The Derivational Complexity Induced by the Dependency Pair Method*

Georg Moser and Andreas Schnabl

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

Abstract. We study the derivational complexity induced by the (basic) dependency pair method. Suppose the derivational complexity induced by a termination method is closed under elementary functions. We show that the derivational complexity induced by the dependency pair method based on this termination technique is the same as for the direct technique. Therefore, the derivational complexity induced by the dependency pair method based on lexicographic path orders or multiset path orders is multiple recursive or primitive recursive, respectively. Moreover for the dependency pair method based on Knuth-Bendix orders, we obtain that the derivational complexity function is majorised by the Ackermann function. These characterisations are essentially optimal.

1 Introduction

In order to assess the complexity of a terminating term rewrite system (TRS for short) it is natural to look at the maximal length of derivation sequences, as suggested by Hofbauer and Lautemann in [1]. More precisely, the *derivational complexity function* with respect to a terminating TRS $\mathcal{R}$ relates the length of the longest derivation sequence to the size of the initial term. For direct termination methods a considerable number of results establish essentially optimal upper bounds on the growth rate of the derivational complexity function. See e.g. [2,3] for recent results in this direction. However, for transformation techniques like semantic labelling [4] or the dependency pair method [5] the situation changes drastically. Apart from the trivial case of labelling with finite models, only partial results are known. With respect to semantic labelling, [6] establishes bounds on the derivation length of TRS, when natural numbers are used as labels and termination is shown via the Knuth-Bendix order (KBO). And recently in [7,8] the derivation length induced by the basic dependency pair method is investigated. Still in both cases only restricted variants of semantic labelling or the dependency pair method could be analysed, compare [6,7,8].

In this paper we investigate the derivational complexity induced by the basic dependency pair method based on reasonably strong base orders. Suppose the class of derivational complexity functions induced by a direct termination method is closed under elementary functions. Then we show that the derivational

* This research is partly supported by FWF (Austrian Science Fund) project P20133.

complexity induced by the dependency pair method based on this termination technique is the same as for the direct technique. More precisely we show that the derivational complexity of a TRS whose termination is established via the dependency pair method combined with some base order is triple exponential in the derivational complexity induced by the base order directly. Moreover, we present an example which shows that at least two of the three exponentials in our upper bound can actually be reached.

It should be emphasised that the notion of dependency pair method studied here amounts to the original technique as introduced by Arts and Giesl [5] (see also [9]). Consider the following TRS $\mathcal{R}_1$ taken from [10]:

$$1\colon\quad (x \times y) \times z \to x \times (y \times z) \qquad 3\colon\ (x + y) \times z \to (x \times z) + (y \times z)$$
$$2\colon\ z \times (x + \mathsf{f}(y)) \to \mathsf{g}(z, y) \times (x + \mathsf{a})$$

Due to rule 2, termination of $\mathcal{R}_1$ cannot be concluded by the lexicographic path order (LPO), cf. [10]. On the other hand, termination follows easily by the dependency pair method based on LPO, if we use argument filtering.

The gist of our result is that for this standard application of the dependency pair method the derivational complexity induced by LPO directly (which is multiple recursive, cf. [11]) bounds the derivation lengths admitted by the investigated TRS $\mathcal{R}_1$. From this we can conclude that the derivational complexity function of $\mathcal{R}_1$ is multiple recursive. Analogous results hold if we employ the multiset path order (MPO) or KBO as base order. Moreover the thus obtained upper bounds are still tight, which essentially follows from the tight characterisation of the derivational complexity by the indicated base orders [12,11,13].

Note the challenges of such an investigation: In order to estimate the derivation length of $\mathcal{R}_1$ we only consider the derivation length induced by the base order. This implies that we use an upper bound on the maximal number of dependency pair steps to bound the length of derivations. It remains open to what extent such a result holds in general, i.e., beyond the basic dependency pair method. The challenge of such an endeavour is most prominent if we allow an iterative use of the dependency pair transformation as for example in the recursive SCC algorithm (see [9]) or the dependency pair framework (see [14]). It is well-known that two iterations of the recursive SCC algorithm based on the subterm criterion (see [15]) suffice to show termination of (the standard formulation of) the Ackermann function. See also [16] for a like minded example. Clearly in this context a triple exponential function is by far not sufficient to bound the difference between the derivational complexity of the TRS and the derivational complexity induced by the base method directly.

The rest of this paper is organised as follows. In Section 2 we present basic notions and starting points of the paper. Section 3 introduces suitable notions to trace an *implicit dependency pair derivation* in a given derivation over a TRS. Our main result is proved in Section 4, while Section 5 presents the above mentioned example on the lower bound. Finally, we conclude in Section 6.

2 Dependency Pairs

We assume familiarity with the basics of term rewriting, see [17,18]. Below we recall the bare essentials of the basic dependency pair method as put forward in [5], but at least nodding acquaintance with [5] or [9] will prove helpful.

Let $\mathcal{V}$ denote a countably infinite set of variables and $\mathcal{F}$ a signature. The set of terms over $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F},\mathcal{V})$. The (proper) subterm relation is denoted as $\trianglelefteq$ ($\triangleleft$). The *root symbol* (denoted as $\mathsf{rt}(t)$) of a term t is either t itself, if $t \in \mathcal{V}$, or the symbol f, if $t = f(t_1,\ldots,t_n)$. The set of *positions* $\mathcal{P}\mathsf{os}(t)$ of a term t is defined as usual. We write $p \leqslant q$ ($p < q$) to denote that p is a (proper) prefix of q, and $p \parallel q$ if neither $p \leqslant q$ nor $q \leqslant p$. The subterm of t at position p is denoted as $t|_p$. We write $\mathcal{P}\mathsf{os}_{\mathcal{F}}(t)$ ($\mathcal{P}\mathsf{os}_{\mathcal{V}}(t)$) for the set of positions p such that $\mathcal{F}$ ($\mathcal{V}$) contains $\mathsf{rt}(t_p)$. The *size* $|t|$ and the *depth* $\mathsf{dp}(t)$ of a term t are defined as usual (e.g., $|\mathsf{f}(\mathsf{a},x)| = 3$ and $\mathsf{dp}(\mathsf{f}(\mathsf{a},x)) = 1$). To simplify the exposition, we often confuse terms and their tree representations. I.e., we call a maximal set of positions B in a term t such that for no $q, q' \in B$, we have $q \parallel q'$, a *branch* of t.

Let $\mathcal{R}$ be a finite TRS over $\mathcal{F}$. We write $\to_{\mathcal{R}}$ (or simply $\to$) for the induced rewrite relation. If we wish to indicate the redex position p and the applied rewrite rule $l \to r$ in a reduction from s to t, we write $s \to_{p,l\to r} t$. The set of defined function symbols is denoted as $\mathcal{D}$, while the constructor symbols are collected in $\mathcal{C}$. The n-fold composition of $\to$ is denoted as $\to^n$ and the *derivation length* of a term s with respect to a finitely branching, well-founded binary relation $\to$ on terms is defined as $\mathsf{dl}(s,\to) := \max\{n \mid \exists t\ s \to^n t\}$. The *derivational complexity function* of $\mathcal{R}$ is defined as: $\mathsf{dc}_{\mathcal{R}}(n) = \max\{\mathsf{dl}(t,\to_{\mathcal{R}}) \mid |t| \leqslant n\}$.

In analogy to $\mathsf{dc}_{\mathcal{R}}$ we define functions tracing the depth or size. The *potential depth* of a term s with respect to $\to$ is defined as follows: $\mathsf{pdp}(s,\to) := \max\{\mathsf{dp}(t) \mid s \to^* t\}$ and the induced *depth growth function* (with respect to $\mathcal{R}$) is defined as $\mathsf{dpg}_{\mathcal{R}}(n) := \max\{\mathsf{pdp}(t,\to_{\mathcal{R}}) \mid |t| \leqslant n\}$. The *potential size* $\mathsf{psz}(s,\to)$ of a term s and the *size growth function* $\mathsf{szg}_{\mathcal{R}}(n)$ are defined similarly.

We recall the central notions of the dependency pair method, see [5,9]. Let t be a term. We set $t^\sharp := t$ if $t \in \mathcal{V}$, and $t^\sharp := f^\sharp(t_1,\ldots,t_n)$ if $t = f(t_1,\ldots,t_n)$. Here $f^\sharp$ is a new n-ary function symbol called *dependency pair symbol*. For a signature $\mathcal{F}$, we define $\mathcal{F}^\sharp = \mathcal{F} \cup \{f^\sharp \mid f \in \mathcal{F}\}$. The set $\mathsf{DP}(\mathcal{R})$ of *dependency pairs* of a TRS $\mathcal{R}$ is defined as $\{l^\sharp \to u^\sharp \mid l \to r \in \mathcal{R}, u \trianglelefteq r, \mathsf{rt}(u) \in \mathcal{D}, l \not\triangleright u\}$.

Proposition 1 ([5,9]). *A TRS $\mathcal{R}$ is terminating if and only if there exists no infinite derivation of the form $t_1^\sharp \to_{\mathcal{R}}^* t_2^\sharp \to_{\mathsf{DP}(\mathcal{R})} t_3^\sharp \to_{\mathcal{R}}^* \ldots$ such that for all $i > 0$, $t_i^\sharp$ is terminating with respect to $\mathcal{R}$.*

Proposition 1 gives rise to the *dependency pair complexity function*:

$$\mathsf{DPc}_{\mathcal{R}}(n) := \max\{\mathsf{dl}(t^\sharp, \to_{\mathsf{DP}(\mathcal{R})/\mathcal{R}}) \mid |t| \leqslant n\}\ ,$$

where we write $\to_{\mathsf{DP}(\mathcal{R})/\mathcal{R}}$ for $\to_{\mathcal{R}}^* \cdot \to_{\mathsf{DP}(\mathcal{R})} \cdot \to_{\mathcal{R}}^*$, cf. [19]. Now, we fix the notion of *basic* dependency pair method. An *argument filtering* (for a signature $\mathcal{F}$) is a mapping π that assigns to every n-ary function symbol $f \in \mathcal{F}$ an argument position $i \in \{1,\ldots,n\}$ or a (possibly empty) list $[i_1,\ldots,i_m]$ of argument positions with $1 \leqslant i_1 < \cdots < i_m \leqslant n$. The signature $\mathcal{F}_\pi$ consists of all function symbols f such that $\pi(f)$ is some list $[i_1, \ldots, i_m]$, 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:

$$\pi(t) = \begin{cases} t & \text{if } t \text{ is a variable} \\ \pi(t_i) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } \pi(f) = i \\ f(\pi(t_{i_1}), \ldots, \pi(t_{i_m})) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } \pi(f) = [i_1, \ldots, i_m] \end{cases}$$

A *reduction pair* $(\gtrsim, \succ)$ consists of a rewrite preorder $\gtrsim$ and a compatible well-founded order $\succ$ which is closed under substitutions. Here compatibility means the inclusion $\gtrsim \cdot \succ \cdot \gtrsim \subseteq \succ$.

Proposition 2 ([5,9]). *A TRS $\mathcal{R}$ is terminating if and only if there exist an argument filtering π and a reduction pair $(\gtrsim, \succ)$ such that $\pi(\mathsf{DP}(\mathcal{R})) \subseteq \succ$ and $\pi(\mathcal{R}) \subseteq \gtrsim$.*

Let $\mathcal{R}$ be a terminating TRS. In the sequel we show that the derivational complexity function $\mathsf{dc}_\mathcal{R}$ is bounded triple exponentially in the dependency pair complexity function $\mathsf{DPc}_\mathcal{R}$. For that we mainly bound the depth growth function of $\mathcal{R}$ exponentially in $\mathsf{DPc}_\mathcal{R}$. As the maximal length of a nonlooping derivation is exponentially bounded in the size of the occurring terms and the latter is exponentially bounded in their depth, our result then follows. In order to prove the main step we analyse the shape of a potential derivation over $\mathcal{R} \cup \mathsf{DP}(\mathcal{R})$ in the light of a given $\mathcal{R}$-derivation. This is the purpose of the next section.

3 Progenitor and Progeny

We introduce a specific generalisation of the notion of *descendant* of a position p which we call *progeny*. Recall the definition of descendants (see [18, Chapter 4]). Let $A : s \to_{p', l \to r} t$ be a rewriting step, and let $p \in \mathcal{P}\mathsf{os}(s)$. Then the *descendants of p in t* (denoted by $p \setminus A$) are defined as follows:

$$p \setminus A = \begin{cases} \{p\} & \text{if } p < p' \text{ or } p \parallel p', \\ \{p' q_3 q_2 \mid r|_{q_3} = l|_{q_1}\} & \text{if } p = p' q_1 q_2 \text{ with } q_1 \in \mathcal{P}\mathsf{os}_\mathcal{V}(l), \\ \varnothing & \text{otherwise} \end{cases}$$

In our situation, we also want to keep track of redex positions, not just of positions in the context or the substitution of the rewrite rule. This intuition is cast into the following definition.

Definition 3. *Let $A : s \to_{p', l \to r} t$ be a rewriting step, and let $p \in \mathcal{P}\mathsf{os}(s)$. Then the* progenies of p in t *(denoted by $p \setminus\!\!\setminus A$) are:*

$$p \setminus\!\!\setminus A = \begin{cases} \{p\} & \textit{if } p < p' \textit{ or } p \parallel p', \\ \{p' q_3 q_2 \mid r|_{q_3} = l|_{q_1}\} & \textit{if } p = p' q_1 q_2 \textit{ with } q_1 \in \mathcal{P}\mathsf{os}_\mathcal{V}(l), \\ \{p' q_2 \mid r|_{q_2} = l|_{q_1}\} & \textit{if } p = p' q_1 \textit{ with } q_1 \in \mathcal{P}\mathsf{os}_\mathcal{F}(l) \setminus \{\epsilon\}, \\ \{p q_1 \mid r|_{q_1} \ntrianglelefteq l \wedge q_1 \in \mathcal{P}\mathsf{os}_\mathcal{F}(r)\} & \textit{if } p = p' \end{cases}$$

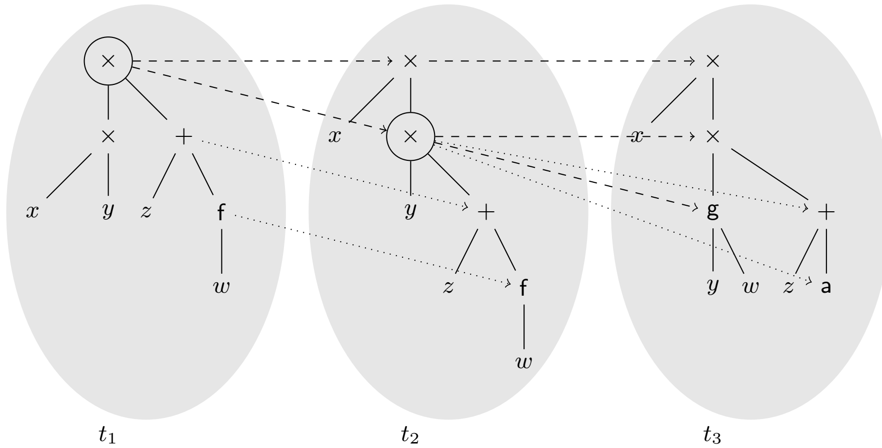


Fig. 1. A derivation, its progeny relation and redex positions.

If $q \in p \setminus\!\!\setminus A$*, then we also say that* p *is a* progenitor of q in s*. We denote the set of progenitors of* q *in* s *by* $A \setminus\!\!\setminus q$*. For a set* $P \subseteq \mathcal{P}\mathsf{os}(s)$*, we define* $P \setminus\!\!\setminus A = \bigcup_{p \in P} p \setminus\!\!\setminus A$*.*

Note that the distinction between the last two cases corresponds to the exclusion of rules $l^\sharp \to u^\sharp$ from $\mathsf{DP}(\mathcal{R})$ where $u \lhd l$, see Section 2.

Example 4. Consider the TRS $\mathcal{R}_1$ from Section 1 and let $t_1 = (x \times y) \times (z + \mathsf{f}(w))$, $t_2 = x \times (y \times (z + \mathsf{f}(w)))$, and $t_3 = x \times (\mathsf{g}(y, w) \times (z + \mathsf{a}))$. We have the derivation $A : t_1 \to t_2 \to t_3$, cf. Figure 1. Redex positions are marked by circles, the progeny relation is marked by dotted and dashed lines (the two kinds of lines will be distinguished in Example 15 below). For clarity progeny relations between variables have been omitted.

Lemma 5. *Let* $\mathcal{R}$ *be a TRS, let* $A : s \to_{\mathcal{R}} t$*, let* $p \in \mathcal{P}\mathsf{os}(s)$*, and let* $q \in \mathcal{P}\mathsf{os}(t)$*. If* $q \in p \setminus\!\!\setminus A$ *and* $\mathsf{rt}(t|_q) \in \mathcal{D}$*, then* $\mathsf{rt}(s|_p) \in \mathcal{D}$ *and* $(s|_p)^\sharp \to^{=}_{\mathcal{R} \cup \mathsf{DP}(\mathcal{R})} (t|_q)^\sharp$*.*

Proof. Suppose that A is $s \to_{p', l \to r} t$. If $p < p'$ or $p \parallel p'$, then by definition, we have $p = q$ and thus $(s|_p)^\sharp \to^{=}_{\mathcal{R}} (t|_q)^\sharp$. On the other hand, if $p = p'$, then there exists $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$ such that $q = p'q_1$. Moreover, $t|_q \ntrianglelefteq s|_p$. By assumption $\mathsf{rt}(t|_q) \in \mathcal{D}$ and thus we obtain $(s|_p)^\sharp \to_{\mathsf{DP}(\mathcal{R})} (t|_q)^\sharp$. Finally, if $p > p'$, then by definition, we have $s|_p = t|_q$. Then (trivially) $(s|_p)^\sharp \to^{=}_{\mathcal{R}} (t|_q)^\sharp$. □

Lemma 6. *Let* $A : s \to_{p', l \to r} t$ *be a rewriting step. Then for every* $q \in \mathcal{P}\mathsf{os}(t)$*, we have* $A \setminus\!\!\setminus q \neq \varnothing$*.*

Proof. If $q < p'$ or $q \parallel p'$, then $A \setminus\!\!\setminus q = \{q\}$. If $q = p'q_1$, $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $t|_q \ntrianglelefteq s|_{p'}$, then $A \setminus\!\!\setminus q = \{p'\}$. If $q = p'q_1$, $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $t|_q \lhd s|_{p'}$, then there is some p_1 such that $s|_{p'p_1} = t|_q$, so $p'p_1 \in A \setminus\!\!\setminus q$. Last, if $q = p'q_1q_2$ and $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(r)$, then there is some p_1 such that $s|_{p'p_1} = t|_{p'q_1}$. Therefore, $p'p_1q_2 \in A \setminus\!\!\setminus q$. □

Definition 3 and Lemmata 5 and 6 extend to derivations in the natural way:

Definition 7. *Let $A : s \to^* t$ be a derivation, and let $p \in \mathcal{P}\mathsf{os}(s)$. Then the* progenies of p in t *(also denoted by $p \mathbin{\backslash\!\!\backslash} A$) are defined as follows:*

- *If A is the empty derivation, then $p \mathbin{\backslash\!\!\backslash} A = \{p\}$.*
- *Otherwise, we can split A into $A_1 : s \to s'$ and $A_2 : s' \to^* t$. Then $p \mathbin{\backslash\!\!\backslash} A = (p \mathbin{\backslash\!\!\backslash} A_1) \mathbin{\backslash\!\!\backslash} A_2$.*

We say p is a progenitor *of q if $p \in A \mathbin{\backslash\!\!\backslash} q$, which holds if $q \in p \mathbin{\backslash\!\!\backslash} A$. Moreover, we have $q \in P \mathbin{\backslash\!\!\backslash} A$ if and only if $q \in p \mathbin{\backslash\!\!\backslash} A$ for some $p \in P$.*

The next lemma follows by straightforward induction using Lemmata 6 and 5.

Lemma 8. *Let $A : s \to^* t$ be a derivation, and let $p \in \mathcal{P}\mathsf{os}(s)$, $q \in \mathcal{P}\mathsf{os}(t)$. Then the set $A \mathbin{\backslash\!\!\backslash} q$ of progenitors of q is not empty. Moreover if $q \in p \mathbin{\backslash\!\!\backslash} A$ with $\mathsf{rt}(t|_q) \in \mathcal{D}$, then $\mathsf{rt}(s|_p) \in \mathcal{D}$ and $(s|_p)^\sharp \to^*_{\mathcal{R} \cup \mathsf{DP}(\mathcal{R})} (t|_q)^\sharp$.*

Using Lemma 8, we can extract derivations over $\mathcal{R} \cup \mathsf{DP}(\mathcal{R})$ from a given derivation in a TRS $\mathcal{R}$ using positions connected by the progeny relation.

Definition 9. *Let $\mathcal{R}$ be a TRS, let $t_1, \ldots, t_n$ be terms, and let $p_1, \ldots, p_n$ be positions in $t_1, \ldots, t_n$, respectively, such that $\mathsf{rt}(t_n|_{p_n}) \in \mathcal{D}$, and for all $1 \leqslant i \leqslant n-1$, we have $A_i : t_i \to_{\mathcal{R}} t_{i+1}$ and $p_{i+1} \in p_i \mathbin{\backslash\!\!\backslash} A_i$. Then we call $A : (t_1|_{p_1})^\sharp \to^*_{\mathcal{R} \cup \mathsf{DP}(\mathcal{R})} (t_n|_{p_n})^\sharp$ the* implicit dependency pair derivation *with respect to $t_1, \ldots, t_n$ and $p_1, \ldots, p_n$. We denote the number of $\mathsf{DP}(\mathcal{R})$-steps in A as $\mathsf{DPI}(A)$.*

Note that Definition 9 is well-defined, due to Lemma 8.

Example 10 (continued from Example 4). The implicit dependency pair derivation with respect to t_1, t_2, t_3 and $\epsilon, 2, 2$ is given as follows:

$$t_1^\sharp \to_{\mathsf{DP}(\mathcal{R}_1)} y \times^\sharp (z + \mathsf{f}(w)) \to_{\mathsf{DP}(\mathcal{R}_1)} \mathsf{g}(y, w) \times^\sharp (z + \mathsf{a}) \,.$$

Lemma 11. *Let $A : s \to_{p', l \to r} t$ be a rewriting step. Let $q, q' \in \mathcal{P}\mathsf{os}(t)$. If $q \leqslant q'$, then for any $p_0 \in A \mathbin{\backslash\!\!\backslash} q$, there exists $p'_0 \in A \mathbin{\backslash\!\!\backslash} q'$ such that $p_0 \leqslant p'_0$.*

Proof. According to Definition 3, there are four cases for q'.

- If $q' < p'$ or $q' \parallel p'$, then also $q < p'$ or $q \parallel p'$. Therefore, $A \mathbin{\backslash\!\!\backslash} q = \{q\}$ and $A \mathbin{\backslash\!\!\backslash} q' = \{q'\}$.
- If $q' = p'q'_1$, $q'_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $r|_{q'_1} \ntriangleleft l$, then either $q < p'$, or $q = p'q_1$, $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $r|_{q_1} \ntriangleleft l$. We have $A \mathbin{\backslash\!\!\backslash} q = \{p_0\}$ and $A \mathbin{\backslash\!\!\backslash} q' = \{p'\}$ with $p_0 = q$ or $p_0 = p'$.
- If $q' = p'q'_1$, $q'_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $r|_{q'_1} \triangleleft l$, then $A \mathbin{\backslash\!\!\backslash} q' = \{p'q'_2 \mid q'_2 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l) \wedge r|_{q'_1} = l|_{q'_2}\}$. Three cases from Definition 3 are applicable for q. Suppose $q < p'$, then $A \mathbin{\backslash\!\!\backslash} q = \{q\}$, and $q < p' \leqslant p'q'_2$ for any $q'_2 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l)$. If $q = p'q_1$, $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $r|_{q_1} \ntriangleleft l$, then $A \mathbin{\backslash\!\!\backslash} q = \{p'\}$. Last, if $q = p'q_1$, $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $r|_{q_1} \triangleleft l$, then $A \mathbin{\backslash\!\!\backslash} q = \{p'q_2 \mid q_2 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l) \wedge r|_{q_1} = l|_{q_2}\}$. We have $q' = qq'_3$, so for any $p'q_2 \in A \mathbin{\backslash\!\!\backslash} q$, also $p'q_2q'_3 \in A \mathbin{\backslash\!\!\backslash} q'$.

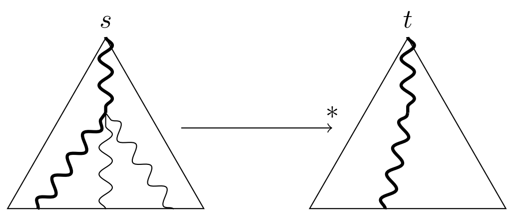
– Otherwise, $q' = p'q'_1q'_2$ with $q'_1 \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(r)$. Then $A \setminus\!\!\setminus q' = \{p'q'_3q'_2 \mid r|_{q'_1} = l|_{q'_3}\}$. Except for $q \parallel p'$, all cases in Definition 3 can happen for q. Suppose $q < p'$, then $A \setminus\!\!\setminus q = \{q\}$, and $q < p' < p'q'_3q'_2$ for any $q'_3 \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(l)$. If $q = p'q_1$, $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $r|_{q_1} \not\lhd l$, then $A \setminus\!\!\setminus q = \{p'\}$. For the next case, suppose $q = p'q_1$, $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$, and $r|_{q_1} \lhd l$. Then $A \setminus\!\!\setminus q = \{p'q_2 \mid q_2 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l) \wedge r|_{q_1} = l|_{q_2}\}$. We have $q' = qq'_4q'_2$, so for any $p'q_2 \in A \setminus\!\!\setminus q$, also $p'q_2q'_4q'_2 \in A \setminus\!\!\setminus q'$. Otherwise, $q = p'q'_1q_2$. Then $A \setminus\!\!\setminus q = \{p'q_3q_2 \mid r|_{q'_1} = l|_{q_3}\}$. We have $q'_2 = q_2q'_4$, hence for any $p'q_3q_2 \in A \setminus\!\!\setminus q$, also $p'q_3q'_2 \in A \setminus\!\!\setminus q'$.

□

Note that each position in a term may have several progenitors:

Example 12. Consider the TRS $\mathcal{R}_2$ consisting of the single rule $\mathsf{f}(x,x) \to g(x)$, and the rewrite step $A : \mathsf{f}(0,0) \to_{\mathcal{R}_2} \mathsf{g}(0)$. Then $A \setminus\!\!\setminus 1 = \{1,2\}$.

We restrict the progenies and progenitors to a single branch in each term. The definition rests on the idea that for a derivation $A\colon s \to^* t$ and a main branch B' in t it is possible to find a *main branch* B in s such that each position $q \in B'$ has a (unique) progenitor in B; see the picture below for an illustration:



In the following definition, the restriction to the leftmost of all candidate positions is arbitrary and can be suitably replaced. Note that its second clause is well-defined by Lemmata 8 and 11.

Definition 13. *Let $A\colon t_1 \to^* t_n$ denote a derivation built up from the rewrite steps $A_i\colon t_i \to t_{i+1}$ for $i = 1,\dots,n-1$. Then the* main branch *of each term in A is inductively defined:*

- *The main branch of t_n is the leftmost branch of maximal length in t_n.*
- *Suppose the main branch of t_{i+1} is denoted as B_{i+1}, $1 \leqslant i \leqslant n-1$. Then consider all branches b in t_i such that for every $q \in B_{i+1}$, the set of progenitors $A_i \setminus\!\!\setminus q$ of q has nonempty intersection with b. The leftmost of these branches is the main branch of t_i, denoted as B_i.*

The next definition specialises progenies and progenitors to the main branch.

Definition 14. *Let $A' : s \to t$ be a rewriting step, let $p \in \mathcal{P}\mathsf{os}(s)$, and let B and B' be branches in s and t. Then the set of* main progenies of p in t *(with respect to A') (denoted as $p \mathbin{\ni}^{B}_{B'} A'$) is defined as follows:*

$$p \mathbin{\ni}^{B}_{B'} A' = \begin{cases} \varnothing & \text{if } p \notin B \\ B' \cap p \setminus\!\!\setminus A' & \text{if } p \in B \end{cases}$$

We naturally extend this definition to derivations, analogous to Definition 7. If the (main) branches B and B' are clear from context, we write $p \supsetdot A'$ instead of $p \supsetdot^{B}_{B'} A'$. If $q \in p \supsetdot A'$, then we also say that p is a main progenitor of q in s *(with respect to A'). We denote the set of main progenitors of q in s by $A' \supsetdot q$. For a set $P \subseteq \mathcal{P}\mathsf{os}(s')$, we define $P \supsetdot A' = \bigcup_{p \in P} p \supsetdot A'$.*

Example 15 (continued from Example 4). Consider the derivation A again. The "central" branch of each term in Figure 1 is its main branch, and the dashed lines denote the main progeny relation.

Lemma 16. *Let $A\colon u \to^* s \to^n t \to^* w$ be a derivation, and denote $A'\colon s \to^n t$. Let $B(s)$ ($B(t)$) denote the main branch of s (t) in A. Then for any $q \in B(t)$, the main progenitor of q in the branch $B(s)$ is unique, i.e., $|A' \supsetdot q| = 1$.*

Proof. By Definition 13, q has at least one main progenitor in s. We show that there exists at most one by induction on n. For $n = 0$ the claim is trivial. Hence assume $n > 0$ and let $A'\colon s \to t' \to^{n-1} t$. Let $B(t')$ denote the main branch in t' with respect to A. By induction hypothesis there exists a unique position q_1 in $B(t')$ such that $(t' \to^{n-1} t) \supsetdot q = \{q_1\}$. Let $A''\colon s \to_{p',l\to r} t'$ denote the first rewrite step in A'. Suppose $q_1 < p'$ or $q_1 \parallel p'$. Then by definition $A'' \setminus\!\!\setminus q_1 = \{q_1\}$. Hence the main progenitor of q in $B(s)$ is unique. On the other hand suppose $q_1 = p'q_2$ with $q_2 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$ such that $r|_{q_2} \ntrianglelefteq l$. Then $A'' \setminus\!\!\setminus q_1 = \{p'\}$ and $A' \supsetdot q$ is a singleton as it should be. Now suppose $q_1 = p'q_2$ with $q_2 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(r)$ such that $r|_{q_2} \triangleleft l$. Then by definition $A'' \setminus\!\!\setminus q_1 = \{p'p_1 \mid p_1 \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l) \wedge l|_{p_1} = r|_{q_2}\}$. Note that $A' \supsetdot q = A'' \setminus\!\!\setminus q_1 \cap B(s)$, which is again a singleton. Finally, if $q_1 = p'q_2q_3$ with $q_2 \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(r)$, then $A'' \setminus\!\!\setminus q_1 = \{p'p_1q_3 \mid p_1 \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(l) \wedge l|_{p_1} = r|_{q_2}\}$. As before, the intersection of the latter set with $B(s)$ is a singleton. Hence the main progenitor of q in $B(s)$ is unique. This concludes the inductive proof. $\square$

Lemma 17. *We assume the same notation as in Lemma 16. For any $p \in B(s)$ such that $\mathsf{rt}(s|_p) \in \mathcal{C} \cup \mathcal{V}$, we have $|p \supsetdot A'| \leqslant 1$, i.e., the number of main progenies for a position, whose root is non-defined is at most 1.*

Proof. By induction on n. It suffices to consider the case $n > 0$, so $A'\colon s \to t' \to^{n-1} t$. Let $A''\colon s \to_{p',l\to r} t'$ denote the first rewrite step in A'. If $p < p'$ or $p \parallel p'$, then $p \supsetdot A'' = \{p\}$. If $p > p'$, then for any $q_1 \in p \setminus\!\!\setminus A''$, we have $s|_p = t'|_{q_1}$, so again, $p \setminus\!\!\setminus A'' \cap B(t')$ is a singleton. In all three cases, the claim follows by induction hypothesis as $\mathsf{rt}(s|_p) = \mathsf{rt}(t|_{q_1})$. This concludes the proof, as the case $p = p'$ is impossible. Otherwise, we derive a contradiction to the assumption that the root of $s|_p$ is not a defined symbol. $\square$

4 Dependency Pairs and Complexity

In this section, we relate the dependency pair complexity and the derivational complexity of a TRS. As mentioned above the main step is to show that the depth growth of a TRS is bounded by a single exponential in its dependency pair complexity. In the sequel, we fix a finite TRS $\mathcal{R}$ and a derivation $A : t_1 \to^* t_n$

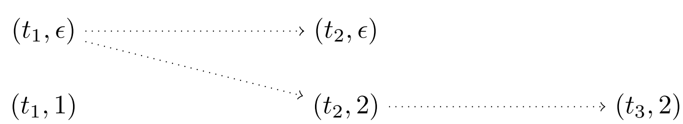
over $\mathcal{R}$ such that $B_1, \ldots, B_n$ denote the main branches with respect to A. We can view the main progeny relation as a graph, called *progenitor graph*. The nodes of a progenitor graph are pairs (t_i, p) representing positions p in terms in A that are directly affected by dependency pair steps. Each edge corresponds to a dependency pair step (and possibly a number of $\mathcal{R}$-steps) in an implicit dependency pair derivation. Each connected component of the progenitor graph is a tree whose height is bounded by the number of dependency pair steps. Using the exponential relationship between the height of this tree and its number of leaves, we bound the depth of the final term in A exponentially in the length of the largest implicit dependency pair derivation, entailing our main result.

Definition 18. *The* progenitor graph *of A is defined as follows.*

- *The nodes are all pairs (t_i, p) such that $p \in B_i$, $\mathsf{rt}(t_i|_p) \in \mathcal{D}$, and either $i = 1$ or the single element of $(t_{i-1} \to t_i) \eth p$ and the redex position in the rewrite step $t_{i-1} \to t_i$ coincide.*
- *There is an edge from (t_i, p) to (t_j, q) whenever $i < j$, $(t_i \to^* t_j) \eth q = \{p\}$, and for all $i \leqslant k < j - 1$, the single element of $(t_k \to^* t_j) \eth q$ and the redex position in the rewrite step $t_k \to t_{k+1}$ do not coincide.*

Note that, due to the definition of the set of nodes in a progenitor graph, the single element of $(t_{j-1} \to t_j) \eth q$ and the redex position in the rewrite step $t_{j-1} \to t_j$ do coincide in the second clause of Definition 18.

Example 19. Consider the derivation A from Example 4 again. Its progenitor graph is shown below:



Lemma 20. *If there is an edge from (t_i, p) to (t_j, q) in G, then there is a derivation $(t_i|_p)^\sharp \to^*_{\mathcal{R}} (t_{j-1}|_{p_1})^\sharp \to_{\mathsf{DP}(\mathcal{R})} (t_j|_q)^\sharp$.*

Proof. By definition, $q \in p \eth (t_i \to^* t_j)$. Therefore, by Lemma 8, we have the implicit dependency pair derivation $A' : (t_i|_p)^\sharp \to^*_{\mathcal{R} \cup \mathsf{DP}(\mathcal{R})} (t_j|_q)^\sharp$. We have $(t_{j-1} \to t_j) \eth q = \{p_1\}$, where by definition p_1 is the redex position of the step $t_{j-1} \to t_j$. Therefore, the last step of A' is a $\mathsf{DP}(\mathcal{R})$-step (see also the last clause of Definition 3). Note that for $i \leqslant k < j - 1$, the single element of $(t_k \to^* t_j) \eth q$ and the redex position in $t_k \to t_{k+1}$ do not coincide. Hence, if there are rewrite steps before the last step, these are $\mathcal{R}$-steps and the lemma follows. $\square$

The next lemma shows, when specialised to the conditions in the first clause of Definition 18, that only nodes which do not contribute to the branching of the progenitor graph, are left out by the definition.

Lemma 21. *Let $p \in B_i$ and $q \in B_j$ such that $i < j$ and $(t_i \to^* t_j) \eth q = \{p\}$. If for all $i \leqslant k \leqslant j - 1$, the single element of $(t_k \to^* t_j) \eth q$ and the redex position in the rewrite step $t_k \to t_{k+1}$ do not coincide, then $p \eth (t_i \to^* t_j) = \{q\}$.*

Proof. We show the lemma by induction on $j-i$. If $i=j$ then the claim trivially holds. Otherwise, the derivation $t_i \to^* t_j$ can be split to $t_i \to t_{i+1} \to^* t_j$. Let p' be the redex position in $t_i \to t_{i+1}$. If $p \parallel p'$, $p < p'$, or $p > p'$, then as in Lemma 17, $|p \partial (t_i \to t_{i+1})| \leqslant 1$, and the lemma follows by induction hypothesis. The last case is again impossible, since by assumption, p and p' do not coincide. □

From now on, let G be the progenitor graph of A. In the next lemmata, we show the properties which allow us to bound $\mathsf{dp}(t_n)$ in the height of G. First, we prove that almost each position in B_n is "covered" by a node in G. Next, we show that each node in G can only cover c positions in B_n, and finally, we show that the branching factor of G is at most c, where $c := \max\{2\} \cup \{\mathsf{dp}(r) \mid l \to r \in \mathcal{R}\}$.

Lemma 22. *For every $q \in B_n$, there either exists $p \in B_1$ such that $\mathsf{rt}(t_1|_p) \in \mathcal{C} \cup \mathcal{V}$ and $A \partial q = \{p\}$, or there exists a node (t_i, p) in G where $q \in p \partial (t_i \to^* t_n)$ and for any successor node (t_j, p_1) of (t_i, p) in G, we have $q \notin p_1 \partial (t_j \to^* t_n)$.*

Proof. By Lemma 16, $A \partial q = \{p\}$ for some $p \in B_1$. If $\mathsf{rt}(t_1|_p) \in \mathcal{C} \cup \mathcal{V}$, the first alternative of the lemma holds. If $\mathsf{rt}(t_1|_p) \in \mathcal{D}$, then $(t_1, p) \in G$. Therefore, there exists a maximal natural number k such that $(t_k, p_2) \in G$ and $q \in p_2 \partial (t_k \to^* t_n)$ for some $p_2 \in B_k$, so the second alternative of the lemma holds for (t_k, p_2). □

Lemma 23. *For every node (t_i, p) in G, there are at most c many positions $q \in B_n$ such that $q \in p \partial (t_i \to^* t_n)$, but for any successor node (t_j, p_1) of (t_i, p), we have $q \notin p_1 \partial (t_j \to^* t_n)$.*

Proof. If there is no $i \leqslant k < n$ such that the redex position of the step $t_k \to t_{k+1}$ and an element of $p \partial (t_i \to^* t_k)$ coincide, then it follows from Lemma 21 that $|p \partial (t_i \to^* t_n)| \leqslant 1$. Otherwise, let k be the smallest number such that $k \geqslant i$ and $p \partial (t_i \to^* t_k) = \{p_2\}$, where p_2 is the redex position of $t_k \to t_{k+1}$. By Definitions 3 and 14, $|p_2 \partial (t_k \to t_{k+1})| \leqslant c$. For each $p_3 \in p_2 \partial (t_k \to t_{k+1})$, if $\mathsf{rt}(t_{k+1}|_{p_3}) \in \mathcal{D}$, then (t_{k+1}, p_3) is a successor node of (t_i, p), and the condition $q \notin p_3 \partial (t_{k+1} \to^* t_n)$ is violated for any main progeny q of p_3. On the other hand, if $\mathsf{rt}(t_{k+1}|_{p_3}) \in \mathcal{C} \cup \mathcal{V}$, then by Lemma 17, $|p_3 \partial (t_{k+1} \to^* t_n)| \leqslant 1$. Thus, in total, there are at most c many elements in $p \partial (t_i \to^* t_n)$ meeting our assumption. □

The following example illustrates the role of Lemma 23.

Example 24. Let $\mathcal{R}_3$ be the TRS consisting of the single rewrite rule

$$\mathsf{d}(\mathsf{S}(x)) \to \mathsf{S}(\mathsf{S}(\mathsf{d}(x))) \ .$$

Let $t_1 = \mathsf{d}(\mathsf{S}(\mathsf{S}(\mathsf{0})))$, $t_2 = \mathsf{S}(\mathsf{S}(\mathsf{d}(\mathsf{S}(\mathsf{0}))))$, and $t_3 = \mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{d}(\mathsf{0})))))$. We have the derivation $A : t_1 \to t_2 \to t_3$ and the following progenitor graph:

$$(t_1, \epsilon) \longrightarrow (t_2, 11) \longrightarrow (t_3, 1111)$$

Note that G leaves out all function symbols S above the d in each term. However, by Lemma 23, the number of positions in the last term of A which are hidden in this way is only linear in the size of the progenitor graph.

Lemma 25. *Every node in G has at most c many successor nodes.*

Proof. Let (t_i, p) be a node in G. If there is no $i \leqslant j < n$ such that the redex position of the step $t_j \to t_{j+1}$ and an element of $p \circlearrowright (t_i \to^* t_j)$ coincide, then (t_i, p) has no successor node, so the claim holds. Otherwise, let j be the smallest number greater than i such that $p \circlearrowright (t_i \to^* t_j) = \{q\}$, where q is the redex position of $t_j \to t_{j+1}$. By Definitions 3 and 14, $|q \circlearrowright (t_j \to t_{j+1})| \leqslant c$. Hence, (t_i, p) has at most c many successor nodes. □

Now we are ready to prove our main lemma.

Lemma 26. *For every finite and terminating TRS $\mathcal{R}$, there exists a constant C such that for all terms s, we have* $\mathsf{pdp}(s, \to_{\mathcal{R}}) \leqslant |s| \cdot 2^{C \cdot \max\{\mathsf{dl}((s')^\sharp, \to_{\mathsf{DP}(\mathcal{R})/\mathcal{R}}) | s' \trianglelefteq s\}}$.

Proof. We show the lemma by proving that for any derivation $A : s \to^*_{\mathcal{R}} t$, there exists a derivation $A' : (s')^\sharp \to^*_{\mathcal{R} \cup \mathsf{DP}(\mathcal{R})} (t')^\sharp$ with $s' \trianglelefteq s$ and $\mathsf{dp}(t) \leqslant |s| \cdot c^{\mathsf{DPl}(A')+2}$ (recall $c = \max\{2\} \cup \{\mathsf{dp}(r) \mid l \to r \in \mathcal{R}\}$). Let k be the number of defined symbols in the main branch of s. The main branch of t consists of $\mathsf{dp}(t) + 1$ many positions, all of which have to fulfil one of the two properties outlined in Lemma 22. By Lemma 17, the first case applies to at most $\mathsf{dp}(s) + 1 - k$ many positions, so for the $\mathsf{dp}(t) - \mathsf{dp}(s) + k$ other positions, the second case applies. By Lemma 23, each node in the progenitor graph G of A can cover at most c many of those positions, so G has to contain at least $\frac{\mathsf{dp}(t) - \mathsf{dp}(s) + k}{c}$ many nodes. There are k many connected components (trees) in G, hence the largest one of them contains at least

$$\frac{\mathsf{dp}(t) - \mathsf{dp}(s) + k}{kc} ,$$

many nodes. Let d be the smallest natural number such that

$$\frac{\mathsf{dp}(t) - \mathsf{dp}(s) + k}{kc} \leqslant c^{d-1} .$$

By Lemma 25, this means that there exists a leaf in the largest tree of G whose distance from the root is at least $d - 2$: recall that any c-ary tree of height $d - 3$ has at most $\frac{c^{d-2}-1}{c-1} \leqslant c^{d-2}$ many nodes (here the height of a tree is the number of edges on the longest path from the root to a leaf). Moreover, by Lemma 20, this path in the graph induces a derivation $A' : (s')^\sharp \to^*_{\mathcal{R} \cup \mathsf{DP}(\mathcal{R})} (t')^\sharp$ with $\mathsf{DPl}(A') \geqslant d - 2$ and $s \trianglerighteq s'$. Reformulating the inequality above yields

$$\mathsf{dp}(t) \leqslant k \cdot c^d + \mathsf{dp}(s) - k \leqslant (\mathsf{dp}(s) + 1) \cdot c^d \leqslant |s| \cdot c^d ,$$

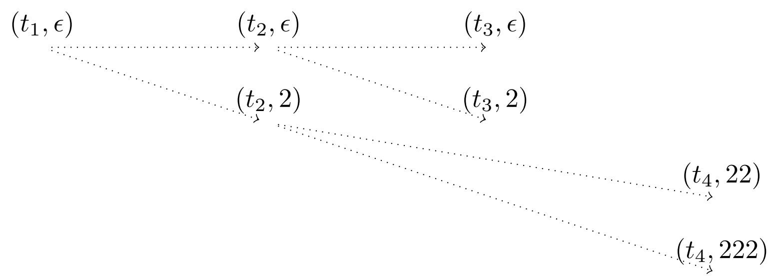
so A' is indeed the derivation we are looking for. □

The main factor of the faster growth of $\mathsf{dp}(t_n)$ compared to the height of G is the difference between the height and the size of G. This becomes apparent in our next example, where G is a full binary tree.

Example 27. Consider the TRS $\mathcal{R}_4$ consisting of the single rewrite rule

$$\mathsf{f}(\mathsf{S}(x), y) \to \mathsf{f}(x, \mathsf{f}(x, y)) \ .$$

Let $t_1 = \mathsf{f}(\mathsf{S}(\mathsf{S}(0)), 0)$, $t_2 = \mathsf{f}(\mathsf{S}(0), \mathsf{f}(\mathsf{S}(0), 0))$, $t_3 = \mathsf{f}(0, \mathsf{f}(0, \mathsf{f}(\mathsf{S}(0), 0)))$, and $t_4 = \mathsf{f}(0, \mathsf{f}(0, \mathsf{f}(0, \mathsf{f}(0, 0))))$. We have the derivation $A : t_1 \to t_2 \to t_3 \to t_4$. The progenitor graph of A is shown below.



Perhaps counter-intuitively in the context of the dependency pair method, the connected component of G with the greatest height need not be the component with the root (t_1, ϵ). All that is left to show is that the derivational complexity of a finite and terminating TRS is bounded double exponentially in its depth growth. This can be achieved by two easy observations.

Lemma 28. *Let $\mathcal{R}$ be a finite and terminating TRS. Then there exists a constant C such that for every term t, we have*

$$\mathsf{dl}(t, \to_{\mathcal{R}}) \leqslant 2^{2^{C \cdot \mathsf{pdp}(t, \to_{\mathcal{R}})}}$$

Proof. We show that there exist constants D and E, such that for all terms t, the inequalities $\mathsf{psz}(t, \to_{\mathcal{R}}) \leqslant 2^{D \cdot \mathsf{pdp}(t, \to_{\mathcal{R}})}$ and $\mathsf{dl}(t, \to_{\mathcal{R}}) \leqslant 2^{E \cdot \mathsf{psz}(t, \to_{\mathcal{R}})}$ hold.

1. For any term t, we have $|t| \leqslant k^{\mathsf{dp}(t)+1}$, where k is the maximum arity of any function symbol in the signature. This proves the first inequality.
2. On the other hand, by assumption the signature $\mathcal{F}$ of $\mathcal{R}$ is finite. Moreover without loss of generality the considered derivation in $\mathcal{R}$ is ground. Hence we can build only $2^{E \cdot m}$ different terms of size at most m, where E depends only on $\mathcal{F}$. This proves the second inequality. □

Based on Lemmata 26 and 28 we obtain our main theorem.

Theorem 29. *For any finite and terminating TRS $\mathcal{R}$, $\mathsf{dc}_{\mathcal{R}}(n) \leqslant 2^{2^{n \cdot 2^{O(\mathsf{DPc}_{\mathcal{R}}(n))}}}$.*

An order $\succ$ on terms is G-*collapsible* for a TRS $\mathcal{R}$ if $s \to^*_{\mathcal{R} \cup \mathsf{DP}(\mathcal{R})} t$ and $s \succ t$ implies $\mathsf{G}(s, \succ) > \mathsf{G}(t, \succ)$ for a mapping G into $\mathbb{N}$. Let $(\gtrsim, \succ)$ be a reduction pair for $\mathcal{R}$. Then $(\gtrsim, \succ)$ is called *collapsible* if there is a mapping G such that $\succ$ is G-collapsible for $\mathcal{R}$.

Theorem 30. *Let $\mathcal{R}$ be a finite TRS, let $(\gtrsim, \succ)$ be a collapsible reduction pair with $\pi(\mathcal{R}) \subseteq \gtrsim$ and $\pi(\mathsf{DP}(\mathcal{R})) \subseteq \succ$ for some argument filtering π. Assume there exists a class of number-theoretic functions $\mathcal{C}$ closed under elementary functions and for some $f \in \mathcal{C}$, and any term t, $\mathsf{G}(\pi(t^\sharp), \succ) \leqslant f(|t|)$. Then $\mathsf{dc}_{\mathcal{R}} \in \mathcal{C}$.*

Proof. By assumption there exists a mapping G that binds the number of dependency pair steps in any $\pi(\mathcal{R}) \cup \pi(\mathsf{DP}(\mathcal{R}))$-derivation. Thus

$$\mathsf{dl}(\pi(t^\sharp), \to_{\pi(\mathsf{DP}(\mathcal{R}))/\pi(\mathcal{R})}) \leqslant \mathsf{G}(\pi(t^\sharp), \succ) \leqslant f(|t|) \ . \tag{1}$$

Moreover it is easy to see that for any derivation in $\mathcal{R} \cup \mathsf{DP}(\mathcal{R})$, there is a derivation in $\pi(\mathcal{R}) \cup \pi(\mathsf{DP}(\mathcal{R}))$ which contains the same number of dependency pair steps. Hence, we obtain

$$\mathsf{dl}(t^\sharp, \to_{\mathsf{DP}(\mathcal{R})/\mathcal{R}}) \leqslant \mathsf{dl}(\pi(t^\sharp), \to_{\pi(\mathsf{DP}(\mathcal{R}))/\pi(\mathcal{R})}) \ .$$

Combining this with (1) and Theorem 29 we obtain $\mathsf{dc}_{\mathcal{R}}(n) \leqslant 2^{2^{n \cdot 2^{a \cdot f(n)}}}$. By assumption the complexity class $\mathcal{C}$ is closed under elementary functions. In particular there exists $g \in \mathcal{C}$ such that $\mathsf{dc}_{\mathcal{R}}(n) \leqslant g(n)$. Thus the theorem follows. □

5 The Lower Bound

By Theorem 29, the derivational complexity of a TRS $\mathcal{R}$ is bounded triple exponentially in its dependency pair complexity. This yields an upper bound. The following TRS establishes a double exponential lower bound.

Example 31. Consider the following TRS $\mathcal{R}_5$, extending the TRS $\mathcal{R}_4$:

$$1\colon\ \mathsf{f}(\mathsf{S}(x), y) \to \mathsf{f}(x, \mathsf{f}(x, y)) \qquad 2\colon\ \mathsf{f}(\mathsf{0}, x) \to \mathsf{c}(x, x)$$

We show that $\mathcal{R}_5$ has linear dependency pair complexity, but admits derivations of double exponential length. Let $F^0_m(x) = x$, $F^{n+1}_m(x) = \mathsf{f}(\mathsf{S}^m(\mathsf{0}), F^n_m(x))$, $C^0(x) = x$, and $C^{n+1}(x) = \mathsf{c}(C^n(x), C^n(x))$. Now, consider the starting term $F^1_n(\mathsf{0})$. As can be easily seen, this term rewrites to $F^{2^n}_0(\mathsf{0})$ in $2^n - 1$ steps using rule 1. Now, we can use rule 2 and an outermost strategy to reach $C^{2^n}(\mathsf{0})$ in $2^{2^n} - 1$ steps, so $\mathsf{dc}_{\mathcal{R}_5}$ is at least double exponential. On the other hand consider $\mathsf{DP}(\mathcal{R}_5)$:

$$3\colon\ \mathsf{f}^\sharp(\mathsf{S}(x), y) \to \mathsf{f}^\sharp(x, \mathsf{f}(x, y)) \qquad 4\colon\ \mathsf{f}^\sharp(\mathsf{S}(x), y) \to \mathsf{f}^\sharp(x, y)$$

We define a (very restricted) polynomial interpretation $\mathcal{A}$ as follows: $\mathsf{f}^\sharp_{\mathcal{A}}(x, y) = x$, $\mathsf{S}_{\mathcal{A}}(x) = x{+}1$, $\mathsf{f}_{\mathcal{A}}(x, y) = \mathsf{c}_{\mathcal{A}}(x, y) = \mathsf{0}_{\mathcal{A}} = 0$, where $\mathcal{R}_5 \subseteq \gtrsim_{\mathcal{A}}$ and $\mathsf{DP}(\mathcal{R}_5) \subseteq >_{\mathcal{A}}$, and $(\gtrsim_{\mathcal{A}}, >_{\mathcal{A}})$ forms a reduction pair. Thus $\mathsf{DPc}_{\mathcal{R}_5}$ is at most linear.

Note that from the proof of Theorem 29 one can distill the following three facts, where each of them is responsible for one of the exponentials in the upper bound:

- The number of leaves in a progenitor graph may be exponential in its height.
- The size of a term may be exponential in its depth.
- The number of terms of size n is exponential in n.

Observe that for an optimal example, we would have to utilise all three criteria, while the just given TRS $\mathcal{R}_5$ utilises only the first two criteria. At this point, it seems impossible to enumerate enough terms of exponential depth and double exponential size so that this is possible. Hence, we conjecture that the upper bound given in Theorem 29 can be improved to double exponential.

6 Conclusion

In this paper we have shown that the derivational complexity of a TRS $\mathcal{R}$ is bounded triple exponentially in its dependency pair complexity. Moreover we have presented an example showing that the relationship is at least double exponential. Furthermore, we conjecture that the upper bound can be improved to a double exponential bound.

The basic dependency pair method [5] forms the basis of our investigations. In particular we allow *argument filtering* for the dependency pairs. A similar result can be shown for *dependency graphs*, but we need to replace the triple exponential correspondence by an even faster growing (but still elementary) correspondence, as shown in the extended version of this paper [20]. Future work will concentrate on establishing better bounds for the studied variants, and analysing the derivational complexity induced by further refinements of the dependency pair method.

To summarise the contribution of this paper, we apply Theorem 30 to three well-studied simplification orders: LPO, MPO and KBO. Recall that the derivational complexity induced by LPO or MPO is multiple recursive or primitive recursive, respectively, cf. [11,12]. Clearly these function classes are closed under elementary functions. Hence by Theorem 30 we obtain that the derivational complexity induced by the basic dependency pair method based on LPO (MPO) is multiple recursive (primitive recursive). On the other hand for a TRS $\mathcal{R}$ compatible with KBO we have that $\mathsf{dc}_{\mathcal{R}}$ belongs to $\mathsf{Ack}(\mathsf{O}(n), 0)$, cf. [13]. Thus applying the theorem in the context of KBO yields that the derivational complexity function induced by the dependency pair method based on KBO is majorised by the Ackermann function. Recall that in all three cases the bounds are tight (see [11,12,13]) and using the same examples, we obtain tightness of the here established bounds.

To conclude, we consider a version of the Ackermann function, introduced by Hofbauer [21] in a slightly simpler way, which we denote as $\mathcal{R}_6$.

$$\begin{aligned} \mathsf{i}(x) \circ (y \circ z) &\to \mathsf{f}(x, \mathsf{i}(x)) \circ (\mathsf{i}(\mathsf{i}(y)) \circ z) & \mathsf{i}(x) &\to x \\ \mathsf{i}(x) \circ (y \circ (z \circ w)) &\to \mathsf{f}(x, \mathsf{i}(x)) \circ (z \circ (y \circ w)) & \mathsf{f}(x, y) &\to x \end{aligned}$$

Note that $\mathcal{R}_6$ is not simply terminating and the derivational complexity of $\mathcal{R}_6$ dominates the Ackermann function. (The latter follows by the same argument as in [21].) However, termination can be shown easily by the basic dependency pair method in conjunction with argument filtering and KBO.

There are nine dependency pairs. For the argument filtering π, we set $\pi(\mathsf{f}) = \pi(\mathsf{f}^\sharp) = \pi(\mathsf{i}^\sharp) = 1$, $\pi(\mathsf{i}) = [1]$, and $\pi(\circ) = \pi(\circ^\sharp) = [1, 2]$. To apply Proposition 2 we use the reduction pair $(\geqslant^{\pi}_{\mathsf{KBO}}, >^{\pi}_{\mathsf{KBO}})$ induced by the admissible weight function w with $w_0 = 1$, $w(\circ) = w(\circ^\sharp) = 1$, and $w(\mathsf{i}) = 0$, together with the precedence $\mathsf{i} \succ \circ, \circ^\sharp$. Hence, by Theorem 30 the derivational complexity of $\mathcal{R}_6$ belongs to $\mathsf{Ack}(\mathsf{O}(n), 0)$ and this bound is optimal, compare [13].

Acknowledgements. The second author would like to thank Dieter Hofbauer for his hospitality, and fruitful discussions about a draft of this paper, during the his stay in Kassel.

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. Koprowski, A., Waldmann, J.: Arctic termination ... below zero. In: Proc. 19th RTA. Volume 5117 of LNCS. (2008) 202–216
3. Moser, G., Schnabl, A., Waldmann, J.: Complexity analysis of term rewriting based on matrix and context dependent interpretations. In: Proc. 28th FSTTCS. Volume 08004 of Dagstuhl Seminar Proceedings. (2008) 304–315
4. Zantema, H.: Termination of term rewriting by semantic labelling. FI **24**(1,2) (1995) 89–105
5. Arts, T., Giesl, J.: Termination of term rewriting using dependency pairs. TCS **236**(1,2) (2000) 133–178
6. Moser, G.: Derivational complexity of Knuth Bendix orders revisited. In: Proc. 13th LPAR. Volume 4246 of LNCS. (2006) 75–89
7. Hirokawa, N., Moser, G.: Automated complexity analysis based on the dependency pair method. In: Proc. 4th IJCAR. Volume 5195 of LNCS. (2008) 364–379
8. Hirokawa, N., Moser, G.: Complexity, graphs, and the dependency pair method. In: Proc. 15th LPAR. Volume 5330 of LNCS. (2008) 652–666
9. Hirokawa, N., Middeldorp, A.: Automating the dependency pair method. IC **199**(1,2) (2005) 172–199
10. Dershowitz, N.: Termination dependencies. In: Proc. 6th WST. Technical Report DSIC-II/15/03, Universidad Politecnica de Valencia (2003) 27–30
11. Weiermann, A.: Termination proofs for term rewriting systems with lexicographic path orderings imply multiply recursive derivation lengths. TCS **139**(1,2) (1995) 355–362
12. Hofbauer, D.: Termination proofs by multiset path orderings imply primitive recursive derivation lengths. TCS **105**(1) (1992) 129–140
13. Lepper, I.: Derivation lengths and order types of Knuth-Bendix orders. TCS **269**(1,2) (2001) 433–450
14. Thiemann, R.: The DP Framework for Proving Termination of Term Rewriting. PhD thesis, University of Aachen (2007)
15. Hirokawa, N., Middeldorp, A.: Tyrolean termination tool: Techniques and features. IC **205** (2007) 474–511
16. Endrullis, J., Waldmann, J., Zantema, H.: Matrix interpretations for proving termination of term rewriting. JAR **40**(3) (2008) 195–220
17. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press (1998)
18. TeReSe: Term Rewriting Systems. Volume 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (2003)
19. Geser, A.: Relative Termination. PhD thesis, Universität Passau (1990)
20. Moser, G., Schnabl, A.: The derivational complexity induced by the dependency pair method. CoRR **abs/0904.0570** (2009)
21. Hofbauer, D.: Termination Proofs and Derivation Lengths in Term Rewriting Systems. PhD thesis, Technische Universität Berlin (1992)