Root-Labeling*

Christian Sternagel and Aart Middeldorp

Institute of Computer Science
University of Innsbruck
Austria

Abstract. In 2006 Jambox, a termination prover developed by Endrullis, surprised the termination community by winning the string rewriting division and almost beating AProVE in the term rewriting division of the international termination competition. The success of Jambox for strings is partly due to a very special case of semantic labeling. In this paper we integrate this technique, which we call root-labeling, into the dependency pair framework. The result is a simple processor with help of which TTT2 surprised the termination community in 2007 by producing the first automatically generated termination proof of a string rewrite system with non-primitive recursive complexity (Touzet, 1998). Unlike many other recent termination methods, the root-labeling processor is trivial to automate and completely unsuitable for producing human readable proofs.

1 Introduction

Semantic labeling is a complete method for proving the termination of term rewrite systems (TRSs), introduced by Zantema [24]. It transforms a given TRS into a termination equivalent TRS by labeling function symbols based on their semantics. The challenge when applying and automating semantic labeling is to choose the labeling functions in such a way that the resulting TRS is easier to prove terminating. Koprowski and Zantema [13, 15] showed how this can be done when algebras over the natural numbers are used together with the lexicographic path order to deal with the resulting infinite TRSs over infinite signatures. In [14] Koprowski and Middeldorp combined predictive labeling—a version of semantic labeling with less constraints [11]—with dependency pairs and modeled the search space as a SAT problem.

A very special version of semantic labeling for string rewrite systems (SRSs), due to Johannes Waldmann (Jörg Endrullis, personal communication), in which the semantic and labeling components are completely determined by the SRS at hand was used by Matchbox/SatELite [23] and Jambox [4] in the string rewriting division of the 2006 international termination competition[1] with remarkable success. This special version, which we call root-labeling, is extended to TRSs in this paper. More importantly but equally straightforward, we present root-labeling

* This research is supported by FWF (Austrian Science Fund) project P18763.

[1] www.lri.fr/~marche/termination-competition/2006/

as a processor in the dependency pair framework [8, 19]. Due to this new root-labeling processor, in 2007 $\mathsf{T\!T\!T\!2}$ [16] could prove the termination of exactly one SRS that had eluded all termination tools (many of which use highly specialized techniques for SRSs) before, resulting in the first automatic termination proof of an SRS whose derivational complexity is not primitive recursive (Touzet [21]).

The remainder of this paper is organized as follows. In the next section we recall basic definitions and results concerning semantic labeling and dependency pairs. In Section 3, semantic labeling is specialized to root-labeling. Incorporating dependency pairs is the topic of Section 4 and in Section 5 we present our main example in some detail. Experimental results are presented in Section 6 and we conclude in Section 7 with suggestions for future research.

2 Preliminaries

We assume basic knowledge of term rewriting [2, 18]. Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and let $\mathcal{A} = (A, \{f_\mathcal{A}\}_{f\in\mathcal{F}})$ be an $\mathcal{F}$-algebra. Let $\mathcal{V}$ be the set of variables. We say that $\mathcal{A}$ is a model of $\mathcal{R}$ if $[\alpha]_\mathcal{A}(l) = [\alpha]_\mathcal{A}(r)$ for every rule $l \to r \in \mathcal{R}$ and every assignment $\alpha\colon \mathcal{V} \to A$. A labeling ℓ for $\mathcal{A}$ consists of sets of labels L_f for every $f \in \mathcal{F}$ together with mappings $\ell_f\colon A^n \to L_f$ for every n-ary function symbol $f \in \mathcal{F}$ with $L_f \neq \varnothing$. The labeled signature $\mathcal{F}_\mathsf{lab}$ consists of n-ary function symbols f_a for every n-ary function symbol $f \in \mathcal{F}$ and label $a \in L_f$ together with all function symbols $f \in \mathcal{F}$ such that $L_f = \varnothing$. The mapping ℓ_f determines the label of the root symbol f of a term $f(t_1, \dots, t_n)$ based on the values of the arguments $t_1, \dots, t_n$. For every assignment $\alpha\colon \mathcal{V} \to A$ the mapping $\mathsf{lab}_\alpha\colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}_\mathsf{lab}, \mathcal{V})$ is inductively defined as follows: $\mathsf{lab}_\alpha(t) = t$ if t is a variable, $\mathsf{lab}_\alpha(f(t_1, \dots, t_n)) = f(\mathsf{lab}_\alpha(t_1), \dots, \mathsf{lab}_\alpha(t_n))$ if $L_f = \varnothing$, and $\mathsf{lab}_\alpha(f(t_1, \dots, t_n)) = f_a(\mathsf{lab}_\alpha(t_1), \dots, \mathsf{lab}_\alpha(t_n))$ if $L_f \neq \varnothing$ where a denotes the label $\ell_f([\alpha]_\mathcal{A}(t_1), \dots, [\alpha]_\mathcal{A}(t_n))$. The *labeled* TRS $\mathcal{R}_\mathsf{lab}$ over the signature $\mathcal{F}_\mathsf{lab}$ consists of the rewrite rules $\mathsf{lab}_\alpha(l) \to \mathsf{lab}_\alpha(r)$ for all rules $l \to r \in \mathcal{R}$ and assignments $\alpha\colon \mathcal{V} \to A$.

Theorem 1 (Zantema [24])**.** *Let $\mathcal{R}$ be a TRS. Let the algebra $\mathcal{A}$ be a non-empty model of $\mathcal{R}$ and let ℓ be a labeling for $\mathcal{A}$. The TRS $\mathcal{R}$ is terminating if and only if the TRS $\mathcal{R}_\mathsf{lab}$ is terminating.* □

Example 2 ([24]). Consider the TRS $\mathcal{R}$ (Toyama [22]) consisting of the single rule $\mathsf{f}(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}(x, x, x)$. To ease the termination proof, we label function symbol f such that its occurrence on the left gets a different label from the one on the right. This is achieved by taking the algebra $\mathcal{A}$ with carrier $\{0, 1\}$ and interpretations $\mathsf{a}_\mathcal{A} = 0$, $\mathsf{b}_\mathcal{A} = 1$, $\mathsf{f}_\mathcal{A}(x, y, z) = 0$ for all $x, y, z \in \{0, 1\}$, together with $L_\mathsf{a} = L_\mathsf{b} = \varnothing$, $L_\mathsf{f} = \{0, 1\}$ and $\ell_\mathsf{f}(x, y, z) = 0$ if $x = y$ and $\ell_\mathsf{f}(x, y, z) = 1$ if $x \neq y$. The algebra $\mathcal{A}$ is a model of $\mathcal{R}$ and $\mathcal{R}_\mathsf{lab}$ consists of the rule $\mathsf{f}_1(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}_0(x, x, x)$. Termination of $\mathcal{R}_\mathsf{lab}$ is obvious as there are no dependency pairs.

A stronger version of semantic labeling is obtained by equipping the carrier of the algebra and the label sets with a well-founded order such that all

algebra functions and all labeling functions are weakly monotone in all coordinates. The model condition is then weakened to $[\alpha]_{\mathcal{A}}(l) \geqslant [\alpha]_{\mathcal{A}}(r)$ for every rule $l \to r \in \mathcal{R}$ and every assignment $\alpha\colon \mathcal{V} \to A$. Further, all rules of the form $f_a(x_1, \ldots, x_n) \to f_b(x_1, \ldots, x_n)$ with $a, b \in L_f$ such that $a > b$ have to be added to $\mathcal{R}_{\mathsf{lab}}$ in order to obtain a sound transformation (Zantema [24]). This version of semantic labeling is capable of transforming any terminating TRS into a TRS whose termination proof is particularly simple, see [17]. This result is however only of theoretical interest. Recent variants inspired by the need for automation are presented in [11, 20].

The dependency pair method [1] is a powerful approach for proving termination of TRSs. It is used in most termination tools for term rewriting. The dependency pair framework [8, 19] is a modular reformulation and improvement of this approach. We present a simplified version which is sufficient for our purposes. Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}_{\mathcal{R}}$. The signature $\mathcal{F}_{\mathcal{R}}$ is extended with symbols $f^\sharp$ for every symbol $f \in \{\mathsf{root}(l) \mid l \to r \in \mathcal{R}\}$, where $f^\sharp$ has the same arity as f. In examples we write F for $f^\sharp$. If $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ with $\mathsf{root}(t)$ defined then $t^\sharp$ denotes the term that is obtained from t by replacing its root symbol with $\mathsf{root}(t)^\sharp$. If $l \to r \in \mathcal{R}$ and t is a subterm of r with a defined root symbol that is not a proper subterm of l then the rule $l^\sharp \to t^\sharp$ is a *dependency pair* of $\mathcal{R}$. The set of dependency pairs of $\mathcal{R}$ is denoted by $\mathsf{DP}(\mathcal{R})$. A *DP problem* is a pair of TRSs $(\mathcal{P}, \mathcal{R})$ such that symbols in $\mathcal{F}^\sharp = \{\mathsf{root}(l), \mathsf{root}(r) \mid l \to r \in \mathcal{P}\}$ do neither occur in $\mathcal{R}$ nor in proper subterms of the left and right-hand sides of rules in $\mathcal{P}$. Writing $\mathcal{F}_{\mathcal{P}}$ for the signature of $\mathcal{P}$, the signature $\mathcal{F}_{\mathcal{R}} \cup (\mathcal{F}_{\mathcal{P}} \setminus \mathcal{F}^\sharp)$ is denoted by $\mathcal{F}$. The problem $(\mathcal{P}, \mathcal{R})$ is said to be *finite* if there is no infinite sequence $s_1 \xrightarrow{\epsilon}_{\mathcal{P}} t_1 \to^*_{\mathcal{R}} s_2 \xrightarrow{\epsilon}_{\mathcal{P}} t_2 \to^*_{\mathcal{R}} \cdots$ such that all terms t_1, t_2, ... are terminating with respect to $\mathcal{R}$. Such an infinite sequence is said to be *minimal*. Here the ϵ in $\xrightarrow{\epsilon}_{\mathcal{P}}$ denotes that the application of the rule in $\mathcal{P}$ takes place at the root position. The main result underlying the dependency pair approach states that a TRS $\mathcal{R}$ is terminating if and only if the DP problem $(\mathsf{DP}(\mathcal{R}), \mathcal{R})$ is finite.

In order to prove finiteness of a DP problem a number of so-called *DP processors* have been developed. DP processors are functions that take a DP problem as input and return a set of DP problems as output. In order to be employed to prove termination they need to be *sound*, that is, if all DP problems in a set returned by a DP processor are finite then the initial DP problem is finite. *Complete* DP processors, which are those processors with the property that if one of the returned DP problems is not finite then the original DP problem is not finite, can be used to prove non-termination.

3 Plain Root-Labeling[2]

The challenge when implementing semantic labeling is to find an appropriate carrier and suitable interpretation and labeling functions such that the labeled

[2] As stated in the introduction, the results in this section for SRSs are due to Johannes Waldmann; they are mentioned in [26].

system is easier to prove terminating. This issue has been addressed in several recent papers ([13, 15, 14]).

In the special version of semantic labeling defined in this section, everything is fixed. This has the disadvantage of reducing the power of semantic labeling significantly but the advantage of making automation a trivial issue.

Definition 3. *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$. The algebra $\mathcal{A}_\mathcal{F}$ has carrier $\mathcal{F}$ and interpretation functions $f_{\mathcal{A}_\mathcal{F}}(x_1, \ldots, x_n) = f$ for every n-ary $f \in \mathcal{F}$ and all $x_1, \ldots, x_n \in \mathcal{F}$. The labeling ℓ is defined as follows: $L_f = \mathcal{F}^n$ if the arity n of f is at least 1 and $L_f = \varnothing$ otherwise, and $\ell_f(x_1, \ldots, x_n) = (x_1, \ldots, x_n)$ for all f with $L_f \neq \varnothing$ and all $x_1, \ldots, x_n \in \mathcal{F}$. In examples we write x_1 for (x_1). The resulting labeled TRS $\mathcal{R}_{\mathsf{lab}}$ is denoted by $\mathcal{R}_{\mathsf{rl}}$.*

Example 4. Consider the TRS $\mathcal{R}$ from Example 2, extended with the two rules $\mathsf{c} \to \mathsf{a}$ and $\mathsf{c} \to \mathsf{b}$. The TRS $\mathcal{R}_{\mathsf{rl}}$ consists of the six rules

$$\begin{array}{lll} \mathsf{f}_{(\mathsf{a},\mathsf{b},\mathsf{a})}(\mathsf{a},\mathsf{b},x) \to \mathsf{f}_{(\mathsf{a},\mathsf{a},\mathsf{a})}(x,x,x) & \mathsf{f}_{(\mathsf{a},\mathsf{b},\mathsf{c})}(\mathsf{a},\mathsf{b},x) \to \mathsf{f}_{(\mathsf{c},\mathsf{c},\mathsf{c})}(x,x,x) & \mathsf{c} \to \mathsf{a} \\ \mathsf{f}_{(\mathsf{a},\mathsf{b},\mathsf{b})}(\mathsf{a},\mathsf{b},x) \to \mathsf{f}_{(\mathsf{b},\mathsf{b},\mathsf{b})}(x,x,x) & \mathsf{f}_{(\mathsf{a},\mathsf{b},\mathsf{f})}(\mathsf{a},\mathsf{b},x) \to \mathsf{f}_{(\mathsf{f},\mathsf{f},\mathsf{f})}(x,x,x) & \mathsf{c} \to \mathsf{b} \end{array}$$

and is terminating because there are no dependency pairs. The TRS $\mathcal{R}$, however, admits an infinite rewrite sequence starting from the term $\mathsf{f}(\mathsf{a},\mathsf{b},\mathsf{c})$.

The problem in the previous example is that $\mathcal{A}_\mathcal{F}$ is not a model of $\mathcal{R}$. In order to solve this problem we close every rule $l \to r$ where l and r have different root symbols under flat contexts, before performing the root-labeling operation.

Definition 5. *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$. The rules in $\mathcal{R}_\mathsf{p} = \{l \to r \in \mathcal{R} \mid \mathsf{root}(l) = \mathsf{root}(r)\}$ are* root-preserving. *The rules in $\mathcal{R}_\mathsf{a} = \mathcal{R} \setminus \mathcal{R}_\mathsf{p}$ are* root-altering. *The set $\{f(x_1, \ldots, x_{i-1}, \Box, x_{i+1}, \ldots, x_n) \mid f \in \mathcal{F}$ has arity $n \geqslant 1$ and $1 \leqslant i \leqslant n\}$ of* flat *contexts is denoted by $\mathcal{FC}$. The* flat context closure *of $\mathcal{R}$ is defined as $\mathcal{FC}(\mathcal{R}) = \mathcal{R}_\mathsf{p} \cup \{C[l] \to C[r] \mid l \to r \in \mathcal{R}_\mathsf{a}$ and $C \in \mathcal{FC}\}$.*

Example 6. For the TRS $\mathcal{R}$ from Example 4, the TRS $\mathcal{FC}(\mathcal{R})$ consisting of the rules

$$\begin{array}{lll} \mathsf{f}(\mathsf{a},\mathsf{b},x) \to \mathsf{f}(x,x,x) & \mathsf{f}(\mathsf{c},x,y) \to \mathsf{f}(\mathsf{a},x,y) & \mathsf{f}(\mathsf{c},x,y) \to \mathsf{f}(\mathsf{b},x,y) \\ & \mathsf{f}(x,\mathsf{c},y) \to \mathsf{f}(x,\mathsf{a},y) & \mathsf{f}(x,\mathsf{c},y) \to \mathsf{f}(x,\mathsf{b},y) \\ & \mathsf{f}(x,y,\mathsf{c}) \to \mathsf{f}(x,y,\mathsf{a}) & \mathsf{f}(x,y,\mathsf{c}) \to \mathsf{f}(x,y,\mathsf{b}) \end{array}$$

is obtained. Note that like $\mathcal{R}$, $\mathcal{FC}(\mathcal{R})$ is non-terminating for $\mathsf{f}(\mathsf{a},\mathsf{b},\mathsf{c})$.

Lemma 7. *The transformation $\mathcal{FC}(\cdot)$ on TRSs is termination preserving and reflecting, i.e., $\mathcal{R}$ is terminating if and only if $\mathcal{FC}(\mathcal{R})$ is terminating.*

Proof. By construction, every rewrite step in $\mathcal{FC}(\mathcal{R})$ can be simulated by a rewrite step in $\mathcal{R}$. This proves the "only if" direction. For the "if" direction we reason as follows. Suppose $\mathcal{R}$ is not terminating. Then there exists an infinite sequence $t_1 \to_\mathcal{R} t_2 \to_\mathcal{R} \cdots$. Let $C \in \mathcal{FC}$ be an arbitrary flat context. Since

rewriting is closed under contexts, we obtain $C[t_1] \to_{\mathcal{R}} C[t_2] \to_{\mathcal{R}} \cdots$. We claim that this sequence is a rewrite sequence in $\mathcal{FC}(\mathcal{R})$. Fix $i \geqslant 1$ and consider the step $C[t_i] \to_{\mathcal{R}} C[t_{i+1}]$. Let $l \to r \in \mathcal{R}$ be the employed rewrite rule. We distinguish two cases.

1. If $l \to r$ is root-preserving then $l \to r$ belongs to $\mathcal{FC}(\mathcal{R})$ and the result is clear.
2. Suppose $l \to r$ is root-altering. If the rule was applied below the root position in $t_i \to_{\mathcal{R}} t_{i+1}$, then $t_i \to_{\mathcal{FC}(\mathcal{R})} t_{i+1}$ by applying the rule $C'[l] \to C'[r]$ for the flat context $C' \in \mathcal{FC}$ uniquely determined by the function symbol in t_i directly above the redex and the position of the redex. Formally, if $\pi{\cdot}j$ is the redex position in t_i then $C' = f(x_1, \ldots, x_{j-1}, \Box, x_{j+1}, \ldots, x_n)$ with $f = \mathsf{root}(t_i|_\pi)$. Hence also $C[t_i] \to_{\mathcal{FC}(\mathcal{R})} C[t_{i+1}]$. If the rule $l \to r$ was applied at the root position in $t_i \to_{\mathcal{R}} t_{i+1}$ then $C[t_i] \to_{\mathcal{FC}(\mathcal{R})} C[t_{i+1}]$ by applying the rule $C[l] \to C[r] \in \mathcal{FC}(\mathcal{R})$.

So $C[t_1] \to_{\mathcal{FC}(\mathcal{R})} C[t_2] \to_{\mathcal{FC}(\mathcal{R})} \cdots$ and thus $\mathcal{FC}(\mathcal{R})$ is non-terminating. □

To pave the way for the developments in the next section, we need the observation that Lemma 7 remains true if we allow an arbitrary extension $\mathcal{G}$ of the signature $\mathcal{F}$ of $\mathcal{R}$ when building flat contexts. We write $\mathcal{FC}_{\mathcal{G}}$ when we want to make the signature of the flat contexts clear.

Theorem 8. *The transformation $\mathcal{FC}(\cdot)_{\mathsf{rl}}$ on TRSs is termination preserving and reflecting, i.e., $\mathcal{R}$ is terminating if and only if $\mathcal{FC}(\mathcal{R})_{\mathsf{rl}}$ is terminating.*

Proof. According to Lemma 7, termination of $\mathcal{R}$ is equivalent to termination of $\mathcal{FC}(\mathcal{R})$. By construction, all rules in $\mathcal{FC}(\mathcal{R})$ are root-preserving. Hence $\mathcal{A}_{\mathcal{F}}$ is a model for $\mathcal{FC}(\mathcal{R})$ and Theorem 1 yields the termination equivalence of $\mathcal{FC}(\mathcal{R})$ and $\mathcal{FC}(\mathcal{R})_{\mathsf{rl}}$. Combining the two equivalences yields the desired result. □

We conclude this section with two more examples. A string rewrite system (SRS) is a TRS over a signature consisting of unary function symbols. We write strings $\mathsf{a}(\mathsf{b}(\mathsf{c}(x)))$ as abc (the variable is implicit).

Example 9. Consider the SRS $\mathcal{R} = \{\mathsf{aa} \to \mathsf{aba}\}$. Since the rule is root-preserving, $\mathcal{FC}(\mathcal{R}) = \mathcal{R}$. The SRS $\mathcal{FC}(\mathcal{R})_{\mathsf{rl}}$ consists of the two rules $\mathsf{a_a a_a} \to \mathsf{a_b b_a a_a}$ and $\mathsf{a_a a_b} \to \mathsf{a_b b_a a_b}$, and is terminating because its rules are oriented from left to right by the polynomial interpretation $[\mathsf{a_a}](x) = x + 1$ and $[\mathsf{a_b}](x) = [\mathsf{b_a}](x) = x$.

Example 10. Consider the TRS $\mathcal{R}$ (`teparla3.trs`) consisting of the two rules $\mathsf{f}(y, \mathsf{f}(x, \mathsf{f}(\mathsf{a}, x))) \to \mathsf{f}(\mathsf{f}(\mathsf{a}, \mathsf{f}(x, \mathsf{a})), \mathsf{f}(\mathsf{a}, y))$ and $\mathsf{f}(x, \mathsf{f}(x, y)) \to \mathsf{f}(\mathsf{f}(\mathsf{f}(x, \mathsf{a}), \mathsf{a}), \mathsf{a})$. None of the tools participating in the 2007 international competition could prove its termination. Like in the preceding example we have $\mathcal{R} = \mathcal{R}_{\mathsf{p}}$ and hence $\mathcal{FC}(\mathcal{R}) = \mathcal{R}$. The TRS $\mathcal{FC}(\mathcal{R})_{\mathsf{rl}}$ consists of the following eight rules

$$\mathsf{f}_{(\mathsf{a},\mathsf{f})}(y, \mathsf{f}_{(\mathsf{a},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{a},\mathsf{a})}(\mathsf{a}, x))) \to \mathsf{f}_{(\mathsf{f},\mathsf{f})}(\mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, \mathsf{f}_{(\mathsf{a},\mathsf{a})}(x, \mathsf{a})), \mathsf{f}_{(\mathsf{a},\mathsf{a})}(\mathsf{a}, y))$$
$$\mathsf{f}_{(\mathsf{f},\mathsf{f})}(y, \mathsf{f}_{(\mathsf{a},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{a},\mathsf{a})}(\mathsf{a}, x))) \to \mathsf{f}_{(\mathsf{f},\mathsf{f})}(\mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, \mathsf{f}_{(\mathsf{a},\mathsf{a})}(x, \mathsf{a})), \mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, y))$$

$$\mathsf{f}_{(\mathsf{a},\mathsf{f})}(y, \mathsf{f}_{(\mathsf{f},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, x))) \to \mathsf{f}_{(\mathsf{f},\mathsf{f})}(\mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, \mathsf{f}_{(\mathsf{f},\mathsf{a})}(x, \mathsf{a})), \mathsf{f}_{(\mathsf{a},\mathsf{a})}(\mathsf{a}, y))$$
$$\mathsf{f}_{(\mathsf{f},\mathsf{f})}(y, \mathsf{f}_{(\mathsf{f},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, x))) \to \mathsf{f}_{(\mathsf{f},\mathsf{f})}(\mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, \mathsf{f}_{(\mathsf{f},\mathsf{a})}(x, \mathsf{a})), \mathsf{f}_{(\mathsf{a},\mathsf{f})}(\mathsf{a}, y))$$
$$\mathsf{f}_{(\mathsf{a},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{a},\mathsf{a})}(x, y)) \to \mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{a},\mathsf{a})}(x, \mathsf{a}), \mathsf{a}), \mathsf{a})$$
$$\mathsf{f}_{(\mathsf{a},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{a},\mathsf{f})}(x, y)) \to \mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{a},\mathsf{a})}(x, \mathsf{a}), \mathsf{a}), \mathsf{a})$$
$$\mathsf{f}_{(\mathsf{f},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{f},\mathsf{a})}(x, y)) \to \mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{f},\mathsf{a})}(x, \mathsf{a}), \mathsf{a}), \mathsf{a})$$
$$\mathsf{f}_{(\mathsf{f},\mathsf{f})}(x, \mathsf{f}_{(\mathsf{f},\mathsf{f})}(x, y)) \to \mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{f},\mathsf{a})}(\mathsf{f}_{(\mathsf{f},\mathsf{a})}(x, \mathsf{a}), \mathsf{a}), \mathsf{a})$$

and can be proved terminating by the 2007 competition versions of AProVE [7], Jambox, and TTT2.

4 Root-Labeling with Dependency Pairs

The performance of Jambox and Matchbox/SatELite [23] in the 2006 international termination competition revealed that root-labeling is a powerful transformation on SRSs. So it is not a surprise that other tools adopted this technique, too. In 2007, MultumNonMulta [12] and Torpa [25] followed suit. So did TTT2, with one important difference: the combination with dependency pairs.

In the previous section we defined root-labeling as a transformation on TRSs. In order to benefit from the numerous termination techniques that are available in connection with the dependency pair framework, it is worthwhile to extend root-labeling to dependency pair problems. (For the same reason, in [14] semantic labeling with respect to quasi-models is extended to dependency pair problems.) In this section we present two different approaches that achieve this.

In the first approach, which is the one implemented in the 2007 competition version of TTT2, the insertion of a fresh unary function symbol below dependency pair symbols ensures that the strict separation of rules in $\mathcal{P}$ and in $\mathcal{R}$ is maintained.

Definition 11. *Let $(\mathcal{P}, \mathcal{R})$ be a DP problem. Let $\mathcal{F}_\mathcal{R}$ be the signature of $\mathcal{R}$ and let $\mathcal{F}_\mathcal{P}$ be the signature of $\mathcal{P}$. We denote $\{\mathsf{root}(l), \mathsf{root}(r) \mid l \to r \in \mathcal{P}\}$ by $\mathcal{F}^\sharp$ and $\mathcal{F}_\mathcal{R} \cup (\mathcal{F}_\mathcal{P} \setminus \mathcal{F}^\sharp)$ by $\mathcal{F}$. Let $\triangle$ be a fresh unary function symbol. The function* block *inserts $\triangle$ between the root symbol f and the arguments $t_1, \ldots, t_n$ of a term $f(t_1, \ldots, t_n)$ with $n \geqslant 1$: $\mathsf{block}(t) = t$ if t is a variable or a constant and $\mathsf{block}(t) = f(\triangle(t_1), \ldots, \triangle(t_n))$ if $t = f(t_1, \ldots, t_n)$ with $n \geqslant 1$. We define $\mathcal{FC}_1(\mathcal{P}, \mathcal{R})$ as the pair $(\mathsf{block}(\mathcal{P}), \mathcal{FC}_1(\mathcal{R}))$ where $\mathsf{block}(\mathcal{P}) = \{\mathsf{block}(l) \to \mathsf{block}(r) \mid l \to r \in \mathcal{P}\}$ and $\mathcal{FC}_1(\mathcal{R}) = \mathcal{FC}_{\mathcal{F} \cup \{\triangle\}}(\mathcal{R})$.*

Lemma 12. *The pair $\mathcal{FC}_1(\mathcal{P}, \mathcal{R})$ is a DP problem.*

Proof. The set $\{\mathsf{root}(l), \mathsf{root}(r) \mid l \to r \in \mathsf{block}(\mathcal{P})\}$ coincides with $\mathcal{F}^\sharp$ and function symbols in $\mathcal{F}^\sharp$ do occur neither in $\mathcal{FC}_1(\mathcal{R})$ (as $\mathcal{F}^\sharp \cap (\mathcal{F} \cup \{\triangle\}) = \varnothing$) nor in proper subterms of terms in $\mathsf{block}(\mathcal{P})$. □

Lemma 13. *The DP problem $(\mathcal{P}, \mathcal{R})$ is finite if and only if the DP problem $\mathcal{FC}_1(\mathcal{P}, \mathcal{R})$ is finite.*

Proof. First assume that $(\mathcal{P},\mathcal{R})$ is not finite. Then there exists a minimal sequence $s_1 \xrightarrow{\epsilon}_{\mathcal{P}} t_1 \to^*_{\mathcal{R}} s_2 \xrightarrow{\epsilon}_{\mathcal{P}} t_2 \to^*_{\mathcal{R}} \cdots$. We may assume without loss of generality that all function symbols occurring at non-root positions belong to $\mathcal{F}$. (By replacing all maximal proper subterms with a root symbol that belongs to $\mathcal{F}^\sharp$ by the same variable, we obtain a minimal sequence that satisfies this property.) We claim that

$$\mathsf{block}(s_1) \xrightarrow{\epsilon}_{\mathsf{block}(\mathcal{P})} \mathsf{block}(t_1) \to^*_{\mathcal{FC}_1(\mathcal{R})} \mathsf{block}(s_2) \xrightarrow{\epsilon}_{\mathsf{block}(\mathcal{P})} \mathsf{block}(t_2) \to^*_{\mathcal{FC}_1(\mathcal{R})} \cdots$$

is a minimal sequence with respect to $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$. Fix $i \geqslant 1$. By construction of $\mathsf{block}(\mathcal{P})$, the step $s_i \xrightarrow{\epsilon}_{\mathcal{P}} t_i$ gives rise to the step $\mathsf{block}(s_i) \xrightarrow{\epsilon}_{\mathsf{block}(\mathcal{P})} \mathsf{block}(t_i)$. Next we consider the sequence $t_i \to^*_{\mathcal{R}} s_{i+1}$. Let $s \to_{\mathcal{R}} t$ be an arbitrary step in this sequence, using the rewrite rule $l \to r \in \mathcal{R}$ at position π. Since $\pi > \epsilon$ we may write $s = F(u_1,\ldots,u_n)$ and $t = F(u_1,\ldots,u_{j-1},v_j,u_{j+1},\ldots,u_n)$ with $u_j \to_{\mathcal{R}} v_j$ and $\pi \geqslant j$. If $l \to r$ is root-preserving then $l \to r \in \mathcal{FC}_1(\mathcal{R})$ and thus $u_j \to_{\mathcal{FC}_1(\mathcal{R})} v_j$, which implies $\triangle(u_j) \to_{\mathcal{FC}_1(\mathcal{R})} \triangle(v_j)$. Similar as in the proof of Lemma 7, if $l \to r$ is root-altering then we obtain $\triangle(u_j) \to_{\mathcal{FC}_1(\mathcal{R})} \triangle(v_j)$ by using an appropriate flat context from $\mathcal{FC}_\mathcal{F}$ when $\pi > j$ and the flat context $\triangle(\Box)$ when $\pi = j$. So in all cases we have $\triangle(u_j) \to_{\mathcal{FC}_1(\mathcal{R})} \triangle(v_j)$ and hence also $\mathsf{block}(s) \to_{\mathcal{FC}_1(\mathcal{R})} \mathsf{block}(t)$. Since the step $s \to_{\mathcal{R}} t$ was an arbitrary step in the sequence $t_i \to^*_{\mathcal{R}} s_{i+1}$, we obtain $\mathsf{block}(t_i) \to^*_{\mathcal{FC}_1(\mathcal{R})} \mathsf{block}(s_{i+1})$. Next we show that $\mathsf{block}(t_i)$ is terminating with respect to $\mathcal{FC}_1(\mathcal{R})$. By construction of $\mathcal{FC}_1(\mathcal{R})$, every application of a rule in $\mathcal{FC}_1(\mathcal{R})$ can be performed by a rule in $\mathcal{R}$. Hence if $\mathsf{block}(t_i)$ is not terminating with respect to $\mathcal{FC}_1(\mathcal{R})$ then $\mathsf{block}(t_i)$ is not terminating with respect to $\mathcal{R}$ and this implies in turn that t_i is not terminating with respect to $\mathcal{R}$ as $\triangle$ does not appear in the rules of $\mathcal{R}$. This, however, contradicts the minimality of the initial sequence $s_1 \xrightarrow{\epsilon}_{\mathcal{P}} t_1 \to^*_{\mathcal{R}} s_2 \xrightarrow{\epsilon}_{\mathcal{P}} t_2 \to^*_{\mathcal{R}} \cdots$. It follows that the sequence displayed above is a minimal sequence with respect to $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$. Therefore, $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$ is not finite, which concludes the proof of the "if" direction.

For the "only if" direction, suppose that

$$s_1 \xrightarrow{\epsilon}_{\mathsf{block}(\mathcal{P})} t_1 \to^*_{\mathcal{FC}_1(\mathcal{R})} s_2 \xrightarrow{\epsilon}_{\mathsf{block}(\mathcal{P})} t_2 \to^*_{\mathcal{FC}_1(\mathcal{R})} \cdots$$

is a minimal sequence with respect to $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$. Let the mapping $\mathsf{unblock}$ erase all occurrences of $\triangle$ from terms:

$$\mathsf{unblock}(t) = \begin{cases} t & \text{if } t \text{ is a variable,} \\ \mathsf{unblock}(t_1) & \text{if } t = \triangle(t_1), \\ f(\mathsf{unblock}(t_1),\ldots,\mathsf{unblock}(t_n)) & \text{if } t = f(t_1,\ldots,t_n) \text{ with } f \neq \triangle. \end{cases}$$

Using similar reasoning as in the "if" direction, we easily obtain

$$\mathsf{unblock}(s_1) \xrightarrow{\epsilon}_{\mathcal{P}} \mathsf{unblock}(t_1) \to^*_{\mathcal{R}} \mathsf{unblock}(s_2) \xrightarrow{\epsilon}_{\mathcal{P}} \mathsf{unblock}(t_2) \to^*_{\mathcal{R}} \cdots$$

To show minimality, suppose that $\mathsf{unblock}(t_i)$ is non-terminating with respect to $\mathcal{R}$. Using the special structure of t_i, it readily follows that t_i is non-terminating with respect to $\mathcal{R}$ and with respect to $\mathcal{FC}_1(\mathcal{R})$. The latter provides the desired contradiction. □

When applying root-labeling to $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$, it is not useful to label the root symbols of $\mathsf{block}(\mathcal{P})$, since identical symbols will always have identical labels consisting solely of $\triangle$'s. This is reflected in the following definition.

Definition 14. *Let $(\mathcal{P},\mathcal{R})$, $\mathcal{F}^\sharp$, $\mathcal{F}$, and $\triangle$ be as in Definition 11. The first root-labeling transformation $\mathcal{FC}_1(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$ is defined as the pair $(\mathsf{block}(\mathcal{P})_{\mathsf{rl}},\mathcal{FC}_1(\mathcal{R})_{\mathsf{rl}})$ with $L_f=\varnothing$ if $f\in\mathcal{F}^\sharp$ or if f is a constant in $\mathcal{F}$ and $L_f=\mathcal{F}^n$ if $f\in\mathcal{F}\cup\{\triangle\}$ has arity $n\geqslant 1$, and $f_{\mathcal{A}_{\mathcal{F}\cup\mathcal{F}^\sharp\cup\{\triangle\}}}(x_1,\ldots,x_n)=g$ for every $f\in\mathcal{F}^\sharp$ and arbitrary but fixed $g\in\mathcal{F}^\sharp$.*

The modification of the algebra $\mathcal{A}_{\mathcal{F}\cup\mathcal{F}^\sharp\cup\{\triangle\}}$ in the last line of the above definition ensures that the model condition is trivially satisfied for the rules in $\mathcal{P}$. Hence these rules need not be closed under flat contexts, even if they are root-altering.

Theorem 15. *The DP problem $(\mathcal{P},\mathcal{R})$ is finite if and only if the DP problem $\mathcal{FC}_1(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$ is finite.*

In other words, the mapping $(\mathcal{P},\mathcal{R})\mapsto\{\mathcal{FC}_1(\mathcal{P},\mathcal{R})_{\mathsf{rl}}\}$ is a sound and complete DP processor.

Proof. According to Lemma 13 finiteness of $(\mathcal{P},\mathcal{R})$ is equivalent to finiteness of $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$. The latter is equivalent to finiteness of $\mathcal{FC}_1(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$. The proof is standard. Starting from a minimal sequence in $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$, a minimal sequence in $\mathcal{FC}_1(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$ is obtained by applying lab_α to every term in the sequence in $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$, where α assigns an arbitrary element of $\mathcal{F}$ to every variable. Conversely, a minimal sequence in $\mathcal{FC}_1(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$ is transformed into a minimal sequence in $\mathcal{FC}_1(\mathcal{P},\mathcal{R})$ by simply erasing all labels. □

In the second approach for incorporating root-labeling into the dependency pair framework, we preserve the model condition by closing the rules of $\mathcal{R}$ also under flat contexts with a root symbol from $\mathcal{F}^\sharp$. To avoid problems by mixing up dependency pair symbols with symbols of $\mathcal{R}$, those additional closure rules are moved to the first component of dependency pair problems.

Definition 16. *Let $(\mathcal{P},\mathcal{R})$, $\mathcal{F}^\sharp$, and $\mathcal{F}$ be as in Definition 11. Let $\mathcal{FC}_2(\mathcal{P},\mathcal{R})$ be the pair $(\mathcal{P}\cup\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_{\mathsf{a}}),\mathcal{FC}_{\mathcal{F}}(\mathcal{R}))$. The second root-labeling transformation $\mathcal{FC}_2(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$ is defined as the pair $(\mathcal{P}_{\mathsf{rl}}\cup\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_{\mathsf{a}})_{\mathsf{rl}},\mathcal{FC}_{\mathcal{F}}(\mathcal{R})_{\mathsf{rl}})$ with $L_f=\varnothing$ if f is a constant in $\mathcal{F}\cup\mathcal{F}^\sharp$ and $L_f=\mathcal{F}^n$ if $f\in\mathcal{F}\cup\mathcal{F}^\sharp$ has arity $n\geqslant 1$, and $f_{\mathcal{A}_{\mathcal{F}\cup\mathcal{F}^\sharp}}(x_1,\ldots,x_n)=g$ for every $f\in\mathcal{F}^\sharp$ and arbitrary but fixed $g\in\mathcal{F}^\sharp$.*

It is obvious that the pair $\mathcal{FC}_2(\mathcal{P},\mathcal{R})$ is a DP problem.

Lemma 17. *The DP problem $(\mathcal{P},\mathcal{R})$ is finite if and only if the DP problem $\mathcal{FC}_2(\mathcal{P},\mathcal{R})$ is finite.*

Proof. We abbreviate $\mathcal{P}\cup\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_{\mathsf{a}})$ to $\mathcal{FC}_2(\mathcal{P})$. Assume that $(\mathcal{P},\mathcal{R})$ is not finite. Hence there exists a minimal sequence $s_1\xrightarrow{\epsilon}_{\mathcal{P}}t_1\to^*_{\mathcal{R}}s_2\xrightarrow{\epsilon}_{\mathcal{P}}t_2\to^*_{\mathcal{R}}\cdots$.

Without loss of generality we assume that all function symbols occurring at non-root positions belong to $\mathcal{F}$. We claim that

$$s_1 \xrightarrow{\epsilon}_{\mathcal{P}} t_1 \to^*_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})\cup\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_\mathsf{a})} s_2 \xrightarrow{\epsilon}_{\mathcal{P}} t_2 \to^*_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})\cup\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_\mathsf{a})} \cdots$$

is a minimal sequence with respect to $\mathcal{FC}_2(\mathcal{P},\mathcal{R})$. Fix $i \geqslant 1$. We obviously have $s_i \xrightarrow{\epsilon}_{\mathcal{P}} t_i$. Let $s \to_{\mathcal{R}} t$ be an arbitrary step in the sequence $t_i \to^*_{\mathcal{R}} s_{i+1}$, using the rewrite rule $l \to r \in \mathcal{R}$ at position π. Since $\pi > \epsilon$ we may write $s = F(u_1,\ldots,u_n)$ and $t = F(u_1,\ldots,u_{j-1},v_j,u_{j+1},\ldots,u_n)$ with $u_j \to_{\mathcal{R}} v_j$ and $\pi \geqslant j$. If $l \to r$ is root-preserving then $l \to r \in \mathcal{FC}_{\mathcal{F}}(\mathcal{R})$ and thus $s \to_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})} t$. Similar as in the proof of Lemma 7, if $l \to r$ is root-altering and $\pi > j$ then we obtain $u_j \to_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})} v_j$ and thus $s \to_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})} t$ by using an appropriate flat context from $\mathcal{FC}_{\mathcal{F}}$. If $\pi = j$ then $s \to_{\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_\mathsf{a})} t$ by using the flat context $F(x_1,\ldots,x_{j-1},\Box,x_{j+1},\ldots,x_n) \in \mathcal{FC}_{\mathcal{F}^\sharp}$. So in all cases we have $s \to_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})\cup\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_\mathsf{a})} t$. Hence the sequence displayed above exists. By pinpointing the steps from $\mathcal{P} \cup \mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_\mathsf{a})$, this sequence can be written as

$$s_1 \xrightarrow{\epsilon}_{\mathcal{FC}_2(\mathcal{P})} t'_1 \to^*_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})} s'_2 \xrightarrow{\epsilon}_{\mathcal{FC}_2(\mathcal{P})} t'_2 \to^*_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})} \cdots$$

where for every $i \geqslant 1$ there exists a $j \geqslant i$ such that $t_i = t'_j$. We need to show that every t'_j is terminating with respect to $\mathcal{FC}_{\mathcal{F}}(\mathcal{R})$. Let $j \geqslant 1$. We distinguish two cases. If $t'_j = t_i$ for some i then t'_j is terminating with respect to $\mathcal{R}$, due to the minimality of the initial sequence in $(\mathcal{P},\mathcal{R})$. According to Lemma 7, more precisely the extension of Lemma 7 mentioned in the paragraph after the proof, t'_j is terminating with respect to $\mathcal{FC}_{\mathcal{F}}(\mathcal{R})$. In the other case we have $t_i \to^*_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})\cup\mathcal{FC}_{\mathcal{F}^\sharp}(\mathcal{R}_\mathsf{a})} t'_j$ for some i. If t'_j is not terminating with respect to $\mathcal{FC}_{\mathcal{F}}(\mathcal{R})$ then it is also not terminating with respect to $\mathcal{FC}_{\mathcal{F}\cup\mathcal{F}^\sharp}(\mathcal{R})$ and hence t_i is not terminating with respect $\mathcal{FC}_{\mathcal{F}\cup\mathcal{F}^\sharp}(\mathcal{R})$. This contradicts Lemma 7 because t_i is terminating with respect to $\mathcal{R}$, due to minimality. This concludes the proof of minimality. We conclude that $\mathcal{FC}_2(\mathcal{P},\mathcal{R})$ is not finite, which settles the "if" direction.

For the "only if" direction, suppose that

$$s_1 \xrightarrow{\epsilon}_{\mathcal{FC}_2(\mathcal{P})} t_1 \to^*_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})} s_2 \xrightarrow{\epsilon}_{\mathcal{FC}_2(\mathcal{P})} t_2 \to^*_{\mathcal{FC}_{\mathcal{F}}(\mathcal{R})} \cdots$$

is a minimal sequence with respect to $\mathcal{FC}_2(\mathcal{P},\mathcal{R})$. Without loss of generality we assume that symbols from $\mathcal{F}^\sharp$ occur exclusively at root positions. Using the termination equivalence of $\mathcal{FC}_{\mathcal{F}}(\mathcal{R})$ and $\mathcal{FC}_{\mathcal{F}\cup\mathcal{F}^\sharp}(\mathcal{R})$, which is a consequence of Lemma 7, and the fact that every t_i is terminating with respect to $\mathcal{FC}_{\mathcal{F}}(\mathcal{R})$, it follows that this sequence contains infinitely many steps in $\mathcal{P}$. Since every rule in $\mathcal{FC}_{\mathcal{F}\cup\mathcal{F}^\sharp}(\mathcal{R})$ is simulated by a rule in $\mathcal{R}$, the sequence is a sequence in $(\mathcal{P},\mathcal{R})$. After dropping the (possibly empty) initial steps using rules from $\mathcal{R}_\mathsf{a}$, we obtain a sequence in $(\mathcal{P},\mathcal{R})$ which is easily shown to be minimal. □

Theorem 18. *The DP problem $(\mathcal{P},\mathcal{R})$ is finite if and only if the DP problem $\mathcal{FC}_2(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$ is finite.*

Proof. According to Lemma 17 finiteness of $(\mathcal{P},\mathcal{R})$ is equivalent to finiteness of $\mathcal{FC}_2(\mathcal{P},\mathcal{R})$. The equivalence of the latter with finiteness of $\mathcal{FC}_2(\mathcal{P},\mathcal{R})_{\mathsf{rl}}$ follows as in the proof of Theorem 15. □

5 Touzet's SRS

The first approach detailed in the preceding section, $\mathcal{FC}_1(\cdot)_{\mathsf{rl}}$, was implemented and incorporated into the 2007 competition version of $\mathsf{TTT2}$. Together with linear polynomial interpretations with coefficients from $\{0, 1\}$ and standard dependency pair refinements (usable rules with argument filtering [9], recursive SCC [10]), $\mathsf{TTT2}$ could automatically prove termination of `z090.srs`, which is an example from Touzet [21] of a simply terminating SRS whose derivational complexity is not primitive recursive. This prompted Johannes Waldmann to write

> "I find this astonishing: (link to url omitted)
> To my knowledge, this would be the first automatic proof
> for an SRS with non-primitive-recursive complexity"

on the termtools[3] mailing list (7 June 2007).

The SRS $\mathcal{T}$ consists of the rules

$$\begin{array}{llll} \mathsf{bu} \to \mathsf{bs} & \mathsf{sbs} \to \mathsf{bt} & \mathsf{tb} \to \mathsf{bs} & \mathsf{ts} \to \mathsf{tt} \\ \mathsf{sb} \to \mathsf{bsss} & \mathsf{su} \to \mathsf{ss} & \mathsf{tbs} \to \mathsf{utb} & \mathsf{tu} \to \mathsf{ut} \end{array}$$

and simulates the following process on fixed-length lists of natural numbers (s denotes successor and b separates numbers):

$$\begin{aligned} (\cdots, n+1, m, \cdots) &\to (\cdots, n, m+3, \cdots) \\ (\cdots, n+1, m+1, k, \cdots) &\to (\cdots, n, k, m+1, \cdots) \end{aligned}$$

Moreover, the function

$$\phi\colon (x, y) \mapsto \max\{\, z \mid (\, y+1, \overbrace{0, \cdots, 0}^{2x+1}) \to^* (\, \overbrace{0, \cdots, 0}^{2x+1}, z+1\,)\,\}$$

dominates the Ackermann function (Touzet [21]), which proves that the derivational complexity of $\mathcal{T}$ is not primitive recursive. The (simple) termination of $\mathcal{T}$ is shown in [21] by a complicated ad-hoc argument.

Below we present some details of the termination proof generated by $\mathsf{TTT2}$. The SRS $\mathcal{T}$ has the following 17 dependency pairs:

$$\begin{array}{llllllll} \mathsf{Bu} \to \mathsf{Bs} & (1) & \mathsf{Sb} \to \mathsf{S} & (6) & \mathsf{Su} \to \mathsf{S} & (10) & \mathsf{Tbs} \to \mathsf{B} & (14) \\ \mathsf{Bu} \to \mathsf{S} & (2) & \mathsf{Sbs} \to \mathsf{Bt} & (7) & \mathsf{Tb} \to \mathsf{Bs} & (11) & \mathsf{Ts} \to \mathsf{Tt} & (15) \\ \mathsf{Sb} \to \mathsf{Bsss} & (3) & \mathsf{Sbs} \to \mathsf{T} & (8) & \mathsf{Tb} \to \mathsf{S} & (12) & \mathsf{Ts} \to \mathsf{T} & (16) \\ \mathsf{Sb} \to \mathsf{Sss} & (4) & \mathsf{Su} \to \mathsf{Ss} & (9) & \mathsf{Tbs} \to \mathsf{Tb} & (13) & \mathsf{Tu} \to \mathsf{T} & (17) \\ \mathsf{Sb} \to \mathsf{Ss} & (5) & & & & & & \end{array}$$

In the first step of the proof, the ensuing DP problem $(\mathsf{DP}(\mathcal{T}), \mathcal{T})$ is subjected to the interpretations $[\mathsf{B}](x) = [\mathsf{S}](x) = [\mathsf{T}](x) = [\mathsf{s}](x) = [\mathsf{t}](x) = [\mathsf{u}](x) =$

[3] `http://lists.lri.fr/pipermail/termtools/`

x and $[\mathsf{b}](x) = x + 1$. which causes the pairs (3)–(8), (11), (12), (14) to be eliminated. The eight remaining dependency pairs give rise to three SCCs: $\{(1)\}$, $\{(9), (10)\}$, and $\{(13), (15)\text{–}(17)\}$. The first two are easily handled. The last one is the problematic one. Dependency pairs (15) and (16), subsequently followed by pair (17), are removed by using the following interpretations (and considering the induced usable rules):

- $[\mathsf{T}](x) = [\mathsf{u}](x) = x$, $[\mathsf{b}](x) = [\mathsf{t}](x) = 0$, and $[\mathsf{s}](x) = x + 1$,
- $[\mathsf{T}](x) = x$, $[\mathsf{u}](x) = x + 1$, and $[\mathsf{b}](x) = [\mathsf{s}](x) = [\mathsf{t}](x) = 0$.

The remaining DP problem $(\{(13)\}, \mathcal{T})$ is very resistant against automatic termination proof methods, even though it has just one dependency pair. This is the point where root-labeling comes into play. Since $\mathcal{T}_{\mathsf{a}}$ contains five of the eight rules of $\mathcal{T}$, the second component of the flat context closure

$$\mathcal{FC}_1(\{(13)\}, \mathcal{T}) = (\{\mathsf{T}\triangle\mathsf{bs} \to \mathsf{T}\triangle\mathsf{b}\}, \mathcal{FC}_{\{\mathsf{b},\mathsf{s},\mathsf{t},\mathsf{u},\triangle\}}(\mathcal{T}))$$

consists of 28 rewrite rules. As there are four symbols in the carrier for the labeling step, $\mathcal{FC}_1(\{(13)\}, \mathcal{T})_{\mathsf{rl}} = (\mathcal{P}, \mathcal{R})$ with 112 rules in $\mathcal{R}$, which the reader will be spared, and $\mathcal{P}$ consisting of the rules

$$\mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{s}}\mathsf{s}_{\mathsf{b}} \to \mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{b}} \quad (a) \qquad \mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{s}}\mathsf{s}_{\mathsf{t}} \to \mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{t}} \quad (c)$$
$$\mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{s}}\mathsf{s}_{\mathsf{s}} \to \mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{s}} \quad (b) \qquad \mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{s}}\mathsf{s}_{\mathsf{u}} \to \mathsf{T}\triangle_{\mathsf{b}}\mathsf{b}_{\mathsf{u}} \quad (d)$$

Rule (a) is eliminated as it is not part of any SCC. By counting function symbols, the 112 rules in $\mathcal{R}$ are successively reduced to 104 ($\triangle_{\mathsf{s}}$ is counted), 92 ($\triangle_{\mathsf{t}}$ is counted), and 78 (u_{s} is counted) rules. Now all rules of $\mathcal{R}$ that hindered the automatic orientation of the rules (b)–(d) were removed and the termination proof of $\mathcal{T}$ is concluded by using the following interpretations:

$$\begin{aligned}
[\mathsf{s}_{\mathsf{s}}](x) &= [\mathsf{s}_{\mathsf{t}}](x) = [\mathsf{s}_{\mathsf{u}}](x) = x + 1\\
[\mathsf{T}](x) &= [\triangle_{\mathsf{b}}](x) = [\mathsf{b}_{\mathsf{s}}](x) = [\mathsf{b}_{\mathsf{t}}](x) = [\mathsf{b}_{\mathsf{u}}](x) = x\\
[\mathsf{b}_{\mathsf{b}}](x) &= [\mathsf{s}_{\mathsf{b}}](x) = [\mathsf{s}_{\mathsf{s}}](x) = [\mathsf{s}_{\mathsf{t}}](x) = [\mathsf{s}_{\mathsf{u}}](x) = [\mathsf{t}_{\mathsf{b}}](x) = [\mathsf{t}_{\mathsf{s}}](x)\\
&= [\mathsf{t}_{\mathsf{t}}](x) = [\mathsf{t}_{\mathsf{u}}](x) = [\mathsf{u}_{\mathsf{b}}](x) = [\mathsf{u}_{\mathsf{t}}](x) = [\mathsf{u}_{\mathsf{u}}](x) = 0
\end{aligned}$$

Inspired by the success of $\mathsf{TTT2}$ on Touzet's SRS, Hans Zantema announced on the termtools mailing list (16 August 2007) a much simpler example of a simply terminating SRS whose derivational complexity is not primitive recursive:

$$\mathsf{ab} \to \mathsf{baa} \qquad \mathsf{abb} \to \mathsf{bc} \qquad \mathsf{ca} \to \mathsf{ac} \qquad \mathsf{cb} \to \mathsf{bb}$$

This SRS can be automatically proved terminating by Torpa [25] (and several other tools as well), without using root-labeling.

6 Experiments

Extensive tests were conducted to evaluate the usefulness of the root-labeling processors. We used the rewrite systems in version 4.0 of the Termination Prob-

Table 1. Experimental results for SRSs.

			$\mathcal{FC}_1(\cdot)_{\mathsf{rl}}$	$\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$	$\mathcal{FC}_2(\cdot)^*_{\mathsf{rl}}$
P(1)		25 (1.12)	94 (2.30)	98 (2.31)	119 (3.03)
P(1)	with usable rules	37 (0.06)	112 (1.84)	118 (2.32)	138 (3.00)
P(2)		49 (0.63)	144 (4.14)	148 (4.05)	172 (6.84)
P(2)	with usable rules	57 (0.53)	155 (4.01)	169 (4.40)	188 (6.33)
P(1;2)		50 (0.80)	160 (3.40)	165 (3.36)	198 (5.28)
P(1;2)	with usable rules	57 (3.35)	171 (3.10)	183 (3.54)	209 (4.92)
M(2,1)	with usable rules	87 (0.73)	181 (3.61)	181 (4.12)	193 (4.36)
M(3,1)	with usable rules	109 (2.99)	151 (10.67)	158 (7.98)	163 (7.50)
total number of proofs		118	214	220	241

lems Data Base,[4] extended with the secret systems of the 2007 termination competition, which amount to 1381 TRSs and 724 SRSs. All tests were performed on a workstation equipped with an Intel® Pentium™ M processor running at a CPU rate of 2 GHz on 1 GB of system memory and with a time limit of 60 seconds.

Our results for SRSs are summarized in Table 1. Numbers in parentheses indicate the average time (in seconds) to prove termination. For every entry in the table the dependency pair framework with common processors based on an estimation of the dependency graph, the recursive SCC algorithm [10], and the rule removal processor [8] (which in the case of P(1) amounts to counting certain function symbols) together with some reduction pair processor are used. Different rows in the table correspond to different reduction pair processors based on polynomial (P) and matrix (M) interpretations. For polynomial interpretations the numbers in parentheses indicate how many bits are used for coefficients in the SAT-encoding described in [6]. Rows with P(1;2) indicate that 2 bits are used only when no progress can be made with 1 bit. This is faster (and thus more powerful) than P(2). For matrix interpretations [5] the first number in parentheses provides the dimension of the matrices and the second number denotes the number of bits used for the matrix elements in the SAT-encoding. In rows containing 'with usable rules' the respective processor is used to orient the usable rules of $\mathcal{R}$ with respect to the implicit argument filtering obtained from the 0 coefficients in the interpretations [9], as opposed to all rules of the $\mathcal{R}$ component of a DP problem $(\mathcal{P}, \mathcal{R})$.

In columns $\mathcal{FC}_1(\cdot)_{\mathsf{rl}}$ and $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$ the two root-labeling processors are executed as soon as the processors described in the preceding paragraph no longer make progress. In the final column an optimization of the $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$ processor is used which takes effect when root-labeling an already labeled SRS is attempted. This is described at the end of this section and goes back to the implementation of root-labeling in Jambox. We now comment on the obtained data.

[4] `www.lri.fr/~marche/tpdb`

Table 2. Experimental results for TRSs.

			$\mathcal{FC}_1(\cdot)_{\mathsf{rl}}$	$\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$
P(1)	with usable rules	549 (0.16)	636 (0.61)	631 (0.39)
P(2)	with usable rules	584 (0.52)	663 (0.92)	665 (0.93)
M(2,1)	with usable rules	666 (0.57)	697 (0.91)	700 (0.88)
M(3,1)	with usable rules	662 (1.58)	680 (1.94)	684 (2.05)
M(2,2)	with usable rules	648 (2.17)	664 (2.62)	668 (2.74)
M(3,2)	with usable rules	608 (4.29)	613 (4.49)	614 (4.51)
total number of proofs		686	716	719

– First of all, when increasing the number of bits for polynomial or matrix interpretations, it can happen that systems which can be proved using smaller values, are no longer handled due to a timeout in the SAT solver (in our case Minisat [3]).
– Due to the nature of root-labeling, every run that does not succeed to prove termination results in a timeout or a memory overflow.
– On first sight it seems that $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$ is strictly stronger than $\mathcal{FC}_1(\cdot)_{\mathsf{rl}}$. However, what is not apparent from the table is that for every row there are a number of systems which can be proved terminating using $\mathcal{FC}_1(\cdot)_{\mathsf{rl}}$ but not using $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$. These numbers are 1, 1, 5, 2, 4, 2, 9, and 6 (top to bottom).

Our results for TRSs are summarized in Table 2. Most of the remarks for SRSs also hold for TRSs. The optimization for SRSs detailed below is however not implemented for TRSs (which is apparent from the missing fourth column).

We conclude this experimental section with a brief description of the optimized root-labeling processor $\mathcal{FC}_2(\cdot)^*_{\mathsf{rl}}$. During the execution of $\mathcal{FC}_1(\cdot)_{\mathsf{rl}}$ and $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$, function symbols of an already labeled DP problem do not have structure, which entails that both closure under flat contexts as well as the actual root-labeling steps create many more new rules due the increased size of the labeled signature. The implementation of plain root-labeling for SRSs in Jambox reduces the number of new rules by using the original signature in subsequent root-labeling steps. This is best explained on a concrete example. When labeling the rule $\mathsf{a_b b_a} \to \mathsf{a_c c_a}$, the unlabeled rule $\mathsf{ab} \to \mathsf{ac}$ is closed under flat contexts

$$\mathsf{aab} \to \mathsf{aac} \qquad \mathsf{bab} \to \mathsf{bac} \qquad \mathsf{cab} \to \mathsf{cac}$$

and subsequently labeled by propagating all one symbol extensions (aa, ab, ac) of the original starting assignment a, resulting in the nine rules:[5]

$$\begin{array}{lll}
\mathsf{a_{ab}a_{ba}b_{aa}} \to \mathsf{a_{ac}a_{ca}c_{aa}} & \mathsf{a_{ab}a_{ba}b_{ab}} \to \mathsf{a_{ac}a_{ca}c_{ab}} & \mathsf{a_{ab}a_{ba}b_{ac}} \to \mathsf{a_{ac}a_{ca}c_{ac}} \\
\mathsf{b_{ab}a_{ba}b_{aa}} \to \mathsf{b_{ac}a_{ca}c_{aa}} & \mathsf{b_{ab}a_{ba}b_{ab}} \to \mathsf{b_{ac}a_{ca}c_{ab}} & \mathsf{b_{ab}a_{ba}b_{ac}} \to \mathsf{b_{ac}a_{ca}c_{ac}} \\
\mathsf{c_{ab}a_{ba}b_{aa}} \to \mathsf{c_{ac}a_{ca}c_{aa}} & \mathsf{c_{ab}a_{ba}b_{ab}} \to \mathsf{c_{ac}a_{ca}c_{ab}} & \mathsf{c_{ab}a_{ba}b_{ac}} \to \mathsf{c_{ac}a_{ca}c_{ac}}
\end{array}$$

[5] The transformation $\mathcal{FC}_2(\cdot)^*_{\mathsf{rl}}$ is easily formalized, starting from the $\mathcal{F}$-algebra $\mathcal{A}_{\mathcal{F}\times\mathcal{F}}$ in Definition 3 with $f_{\mathcal{A}_{\mathcal{F}\times\mathcal{F}}}(a,b) = (f,a)$.

With $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$ we would obtain at least sixteen rules. Although the numbers in Table 1 may suggest otherwise, $\mathcal{FC}_2(\cdot)^*_{\mathsf{rl}}$ does not subsume $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$. For instance, three (`x02.srs`, `z108.srs`, and `z114.srs`) of the 181 systems in the $\mathcal{FC}_2(\cdot)_{\mathsf{rl}}$ entry of the row '$\mathsf{M}(2,1)$ with usable rules' are not included in the 193 systems in the $\mathcal{FC}_2(\cdot)^*_{\mathsf{rl}}$ entry.

7 Summary and Future Work

In this paper we introduced the technique of root-labeling for TRSs into the dependency pair framework, resulting in two different root-labeling processors. Touzet's example showed the usefulness of these processors for SRSs. Although root-labeling is trivial to automate, further research is needed to determine when to apply it. Besides, it is unclear whether one of the two introduced processors is (at least in theory) strictly stronger than the other. As explained in the experimental section, the current implementation in $\mathsf{TTT2}$ applies root-labeling as soon as the other processors do not make progress. In particular, we never undo the effect of root-labeling. Since the root-labeling processors are always applicable, the DP problems quickly grow too large. This is especially a problem for TRSs, where a single application of root-labeling typically blows up the system. A related problem is the possibility to verify the correctness of the produced termination proofs.

Acknowledgments. We are grateful to Johannes Waldmann for inventing root-labeling (for string rewrite systems) and for drawing our attention to the termination proof of `z090.srs` that $\mathsf{TTT2}$ produced during the 2007 international termination competition. Without these contributions, this paper would not have been written. Discussions with Jörg Endrullis and Johannes Waldmann improved our results.

References

1. T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.
2. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
3. N. Eén and N. Sörensson. An extensible SAT-solver. In *Proc. 6th SAT*, volume 2919 of *LNCS*, pages 502–518, 2003.
4. J. Endrullis. Jambox, 2005. Available from `http://joerg.endrullis.de`.
5. J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. In *Proc. 3rd IJCAR*, volume 4130 of *LNAI*, pages 574–588, 2006.
6. C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl. SAT solving for termination analysis with polynomial interpretations. In *Proc. 10th SAT*, volume 4501 of *LNCS*, pages 340–354, 2007.

7. J. Giesl, P. Schneider-Kamp, and R. Thiemann. AProVE 1.2: Automatic termination proofs in the dependency pair framework. In *Proc. 3rd IJCAR*, volume 4130 of *LNAI*, pages 281–286, 2006.
8. J. Giesl, R. Thiemann, and P. Schneider-Kamp. The dependency pair framework: Combining techniques for automated termination proofs. In *Proc. 11th LPAR*, volume 3425 of *LNAI*, pages 301–331, 2004.
9. J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and improving dependency pairs. *Journal of Automated Reasoning*, 37(3):155–203, 2006.
10. N. Hirokawa and A. Middeldorp. Automating the dependency pair method. *Information and Computation*, 199(1,2):172–199, 2005.
11. N. Hirokawa and A. Middeldorp. Predictive labeling. In *Proc. 17th RTA*, volume 4098 of *LNCS*, pages 313–327, 2006.
12. D. Hofbauer. MultumNonMulta, 2006. Available from `www.theory.informatik.uni-kassel.de/~dieter/multum/`.
13. A. Koprowski. TPA: Termination proved automatically. In *Proc. 17th RTA*, volume 4098 of *LNCS*, pages 275–266, 2006.
14. A. Koprowski and A. Middeldorp. Predictive labeling with dependency pairs using SAT. In *Proc. 21st CADE*, volume 4603 of *LNAI*, pages 410–425, 2007.
15. A. Koprowski and H. Zantema. Recursive path ordering for infinite labelled rewrite systems. In *Proc. 3rd IJCAR*, volume 4130 of *LNAI*, pages 332–346, 2006.
16. M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean termination tool 2, 2007. Available from `http://colo6-c703.uibk.ac.at/ttt2`.
17. A. Middeldorp, H. Ohsaki, and H. Zantema. Transforming termination by self-labelling. In *Proc. 13th CADE*, volume 1104 of *LNAI*, pages 373–386, 1996.
18. Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
19. R. Thiemann. *The DP Framework for Proving Termination of Term Rewriting*. PhD thesis, RWTH Aachen, 2007. Available as technical report AIB-2007-17.
20. R. Thiemann and A. Middeldorp. Innermost termination of rewrite systems by labeling. In *Proc. 7th WRS*, volume 204 of *ENTCS*, pages 3–19, 2008.
21. H. Touzet. A complex example of a simplifying rewrite system. In *Proc. 25th ICALP*, volume 1443 of *LNCS*, pages 507–517, 1998.
22. Y. Toyama. Counterexamples to the termination for the direct sum of term rewriting systems. *Information Processing Letters*, 25:141–143, 1987.
23. J. Waldmann. Matchbox: A tool for match-bounded string rewriting. In *Proc. 15th RTA*, volume 3091 of *LNCS*, pages 85–94, 2004.
24. H. Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24:89–105, 1995.
25. H. Zantema. TORPA: Termination of rewriting proved automatically. In *Proc. 15th RTA*, volume 3091 of *LNCS*, pages 95–104, 2004.
26. H. Zantema and J. Waldmann. Termination by quasi-periodic interpretations. In *Proc. 18th RTA*, volume 4533 of *LNCS*, pages 404–418, 2007.