot hope to prove $\mathsf{N}_\pi(s) \blacktriangleright_k \mathsf{N}_\pi(t)$. However, we have the following stronger result.

Lemma 20. *There exists a uniform $k \in \mathbb{N}$ (depending only on $\mathcal{R}$) such that if $\mathcal{R} \subseteq >^{\pi}_{\mathsf{pop}*}$ holds then $s \xrightarrow{\mathsf{v}}^{\varepsilon}_{\mathcal{R}} t$ implies $\mathsf{N}_\pi(s) \blacktriangleright_k \mathsf{N}_\pi(t)$. Moreover, if $\mathcal{R} \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ holds then $s \xrightarrow{\mathsf{v}}_{\mathcal{R}} t$ implies $\mathsf{N}_\pi(s) \underset{\sim}{\blacktriangleright}_k \mathsf{N}_\pi(t)$.*

Proof. We consider the first half of the assertion. Suppose $\mathcal{R} \subseteq >^{\pi}_{\mathsf{pop}*}$ and $s \xrightarrow{\mathsf{v}}^{\varepsilon}_{\mathcal{R}} t$, that is for some rule $f(l_1, \dots, l_n) \to r \in \mathcal{R}$ and substitution $\sigma : \mathcal{V} \to \mathsf{Val}$ we have $s = f(l_1\sigma, \dots, l_n\sigma)$ and $t = r\sigma$. Depending on whether π collapses f, the property either directly follows from Lemma 17 or is a consequence of Lemma 19(2).

In order to conclude the second half of the assertion, one performs induction on the rewrite context. In addition, one shows that for the special case $\mathsf{S}_\pi(s) \approx \mathsf{S}_\pi(t)$, still $\|\pi(s)\| \geqslant \|\pi(t)\|$ holds. From this the lemma follows. □

For constructor TRSs, we can simulate $\xrightarrow{\mathsf{i}}_{\mathcal{R}}$ using $\xrightarrow{\mathsf{v}}_{\mathcal{R}}$. We extend $\mathcal{R}$ with suitable rules $\Phi(\mathcal{R})$, which replace normal forms that are not values by some constructor symbol. To simplify the argument we re-use the symbol $\bot$ from above. We define the TRS $\Phi(\mathcal{R})$ as

$$\Phi(\mathcal{R}) := \{f(t_1, \dots, t_n) \to \bot \mid f(t_1, \dots, t_n) \in \mathsf{NF}(\mathcal{R}) \cap \mathcal{T}(\mathcal{F}) \text{ and } f \in \mathcal{D}\} .$$

Moreover, we define $\phi_{\mathcal{R}}(t) := t{\downarrow}_{\Phi(\mathcal{R})}$. Observe that $\phi_{\mathcal{R}}(\cdot)$ is well-defined since $\Phi(\mathcal{R})$ is confluent and terminating.

Lemma 21. *Let $\mathcal{R} \cup \mathcal{S}$ be a constructor TRS. Define $\mathcal{S}' := \mathcal{S} \cup \Phi(\mathcal{R} \cup \mathcal{S})$. For $s \in \mathcal{T}(\mathcal{F})$,*

$$s \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}} t \quad \textit{implies} \quad \phi_{\mathcal{R}\cup\mathcal{S}}(s) \xrightarrow{\mathsf{v}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}'} \phi_{\mathcal{R}\cup\mathcal{S}}(t) \ ,$$

where $\xrightarrow{\mathsf{v}}_{\mathcal{R}/\mathcal{S}'}$ abbreviates $\xrightarrow{\mathsf{v}}{}^{*}_{\mathcal{S}'} \cdot \xrightarrow{\mathsf{v}}_{\mathcal{R}} \cdot \xrightarrow{\mathsf{v}}{}^{*}_{\mathcal{S}'}$.

Proof. It is easy to see that $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$ implies $\phi_{\mathcal{R}}(s) \xrightarrow{\mathsf{v}}_{\mathcal{R}} \cdot \xrightarrow{\mathsf{v}}{}^{!}_{\Phi(\mathcal{R})} \phi_{\mathcal{R}}(t)$. Suppose $s \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}} t$, then there exist ground terms u and v such that $s \xrightarrow{\mathsf{i}}{}^{*}_{\mathcal{S}} u \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}} v \xrightarrow{\mathsf{i}}{}^{*}_{\mathcal{S}} t$. Let $\phi(t) := \phi_{\mathcal{R}\cup\mathcal{S}}(t)$. From the above, $\phi(s) \xrightarrow{\mathsf{v}}{}^{*}_{\mathcal{S}'} \phi(u) \xrightarrow{\mathsf{v}}{}^{\varepsilon}_{\mathcal{R}} \cdot \xrightarrow{\mathsf{v}}{}^{*}_{\mathcal{S}'} \phi(v) \xrightarrow{\mathsf{v}}{}^{*}_{\mathcal{S}'} \phi(t)$ follows as desired. ☐

Suppose $\mathcal{R} \subseteq >^{\pi}_{\mathsf{pop}*}$ and $\mathcal{S} \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ holds. Together with Lemma 20, the above simulation establishes the promised embedding of $\xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}}$ into $\blacktriangleright_k$.

Lemma 22. *Let $\mathcal{R} \cup \mathcal{S}$ be a constructor TRS, and suppose $\mathcal{R} \subseteq >^{\pi}_{\mathsf{pop}*}$ and $\mathcal{S} \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ hold. Then for k depending only on $\mathcal{R}$ and $\mathcal{S}$, we have for $s \in \mathcal{T}(\mathcal{F})$,*

$$s \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}} t \quad \textit{implies} \quad \mathsf{N}_{\pi}(\phi(s)) \blacktriangleright^{+}_{k} \mathsf{N}_{\pi}(\phi(t)) \ .$$

Proof. Consider a step $s \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}} t$ and set $\phi(t) := \phi_{\mathcal{R}\cup\mathcal{S}}(t)$. By Lemma 21 there exist terms u and v such that $\phi(s) \xrightarrow{\mathsf{v}}{}^{*}_{\mathcal{S}\cup\Phi(\mathcal{R}\cup\mathcal{S})} u \xrightarrow{\mathsf{v}}{}^{\varepsilon}_{\mathcal{R}} v \xrightarrow{\mathsf{v}}{}^{*}_{\mathcal{S}\cup\Phi(\mathcal{R}\cup\mathcal{S})} \phi(t)$. Since $\mathcal{R} \subseteq >^{\pi}_{\mathsf{pop}*}$ holds, by Lemma 20 $\mathsf{N}_{\pi}(u) \blacktriangleright_{k_1} \mathsf{N}_{\pi}(v)$ follows. Moreover from $\mathcal{S} \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ together with Lemma 20 we conclude that $r_1 \xrightarrow{\mathsf{v}}_{\mathcal{S}\cup\Phi(\mathcal{R}\cup\mathcal{S})} r_2$ implies $\mathsf{N}_{\pi}(r_1) \mathrel{\underset{\sim}{\blacktriangleright}}_{k_2} \mathsf{N}_{\pi}(r_2)$. Here we use the easily verified fact that steps using $\Phi(\mathcal{R} \cup \mathcal{S})$ are embeddable into $\mathrel{\underset{\sim}{\blacktriangleright}}_{k_2}$. In both cases k_1 and k_2 depend only on $\mathcal{R}$ and $\mathcal{S}$ respectively; set $k := \max\{k_1, k_2\}$. In sum we have $\mathsf{N}_{\pi}(\phi(s)) \mathrel{\underset{\sim}{\blacktriangleright}}^{*}_{k} \mathsf{N}_{\pi}(u) \blacktriangleright_k \mathsf{N}_{\pi}(v) \mathrel{\underset{\sim}{\blacktriangleright}}^{*}_{k} \mathsf{N}_{\pi}(\phi(t))$. It is an easy to see that $\blacktriangleright_k \cdot \approx \subseteq \blacktriangleright_k$ and $\approx \cdot \blacktriangleright_k \subseteq \blacktriangleright_k$ holds. Hence the lemma follows. ☐

Theorem 23. *Let $\mathcal{R} \cup \mathcal{S}$ be a constructor TRS, and suppose $\mathcal{R} \subseteq >^{\pi}_{\mathsf{pop}*}$ and $\mathcal{S} \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ holds. Then there exists a polynomial p depending only $\mathcal{R} \cup \mathcal{S}$ such that for any basic term t, $\mathrm{dl}(t, \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}}) \leqslant p(|t|)$.*

Proof. Assume $t \notin \mathsf{NF}(\mathcal{R}\cup\mathcal{S})$, otherwise $\mathrm{dl}(t, \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}})$ is trivially bounded. Without loss of generality, we assume that t is ground. As t is a basic term: $\phi_{\mathcal{R}\cup\mathcal{S}}(t) = t$. From Lemma 22 we infer (for some k) $\mathrm{dl}(t, \xrightarrow{\mathsf{i}}{}^{\varepsilon}_{\mathcal{R}/\mathcal{S}}) \leqslant \mathsf{G}_k(\mathsf{N}_{\pi}(\phi_{\mathcal{R}\cup\mathcal{S}}(t))) = \mathsf{G}_k(\mathsf{N}_{\pi}(t))$, such that the latter is polynomially bounded in $|t|$ and the polynomial only depends on k, cf. Lemma 16. Note that k depends only on $\mathcal{R} \cup \mathcal{S}$. ☐

Suppose $\mathcal{R}$ is a constructor TRS, and let $\mathcal{P}$ denote the set of weak innermost dependency pairs. For the moment, suppose that all compound symbols of $\mathcal{P}$ are nullary. Provided that $\mathcal{P}$ is non-duplicating and $\mathcal{U}(\mathcal{P})$ compatible with some SLI, as a consequence of the above theorem paired with Proposition 7, the inclusions $\mathcal{P} \subseteq >^{\pi}_{\mathsf{pop}*}$ and $\mathcal{U}(\mathcal{P}) \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ certify that $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}$ is polynomially bounded. Observe that for the application of $>^{\pi}_{\mathsf{pop}*}$ and $\gtrsim^{\pi}_{\mathsf{pop}*}$ in the context of $\mathcal{P}$ and $\mathcal{U}(\mathcal{P})$, we alter Definitions 10 and 11 such that $f \in \mathcal{D}^{\sharp}$ is demanded.

Example 24 (Example 14 continued). Reconsider the TRS $\mathcal{R}_{\mathsf{bits}}$, and let $\mathcal{P}$ denote $\mathsf{WIDP}(\mathcal{R}_{\mathsf{bits}})$ as drawn in Example 5. By taking the SLI $\mathcal{A}$ with $0_{\mathcal{A}} = 0$, $\mathsf{s}_{\mathcal{A}}(x) = x+1$ and $\mathsf{half}_{\mathcal{A}}(x) = x+1$ we obtain $\mathcal{U}(\mathcal{P}) \subseteq >_{\mathcal{A}}$ and moreover, observe that $\mathcal{P}$ is both non-duplicating and contains only nullary compound symbols. In Example 14 we have seen that $\mathcal{P} \subseteq >^{\pi}_{\mathsf{pop}*}$ holds. Similarly, $\mathcal{U}(\mathsf{WIDP}(\mathcal{R}_{\mathsf{bits}})) \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ can easily be shown. From the above observation we thus conclude a polynomial runtime complexity of $\mathcal{R}_{\mathsf{bits}}$.

The assumption that all compound symbols from $\mathcal{P}$ need to be nullary is straightforward to lift, but technical. Hence, we do not provide a complete proof here, but only indicate the necessary changes, see [18] for the formal construction.

Note that in the general case, it does not suffice to embed root steps of $\mathcal{P}$ into $\blacktriangleright_k$, rather we have to embed steps of form $C[s^\sharp_1, \dots, s^\sharp_i, \dots, s^\sharp_n] \xrightarrow{\mathsf{v}}_{\mathcal{P}} C[s^\sharp_1, \dots, t^\sharp_i, \dots, s^\sharp_n]$ with C being a context built from compound symbols. As first measure we require that the argument filtering π is *safe* [5], that is $\pi(\mathsf{c}) = [1, \dots, n]$ for each compound symbol c of arity n. Secondly, we adapt the predicative interpretation N_π in such a way that compound symbols are interpreted as sequences, and their arguments by the interpretation N_π. This way, the renewed embedding requires $\mathsf{N}_\pi(s^\sharp_i) \blacktriangleright_k \mathsf{N}_\pi(t^\sharp_i)$ instead of $\mathsf{S}_\pi(s^\sharp_i) \blacktriangleright_k \mathsf{S}_\pi(t^\sharp_i)$.

Theorem 25. *Let $\mathcal{R}$ be a constructor TRS, and let $\mathcal{P}$ denote the set of weak innermost dependency pairs. Assume $\mathcal{P}$ is non-duplicating, and suppose $\mathcal{U}(\mathcal{P}) \subseteq >_{\mathcal{A}}$ for some SLI $\mathcal{A}$. Let π be a safe argument filtering. If $\mathcal{P} \subseteq >^{\pi}_{\mathsf{pop}*}$ and $\mathcal{U}(\mathcal{P}) \subseteq \gtrsim^{\pi}_{\mathsf{pop}*}$ then the innermost runtime complexity $\mathrm{rc}^{\mathsf{i}}_{\mathcal{R}}$ induced is polynomially bounded.*

Above it is essential that $\mathcal{R}$ is a constructor TRS. This even holds if POP* is applied directly.

Example 26. Consider the TRS $\mathcal{R}_{\mathsf{exp}}$ below:

$$\mathsf{exp}(x) \to \mathsf{e}(\mathsf{g}(x)) \qquad \mathsf{e}(\mathsf{g}(\mathsf{s}(x))) \to \mathsf{dp}_1(\mathsf{g}(x)) \qquad \mathsf{g}(0) \to 0$$
$$\mathsf{dp}_1(x) \to \mathsf{dp}_2(\mathsf{e}(x), x) \qquad \mathsf{dp}_2(x, y) \to \mathsf{pr}(x, \mathsf{e}(y))$$

The above rules are oriented directly by $>_{\mathsf{pop}*}$ induced by safe and $\succsim$ such that: (i) the argument position of g and exp are normal, the remaining argument positions are safe, and (ii) $\mathsf{exp} \succ \mathsf{g} \succ \mathsf{dp}_1 \succ \mathsf{dp}_2 \succ \mathsf{e} \succ \mathsf{pr} \succ 0$. On the other hand, $\mathcal{R}_{\mathsf{exp}}$ admits at least exponential innermost runtime-complexity, as for instance $\mathsf{exp}(\mathsf{s}^n(0))$ normalizes in exponentially (in n) many innermost rewrite steps.

We adapt the definition of $>_{\mathsf{pop}*}$ in the sense that we refine the notion of defined function symbols as follows. Let $\mathcal{G}_{\mathcal{C}}$ denote the least set containing $\mathcal{C}$ and all symbols appearing in arguments to left-hand sides in $\mathcal{R}$. Moreover, set $\mathcal{G}_{\mathcal{D}} := \mathcal{F} \setminus \mathcal{G}_{\mathcal{C}}$ and set $\mathsf{Val} := \mathcal{T}(\mathcal{G}_{\mathcal{C}}, \mathcal{V})$. Then in order to extend Theorem 25 to non-constructor TRS it suffices to replace $\mathcal{D}$ by $\mathcal{G}_{\mathcal{D}}$ and $\mathcal{C}$ by $\mathcal{G}_{\mathcal{C}}$ in all above given definitions and arguments (see [17] for the formal construction). Thus the next theorem follows easily from combining Proposition 9 and Theorem 25. This theorem can be extended so that in each path different termination techniques (inducing polynomial runtime complexity) are employed, see [6] and Section 6.

Theorem 27. *Let $\mathcal{R}$ be a TRS. Let $\mathcal{G}$ denote the weak innermost dependency graph, and let $\mathcal{F} = \mathcal{G}_\mathcal{D} \uplus \mathcal{G}_\mathcal{C}$ be separated as above. Suppose for every path $(\mathcal{P}_1, \ldots, \mathcal{P}_n)$ in $\mathcal{G}/_\equiv$ there exists an SLI $\mathcal{A}$ and a pair $(\gtrsim^\pi_{\mathsf{pop*}}, >^\pi_{\mathsf{pop*}})$ based on a safe argument filtering π such that (i) $\mathcal{U}(\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_n) \subseteq >_\mathcal{A}$ (ii) $\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_{n-1} \cup \mathcal{U}(\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_n) \subseteq \gtrsim^\pi_{\mathsf{pop*}}$, and (iii) $\mathcal{P}_n \subseteq >^\pi_{\mathsf{pop*}}$ holds. Then the innermost runtime complexity $\mathrm{rc}^{\mathsf{i}}_\mathcal{R}$ induced is polynomially bounded.*

The next theorem establishes that POP* in conjunction with (weak innermost) dependency pairs induces polytime computability of the function described through the analysed TRS. We kindly refer the reader to [18] for the proof.

Theorem 28. *Let $\mathcal{R}$ be an orthogonal, S-sorted and completely defined constructor TRS such that the underlying signature is simple. Let $\mathcal{P}$ denote the set of weak innermost dependency pairs. Assume $\mathcal{P}$ is non-duplicating, and suppose $\mathcal{U}(\mathcal{P}) \subseteq >_\mathcal{A}$ for some SLI $\mathcal{A}$. If $\mathcal{P} \subseteq >^\pi_{\mathsf{pop*}}$ and $\mathcal{U}(\mathcal{P}) \subseteq \gtrsim^\pi_{\mathsf{pop*}}$ then the functions computed by $\mathcal{R}$ are computable in polynomial time.*

Here *simple* signature [19] essentially means that the size of any constructor term depends polynomially on its depth. Such a restriction is always necessary in this context, compare [18] and e.g. [19]. This restriction is also responsible for the introduction of sorts.

6 Experimental Results

All described techniques have been incorporated into the *Tyrolean Complexity Tool* $\mathsf{T_CT}$, an open source complexity analyser[1]. We performed tests on two testbeds: **T** constitutes of the 1394 examples from the Termination Problem Database Version 5.0.2 used in the runtime complexity category of the termination competition 2008[2]. Moreover, testbed **C** is the restriction of testbed **T** to constructor TRSs (638 in total). 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. We orient TRSs using $>^\pi_{\mathsf{pop*}}$ by encoding the constraints on precedence and so forth in *propositional logic* (cf. [17] for details), employing MiniSat [20] for finding satisfying assignments. In a similar spirit, we check compatibility with SLIs via translations to SAT. In order to derive an estimated dependency graph, we use the function ICAP (cf. [21]).

Experimental findings are summarised in Table 1.[3] In each column, we highlight the total on yes-, maybe- and timeout-instances. Furthermore, we annotate

[1] Available at `http://cl-informatik.uibk.ac.at/software/tct`.
[2] See `http://termcomp.uibk.ac.at`.
[3] See `http://cl-informatik.uibk.ac.at/~zini/rta09` for extended results.

		polynomial path orders			dependency graphs mixed			
		DIRECT	WIDP	WIDG	P	PP	M	MP
T	Yes	46/0.03	69/0.09	80/0.07	198/0.54	198/0.51	200/0.63	207/0.48
	Maybe	1348/0.04	1322/0.10	1302/0.14	167/0.77	170/0.82	142/0.61	142/0.63
	Timeout	0	3	12	1029	1026	1052	1045
C	Yes	40/0.03	48/0.08	55/0.05	99/0.40	100/0.38	98/0.26	105/0.23
	Maybe	598/0.05	587/0.10	576/0.13	143/0.72	146/0.77	119/0.51	119/0.54
	Timeout	0	3	7	396	392	421	414

Table 1. Experimental Results

average times in seconds. In the first three columns we contrast POP* as direct technique to POP* as base to (weak innermost) dependency pairs. I.e., the columns WIDP and WIDG show results concerning Proposition 7 together with Theorem 25 or Theorem 27 respectively.

In the remaining four columns we assess the power of Proposition 7 and 9 in conjunction with different base orders, thus verifying that the use of POP* in this context is independent to existing techniques. Column P asserts that the different paths are handled by *linear and quadratic restricted interpretations* [5]. In column PP, in addition POP* is employed. Similar, in column M *restricted matrix interpretations* (that is matrix interpretations [22], where constructors are interpreted by triangular matrices) are used to handle different paths. Again column MP extends column M with POP*. Note that all methods induce polynomial innermost runtime complexity.

Table 1 reflects that the integration of POP* in the context of (weak) dependency pairs, significantly extends the direct approach. Worthy of note, the extension of [2] with quasi-precedences alone gives 5 additional examples. As advertised, POP* is incredibly fast in all settings. Consequently, as evident from the table, polynomial path orders team well with existing techniques, without affecting overall performance: note that due to the addition of POP* the number of timeouts is reduced.

7 Conclusion

In this paper we study the runtime complexity of rewrite systems. We extend polynomial path orders with the scheme of predicative recursion and parameter substitution. If the conditions of our main result are met, we can conclude the innermost polynomial runtime complexity of the studied term rewrite system. Moreover, we obtain an alternative characterization 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.

References

1. Hofbauer, D., Lautemann, C.: Termination proofs and the length of derivations. In: Proc. 3rd RTA. Volume 355 of LNCS. (1989) 167–177
2. Avanzini, M., Moser, G.: Complexity analysis by rewriting. In: Proc. 9th FLOPS. Volume 4989 of LNCS. (2008) 130–146
3. Simmons, H.: The realm of primitive recursion. ARCH **27** (1988) 177–188
4. Bellantoni, S., Cook, S.: A new recursion-theoretic characterization of the polytime functions. CC **2**(2) (1992) 97–110
5. Hirokawa, N., Moser, G.: Automated complexity analysis based on the dependency pair method. In: Proc. 4th IJCAR. Volume 5195 of LNCS. (2008) 364–380
6. Hirokawa, N., Moser, G.: Complexity, graphs, and the dependency pair method. In: Proc. 15th LPAR. Volume 5330 of LNCS. (2008) 667–681
7. Fuhs, C., Giesl, J., Middeldorp, A., Schneider-Kamp, P., Thiemann, R., Zankl, H.: SAT solving for termination analysis with polynomial interpretations. In: Proc. 10th SAT. Volume 4501 of LNCS. (2007) 340–354
8. Geser, A.: Relative Termination. PhD thesis, University of Passau, Faculty for Mathematics and Computer Science (1990)
9. Bonfante, G., Marion, J.Y., Moyen, J.Y.: Quasi-interpretations: A way to control resources. TCS (2009) To appear.
10. Marion, J.Y., Péchoux, R.: Characterizations of polynomial complexity classes with a better intensionality. In: Proc. 10th PPDP, ACM (2008) 79–88
11. Arai, T., Moser, G.: Proofs of termination of rewrite systems for polytime functions. In: Proc. 25th FSTTCS. Volume 3821 of LNCS. (2005) 529–540
12. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press (1998)
13. Terese: Term Rewriting Systems. Volume 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (2003)
14. Thiemann, R.: The DP Framework for Proving Termination of Term Rewriting. PhD thesis, University of Aachen, Department of Computer Science (2007)
15. Arts, T., Giesl, J.: Termination of term rewriting using dependency pairs. TCS **236**(1–2) (2000) 133–178
16. Avanzini, M., Moser, G., Schnabl, A.: Automated implicit computational complexity analysis (system description). In: Proc. 4th IJCAR. Volume 5195 of LNCS. (2008) 132–138
17. Avanzini, M.: 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`.
18. Avanzini, M., Moser, G.: Dependency pairs and polynomial path orders. CoRR **ab/cs/0904.0981** (2009) Available at `http://www.arxiv.org/`.
19. Marion, J.Y.: Analysing the implicit complexity of programs. IC **183** (2003) 2–18
20. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Proc. 6th SAT. Volume 2919 of LNCS. (2003) 502–518
21. Giesl, J., Thiemann, R., Schneider-Kamp, P.: Proving and disproving termination of higher-order functions. In: Proc. 5th FroCoS. Volume 4501 of LNCS. (2005) 340–354
22. Endrullis, J., Waldmann, J., Zantema, H.: Matrix interpretations for proving termination of term rewriting. JAR **40**(2–3) (2008) 195–220