

Total termination of term rewriting is undecidable

Hans Zantema

Utrecht University, Department of Computer Science,
P.O. box 80.089, 3508 TB Utrecht, The Netherlands
e-mail: hansz@cs.ruu.nl

Abstract

Usually termination of term rewriting systems (TRS's) is proved by means of a monotonic well-founded order. If this order is total on ground terms, the TRS is called totally terminating. In this paper we prove that total termination is an undecidable property of finite term rewriting systems. The proof is given by means of Post's Correspondence Problem.

1 Introduction

Termination of term rewriting systems (TRS's) is an important property. Often termination proofs are given by defining an order that is well-founded, and proving that for every rewrite step the value of the term decreases according this order. In many cases the order is monotonic, and it suffices to prove that $l^\sigma > r^\sigma$ for all rewrite rules $l \rightarrow r$ and all ground substitutions σ . Standard techniques following this approach include recursive path order and Knuth-Bendix order, see for example [17]. It is an interesting question whether these orders are total or can be extended to a total monotonic order, or are essentially non-total. It turns out that these standard orders are all total or can be extended to a total monotonic order ([4]), while for terminating systems like

$$f(a) \rightarrow f(b)$$

$$g(b) \rightarrow g(a)$$

it is essentially impossible to orient a and b . A TRS is called *totally terminating* if it is compatible with a monotonic well-founded total order on ground terms. Hence the above example is not totally terminating, while all TRS's of which termination can be proved by means of standard orders are totally terminating. Total termination was introduced in [18, 20] in a more semantical way. An extensive examination of total termination has been given in [4, 5, 6], including the proof that the semantical definition and the order definition are equivalent.

It is well-known that termination is an undecidable property of finite TRS's. The first proof was given in [8]. It has even been proved ([2, 11]) that termination is an undecidable property of single rewrite rules. A TRS is called *simply terminating* if it is compatible with a simplification order, i.e., a well-founded monotonic order possessing the subterm property. Simple termination is stronger than termination, but weaker than total termination. Simple termination has been proved to be undecidable ([1]), even for single rewrite rules ([12]).

In this paper we prove that total termination is undecidable too ¹. As in [11] we give a transformation from an arbitrary instance of Post's Correspondence Problem to a TRS. It is a rather straightforward observation that the TRS is terminating if and only if it is simply terminating, if and only if the instance of Post's Correspondence Problem has no solution. This gives a new proof of undecidability of simple termination. The main part of this paper consists of the proof that the TRS is also totally terminating whenever it is terminating, hence proving undecidability of total termination. This is proved by constructing a suitable monotonic well-founded total order $>$ on ground terms. This construction needs two auxiliary orders \sqsupset and \triangleright and an auxiliary TRS S . Let N denote the normal form with respect to S , then the final order is defined by

$$t > u \iff N(t) \sqsupset N(u) \vee (N(t) = N(u) \wedge t \triangleright u).$$

In section 2 we give some preliminaries. In section 3 we present the construction of the TRS from an arbitrary instance of Post's Correspondence Problem to a TRS. In section 4 we define the auxiliary order \sqsupset and derive some of its properties. In section 5 we define the auxiliary order \triangleright and prove that the order $>$ satisfies all required properties.

2 Preliminaries

Let \mathcal{F} be a signature containing at least one constant. We write $\mathcal{T}(\mathcal{F})$ for the set of ground terms over \mathcal{F} . Any transitive irreflexive relation is called an *order*. An order $>$ is *total* if for any two distinct elements t, u one has either $t > u$ or $u > t$. An order $>$ is called *well-founded* if there is no infinite descending sequence

$$t_1 > t_2 > t_3 > t_4 > \dots$$

An order $>$ on $\mathcal{T}(\mathcal{F})$ is called *monotonic* if

$$t > u \Rightarrow F(\dots, t, \dots) > F(\dots, u, \dots)$$

¹In September 1994 Albert Rubio conjectured that total termination is decidable and Jean-Pierre Jouannaud conjectured that total termination is undecidable. They arrived at a wager: if within two years undecidability of total termination was proved Albert Rubio was willing to offer one bottle of champagne. Jean-Pierre Jouannaud however was more sure of his conjecture: if within the same term the contrary was proved he was willing to offer a box of bottles of champagne. This paper shows that Jean-Pierre Jouannaud was right

for all $F \in \mathcal{F}$. A TRS R and an order $>$ are called *compatible* if $t > u$ for all rewrite steps $t \rightarrow_R u$. For compatibility with a monotonic order it suffices to check that $l^\sigma > r^\sigma$ for all rules $l \rightarrow r$ in R and all ground substitutions σ . It is well-known that a TRS is terminating if and only if it is compatible with some monotonic well-founded order. An order $>$ on $\mathcal{T}(\mathcal{F})$ is said to have the *subterm property* if

$$F(\dots, t, \dots) > t$$

for all $F \in \mathcal{F}$ and $t \in \mathcal{T}(\mathcal{F})$. A monotonic order satisfying the subterm property is called a *simplification order*. A direct consequence of Kruskal's theorem ([10, 7]) is that any simplification order over a finite signature is well-founded. A TRS over a finite signature is called *simply terminating* if it is compatible with a simplification order. In [13] it is described how simplification orders extend to infinite signatures.

A TRS is called *length-preserving* if $|l^\sigma| = |r^\sigma|$ for all rules $l \rightarrow r$ and all ground substitutions σ . Here $|t|$ denotes the number of operation symbols. It is not difficult to prove that any length-preserving terminating TRS is simply terminating.

Definition 1 *A TRS is called totally terminating if it is compatible with a monotonic well-founded total order.*

As remarked above every terminating TRS is compatible with a monotonic well-founded order. On the other hand, using this result and Zorn's lemma it is possible to show that every terminating TRS is compatible with a total well-founded order, see [3]. Hence it is the combination of totality and monotonicity that makes the notion of total termination stronger than termination.

The following well-known proposition (see e.g. [16], proposition 2.3.4) states that the well-foundedness condition may be replaced by the subterm property.

Proposition 2 *Let $>$ be any monotonic total order on ground terms over a finite signature. Then $>$ is well-founded if and only if it has the subterm property.*

Proof: For the 'if'-part assume $>$ has the subterm property. Since it is also monotonic it contains the homeomorphic embedding. Kruskal's theorem states that any order extending the homeomorphic embedding is well-founded, hence $>$ is well-founded.

For the 'only if'-part assume $>$ has not the subterm property. Then there exist terms t and $F(\dots, t, \dots)$ such that not $F(\dots, t, \dots) > t$. Since the terms are distinct and the order is total we obtain $t > F(\dots, t, \dots)$. Write $C[t] = F(\dots, t, \dots)$. From monotonicity we obtain

$$t > C[t] > C[C[t]] > C[C[C[t]]] > \dots$$

contradicting well-foundedness. \square

The ‘if’-part of this proposition will be used to prove well-foundedness of the order we construct in this paper; the ‘only if’-part implies that any totally terminating TRS is simply terminating. We give two standard examples. The TRS

$$f(f(x)) \rightarrow f(g(f(x)))$$

is terminating but not simply terminating. The TRS

$$f(a) \rightarrow f(b)$$

$$g(b) \rightarrow g(a)$$

is simply terminating since it is length preserving, but not totally terminating since a and b are incomparable.

In our order construction we will use the *lexicographic path order*, a well-known variant of the recursive path order introduced in [9]. It is defined recursively as follows. Let \gg be any order on the signature \mathcal{F} . Then for two ground terms $t = F(t_1, \dots, t_n)$ and $u = G(u_1, \dots, u_m)$ one has $t >_{lpo} u$ if and only if

- $t_i = u$ or $t_i >_{lpo} u$ for some $i = 1, \dots, n$, or
- $F \gg G$ and $t >_{lpo} u_i$ for all $i = 1, \dots, m$, or
- $F = G$ and $(t_1, \dots, t_n) >_{lpo}^{lex} (u_1, \dots, u_m)$.

Here for any order $>$ the order $>^{lex}$ means the lexicographic extension of $>$ to sequences. The lexicographic comparison has to be done in a fixed direction; in the paper it will be from right to left. Note that only sequences of equal length are compared, since we require that every symbol has a fixed arity. It is well-known that $>_{lpo}$ is monotonic and has the subterm property. Further $>_{lpo}$ is total on ground terms if and only if \gg is total on \mathcal{F} .

3 The construction of the TRS

Given a finite alphabet Σ , and a finite set $P \subset \Sigma^* \times \Sigma^*$ it is undecidable whether there exists some natural number $n > 0$ and $(\alpha_i, \beta_i) \in P$ for $i = 1, \dots, n$ such that

$$\alpha_1 \alpha_2 \cdots \alpha_n = \beta_1 \beta_2 \cdots \beta_n.$$

This problem is referred to as Post’s Correspondence Problem (PCP)². It originates from Emil Post ([14]); an extensive recent investigation can be found in [15]. A standard method to prove undecidability of some new problem is the following.

²Often the equivalent formulation of PCP is used: given $\alpha_1, \alpha_2, \dots, \alpha_n, \beta_1, \beta_2, \dots, \beta_n \in \Sigma^*$, is there a_1, a_2, \dots, a_m such that $\alpha_{a_1} \alpha_{a_2} \cdots \alpha_{a_m} = \beta_{a_1} \beta_{a_2} \cdots \beta_{a_m}$? Our formulation however needs less indexing.

Start with an arbitrary instance P of PCP. Using this instance P , construct an instance of the new problem such that PCP has a solution for P if and only if the constructed instance of the new problem has a solution. If we have such a construction, then we have proved the undecidability of the new problem. We follow this method in proving undecidability of total termination; we only add a negation. We give a construction of a TRS R_P from an arbitrary instance P of PCP such that R_P is totally terminating if and only if PCP has no solution for P .

Before defining R_P we define the signature and define some notation. Let P be an arbitrary instance of PCP over an alphabet Σ . For any symbol $a \in \Sigma$, we introduce two unary function symbols a and \bar{a} . Further we have one function symbol f of arity 4. Finally we assume a constant c whenever we need ground terms. For any string $\alpha = a_1 a_2 \cdots a_n \in \Sigma^*$ and any term t we define

$$\alpha(t) = a_1(a_2(\cdots(a_n(t))\cdots))$$

and

$$\bar{\alpha}(t) = \bar{a}_n(\bar{a}_{n-1}(\cdots(\bar{a}_1(t))\cdots)).$$

Now the TRS R_P consists of the rules

$$\boxed{f(\alpha(x), y, \beta(z), w) \rightarrow f(x, \bar{\alpha}(y), z, \bar{\beta}(w))}$$

for all $(\alpha, \beta) \in P$, and

$$\boxed{f(x, \bar{a}(y), x, \bar{a}(y)) \rightarrow f(a(x), y, a(x), y)}$$

for all $a \in \Sigma$. Now we can state our main theorem, from which undecidability of total termination is a direct consequence.

Theorem 3 *Let P be an arbitrary instance of PCP and let R_P be defined as above. Then the following statements are equivalent:*

1. R_P is totally terminating;
2. R_P is terminating;
3. PCP has no solution for P .

The implication (1) \Rightarrow (2) is trivial. The major part of this paper is devoted to proving the implication (2) \Rightarrow (1). The equivalence between (2) and (3) is immediate from the following two propositions.

Proposition 4 *If PCP has a solution for P then R_P admits an infinite reduction.*

Proof: Using the second kind of rules in R_P it is clear that

$$f(x, \bar{\alpha}(y), x, \bar{\alpha}(y)) \rightarrow_{R_P}^* f(\alpha(x), y, \alpha(x), y)$$

for any string α . Let

$$\alpha = \alpha_1 \alpha_2 \cdots \alpha_n = \beta_1 \beta_2 \cdots \beta_n$$

be a solution of PCP for P . Then we have the infinite reduction

$$\begin{aligned} f(\alpha(x), y, \alpha(x), y) &= f(\alpha_1(\cdots(\alpha_n(x))\cdots), y, \beta_1(\cdots(\beta_n(x))\cdots), y) \\ &\rightarrow_{R_P} f(\alpha_2(\cdots(\alpha_n(x))\cdots), \bar{\alpha}_1(y), \beta_2(\cdots(\beta_n(x))\cdots), \bar{\beta}_1(y)) \\ &\rightarrow_{R_P}^* f(x, \bar{\alpha}_n(\cdots(\bar{\alpha}_1(y))\cdots), x, \bar{\beta}_n(\cdots(\bar{\beta}_1(y))\cdots)) \\ &= f(x, \bar{\alpha}(y), x, \bar{\alpha}(y)) \\ &\rightarrow_{R_P}^* f(\alpha(x), y, \alpha(x), y) \\ &\dots \quad \dots \end{aligned}$$

□

Proposition 5 *If R_P admits an infinite reduction then PCP has a solution for P .*

Proof: We introduce a many-sorted TRS R'_P having the same rules as R_P , but in which there are two distinct sorts s_1 and s_2 . The symbols a and \bar{a} have type $s_1 \rightarrow s_1$, the symbol f has type $s_1 \times s_1 \times s_1 \times s_1 \rightarrow s_2$, all variables have type s_1 . All rules of R_P are well-typed, i.e., there are no type clashes and each left hand side has the same type as the corresponding right hand side. The type elimination result from [19, 20] states that if the TRS does not contain both collapsing and duplicating rules, then the many-sorted version terminates if and only if the one-sorted version terminates. Since R_P contains neither collapsing nor duplicating rules, this result applies here. Assume that R_P admits an infinite reduction, then also R'_P admits an infinite reduction. Such an infinite R'_P -reduction is an infinite R_P -reduction in which the symbol f only occurs as head symbols³. Fix such a reduction. Let R_1 consist of the rules of R_P of the shape $f(\alpha(x), y, \beta(z), w) \rightarrow f(x, \bar{\alpha}(y), z, \bar{\beta}(w))$ and let R_2 consist of the rules of R_P of the shape $f(x, \bar{a}(y), x, \bar{a}(y)) \rightarrow f(a(x), y, a(x), y)$. Since R_1 is terminating, the infinite reduction contains infinitely many R_2 -steps. Since R_2 is terminating, not all of these infinitely many R_2 -steps are subsequent. Hence there exists a reduction of the shape

$$t_1 \rightarrow_{R_2} t_2 \xrightarrow{+}_{R_1} t_3 \rightarrow_{R_2} t_4.$$

³By means of a minimality argument it is also possible to construct such an infinite R_P -reduction in which the symbol f only occurs as head symbols without using type elimination.

Since f only occurs as the head symbol and due the shape of the rules of R_2 we have $t_2 = f(t, t', t, t')$ and $t_3 = f(u, u', u, u')$ for some terms t, t', u, u' not containing f -symbols. Focussing on the first and third argument of f and due to the shape of the rules of R_1 this implies that $t = \alpha_1(\cdots(\alpha_n(u))\cdots) = \beta_1(\cdots(\beta_n(u))\cdots)$. Hence we obtain a solution of PCP for P . \square

Since R_P is length preserving we conclude that R_P is terminating if and only if it is simply terminating. Together with the above two propositions we obtain that R_P is simply terminating if and only if PCP has no solution for P . This proves that simple termination is undecidable. For people not familiar with the undecidability of uniform termination of linear bounded automata this undecidability proof of simple termination is much simpler than the proof given in [1]. It can even be simplified further: for simple termination there is no need to distinguish between a and \bar{a} . Let R'_P be the simplified version of R_P in which all bars have been removed, then we still have with the above proof that R'_P is simply terminating if and only if PCP has no solution for P . However, for this simplified version R'_P there is no equivalence between termination and total termination any more. For example, if $P = \{(01, 10)\}$ then clearly PCP has no solution for P , and R'_P consisting of the rules

$$\begin{aligned} f(0(1(x)), y, 1(0(z)), w) &\rightarrow f(x, 1(0(y)), z, 0(1(w))) \\ f(x, 0(y), x, 0(y)) &\rightarrow f(0(x), y, 0(x), y) \\ f(x, 1(y), x, 1(y)) &\rightarrow f(1(x), y, 1(x), y) \end{aligned}$$

is terminating. Let $>$ be any monotonic order compatible with R'_P , then the assumption $1(0(c)) > 0(1(c))$ yields the contradiction

$$\begin{aligned} f(0(1(c)), c, 1(0(c)), c) &> f(c, 1(0(c)), c, 0(1(c))) > f(c, 0(1(c)), c, 0(1(c))) > \\ f(0(c), 1(c), 0(c), 1(c)) &> f(1(0(c)), c, 1(0(c)), c) > f(0(1(c)), c, 1(0(c)), c) \end{aligned}$$

while the assumption $0(1(c)) > 1(0(c))$ yields the contradiction

$$\begin{aligned} f(0(1(c)), c, 1(0(c)), c) &> f(c, 1(0(c)), c, 0(1(c))) > f(c, 1(0(c)), c, 1(0(c))) > \\ f(1(c), 0(c), 1(c), 0(c)) &> f(0(1(c)), c, 0(1(c)), c) > f(0(1(c)), c, 1(0(c)), c). \end{aligned}$$

We conclude that $>$ is not total, hence R'_P is not totally terminating.

In the next two sections we assume that R_P is terminating and we finally prove that R_P is totally terminating.

4 An auxiliary order

It remains to show that if the system R_P is terminating then it is also totally terminating. In the next section we construct a total well-founded monotonic order $>$ on ground terms such that $l^\sigma > r^\sigma$ for all rules $l \rightarrow r$ of R_P and all ground substitutions σ , proving total termination. In this section we construct an auxiliary order \sqsupset that plays an essential role in the construction of $>$.

We define the TRS S consisting of the rules

$$f(x, \bar{a}(y), z, w) \rightarrow f(a(x), y, z, w)$$

$$f(x, y, z, \bar{a}(w)) \rightarrow f(x, y, a(z), w)$$

for all $a \in \Sigma$. Clearly S is terminating. Further S is confluent since its critical pairs are convergent. For a term t write $N(t)$ for the unique normal form of t with respect to S . The following proposition follows immediately from the definitions.

Proposition 6 *Let $l \rightarrow r$ be a rule of R_P and let σ be a ground substitution. Then $N(l^\sigma) = N(r^\sigma)$.*

The property that we need for the auxiliary order \sqsupset is given in the following theorem.

Theorem 7 *There exists a total order \sqsupset on ground terms such that if $N(t) \sqsupset N(u)$ for ground terms t and u , then*

- $N(a(t)) \sqsupset N(a(u))$ and
- $N(\bar{a}(t)) \sqsupset N(\bar{a}(u))$ and
- $N(f(t, s, s', s'')) \sqsupset N(f(u, s, s', s''))$ and
- $N(f(s, t, s', s'')) \sqsupset N(f(s, u, s', s''))$ and
- $N(f(s, s', t, s'')) \sqsupset N(f(s, s', u, s''))$ and
- $N(f(s, s', s'', t)) \sqsupset N(f(s, s', s'', u))$

for all $a \in \Sigma$, and all ground terms s, s', s'' .

In order to define an order \sqsupset satisfying this theorem we introduce the function **rev** on ground terms that reverses strings of \bar{a} -symbols and does not affect the rest of the term. For example, we want

$$\begin{aligned} \text{rev}(\bar{0}(\bar{1}(f(\bar{0}(\bar{1}(c)), \bar{0}(0(\bar{1}(c))), 0(1(c)), \bar{0}(\bar{0}(\bar{1}(c))))))) &= \\ \bar{1}(\bar{0}(f(\bar{1}(\bar{0}(c)), \bar{0}(0(\bar{1}(c))), 0(1(c)), \bar{1}(\bar{0}(\bar{0}(c)))))). \end{aligned}$$

The function rev can be defined as follows. Any ground term t can uniquely be written as

$$t = \bar{a}_1(\bar{a}_2(\cdots \bar{a}_n(g(t_1, \dots, t_k))\cdots))$$

where either $g = c$ or $g = f$ or $g = a$ for some $a \in \Sigma$. We recursively define

$$\text{rev}(\bar{a}_1(\bar{a}_2(\cdots \bar{a}_n(g(t_1, \dots, t_k))\cdots))) = \bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(g(\text{rev}(t_1), \dots, \text{rev}(t_k))\cdots))).$$

Lemma 8 *The function rev is bijective.*

Proof: One easily proves by induction that $\text{rev}(\text{rev}(t)) = t$ for all ground terms t , hence rev is bijective. \square

Choose a total precedence \gg on the signature satisfying

$$f \gg \bar{a}, \quad c \gg \bar{a}, \quad b \gg \bar{a}$$

for all $a, b \in \Sigma$, and for which

$$\bar{a} \gg \bar{b} \iff a \gg b$$

for all $a, b \in \Sigma$. Let $>_{lpo}$ be the lexicographic path order associated with this precedence, in which the arguments of f are compared lexicographically from right to left. It is well-known that this order $>_{lpo}$ is monotonic and total on ground terms. We still need two lemmas over $>_{lpo}$.

Lemma 9 *Let $\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(t)\cdots)) >_{lpo} \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(t)\cdots))$ for some term t . Then*

$$\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(u)\cdots)) >_{lpo} \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(u)\cdots))$$

and

$$a_n(a_{n-1}(\cdots a_1(u)\cdots)) >_{lpo} b_m(b_{m-1}(\cdots b_1(u)\cdots))$$

for all terms u .

Proof: If $n = 0$ or $m = 0$ then either the assumption is not fulfilled or the lemma is trivial. So assume $n > 0$ and $m > 0$. We proceed by induction on $n + m$. We distinguish three cases.

- $\bar{a}_n = \bar{b}_m$ and $a_n = b_m$. From the assumption we obtain

$$\bar{a}_{n-1}(\cdots \bar{a}_1(t)\cdots) >_{lpo} \bar{b}_{m-1}(\cdots \bar{b}_1(t)\cdots).$$

Now the induction hypothesis and monotonicity of $>_{lpo}$ yields the desired result.

- $\bar{a}_n \gg \bar{b}_m$ and $a_n \gg b_m$. From the assumption we obtain

$$\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(t)\cdot)) >_{lpo} \bar{b}_{m-1}(\cdots \bar{b}_1(t)\cdot).$$

Now we apply the induction hypothesis and use $\bar{a}_n \gg \bar{b}_m$ and $a_n \gg b_m$ to achieve the desired result.

- $\bar{b}_m \gg \bar{a}_n$ and $b_m \gg a_n$. From the assumption we obtain

$$\bar{a}_{n-1}(\cdots \bar{a}_1(t)\cdot) \geq_{lpo} \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(t)\cdot)).$$

Now we apply the induction hypothesis and use the subterm property to achieve the desired result.

□

Lemma 10 *Let $\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(t)\cdot)) >_{lpo} \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(u)\cdot))$ for ground terms t and u whose root symbols are not of the shape \bar{a} for $a \in , .$ Let $a \in , .$ Then*

$$\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(\bar{a}(t))\cdot)) >_{lpo} \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(\bar{a}(u))\cdot)).$$

Proof: From the assumption and the subterm property we obtain

$$\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(t)\cdot)) >_{lpo} u.$$

Since the root symbol of u is greater than all symbols \bar{a}_i with respect to the precedence \gg we obtain $t \geq_{lpo} u$. We distinguish two cases.

- $t = u$. Now the lemma follows from lemma 9.
- $t >_{lpo} u$. Since the root symbol of t is greater than \bar{a} and all symbols \bar{b}_i with respect to the precedence \gg we obtain $t >_{lpo} \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(\bar{a}(u))\cdot))$. Now the lemma follows from the subterm property.

□

Now we define the order \sqsupset :

$$t \sqsupset u \iff \text{rev}(t) >_{lpo} \text{rev}(u).$$

Since $>_{lpo}$ is transitive and irreflexive, the same holds for \sqsupset . Since $>_{lpo}$ is total and rev is injective by lemma 8, the order \sqsupset is total too.

Let $N(t) \sqsupset N(u)$ for ground terms t and u . Write

$$N(t) = \bar{a}_1(\bar{a}_2(\cdots \bar{a}_n(t')\cdot)) \text{ and } N(u) = \bar{b}_1(\bar{b}_2(\cdots \bar{b}_m(u')\cdot))$$

for terms t' and u' having a root symbol not of the shape \bar{a} for $a \in , .$ We have

$$\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(\text{rev}(t'))\cdot)) = \text{rev}(N(t)) >_{lpo} \text{rev}(N(u)) = \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(\text{rev}(u'))\cdot)).$$

Now we prove the remaining proof obligations of theorem 7.

- $\text{rev}(N(a(t))) = \text{rev}(a(N(t))) = a(\text{rev}(N(t))) >_{lpo}$

$$a(\text{rev}(N(u))) = \text{rev}(a(N(u))) = \text{rev}(N(a(u)))$$

by monotonicity of $>_{lpo}$, hence $N(a(t)) \sqsupset N(a(u))$.

- $\text{rev}(N(\bar{a}(t))) = \text{rev}(\bar{a}(N(t))) = \bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(\bar{a}(\text{rev}(t'))))\cdots) >_{lpo}$

$$\bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(\bar{a}(\text{rev}(u'))))\cdots) = \text{rev}(\bar{a}(N(u))) = \text{rev}(N(\bar{a}(u)))$$

by lemma 10, hence $N(\bar{a}(t)) \sqsupset N(\bar{a}(u))$.

- In $N(f(t, s, s', s''))$ write $s = \bar{c}_1(\bar{c}_2(\cdots \bar{c}_k(\tilde{s}))\cdots)$ where the root symbol of \tilde{s} is not of the shape \bar{a} for $a \in , .$ Using monotonicity of $>_{lpo}$ we obtain

$$\begin{aligned} \text{rev}(N(f(t, s, s', s''))) &= \text{rev}(f(c_k(\cdots (c_1(N(t))))\cdots), N(\tilde{s}), \dots, \dots) \\ &= f(c_k(\cdots (c_1(\text{rev}(N(t))))\cdots), \text{rev}(N(\tilde{s})), \dots, \dots) \\ &>_{lpo} f(c_k(\cdots (c_1(\text{rev}(N(u))))\cdots), \text{rev}(N(\tilde{s})), \dots, \dots) \\ &= \text{rev}(f(c_k(\cdots (c_1(N(u))))\cdots), N(\tilde{s}), \dots, \dots) \\ &= \text{rev}(N(f(u, s, s', s''))), \end{aligned}$$

hence $N(f(t, s, s', s'')) \sqsupset N(f(u, s, s', s''))$. The proof of $N(f(s, s', t, s'')) \sqsupset N(f(s, s', u, s''))$ is similar.

- As in lemma 10 we conclude from

$$\bar{a}_n(\bar{a}_{n-1}(\cdots \bar{a}_1(\text{rev}(t'))\cdots) >_{lpo} \bar{b}_m(\bar{b}_{m-1}(\cdots \bar{b}_1(\text{rev}(u'))\cdots))$$

that $\text{rev}(t') \geq_{lpo} \text{rev}(u')$. In the following we apply the fact that the arguments of f are compared lexicographically from right to left in the case that $\text{rev}(t') >_{lpo} \text{rev}(u')$, in the case of $\text{rev}(t') = \text{rev}(u')$ we apply lemma 9:

$$\begin{aligned} \text{rev}(N(f(s, t, s', s''))) &= \text{rev}(N(f(s, N(t), s', s''))) \\ &= \text{rev}(N(f(s, \bar{a}_1(\bar{a}_2(\cdots \bar{a}_n(t'))\cdots), s', s''))) \\ &= \text{rev}(f(a_n(a_{n-1}(\cdots a_1(N(s))))\cdots), t', \dots, \dots) \\ &= f(a_n(a_{n-1}(\cdots a_1(\text{rev}(N(s))))\cdots), \text{rev}(t'), \dots, \dots) \\ &>_{lpo} f(b_m(b_{m-1}(\cdots b_1(\text{rev}(N(s))))\cdots), \text{rev}(u'), \dots, \dots) \\ &= \text{rev}(f(b_m(b_{m-1}(\cdots b_1(N(s))))\cdots), u', \dots, \dots) \\ &= \text{rev}(N(f(s, \bar{b}_1(\bar{b}_2(\cdots \bar{b}_m(u'))\cdots), s', s''))) \\ &= \text{rev}(N(f(s, N(u), s', s''))) \\ &= \text{rev}(N(f(s, u, s', s''))). \end{aligned}$$

Hence $N(f(s, t, s', s'')) \sqsupset N(f(s, u, s', s''))$. Note that for this part of the proof the introduction of rev is essential. The proof of $N(f(s, s', s'', t)) \sqsupset N(f(s, s', s'', u))$ is similar.

This concludes the proof of theorem 7.

To achieve the subterm property for our final order we need the following version of the subterm property for \sqsupset .

Proposition 11 *Let t be any ground term. Then*

- $N(a(t)) \sqsupset N(t)$ and
- $N(\bar{a}(t)) \sqsupset N(t)$ and
- $N(f(t, s, s', s'')) \sqsupset N(t)$ and
- $N(f(s, t, s', s'')) \sqsupset N(t)$ and
- $N(f(s, s', t, s'')) \sqsupset N(t)$ and
- $N(f(s, s', s'', t)) \sqsupset N(t)$

for all $a \in \Sigma$, and all ground terms s, s', s'' .

Proof: As before write $N(t) = \bar{a}_1(\bar{a}_2(\dots\bar{a}_n(t')\cdot))$ for a term t' having a root symbol not of the shape \bar{a} for $a \in \Sigma$. By monotonicity and subterm property of $>_{lpo}$ we obtain

- $\text{rev}(N(a(t))) = \text{rev}(a(N(t))) = a(\text{rev}(N(t))) >_{lpo} \text{rev}(N(t))$.
- $\text{rev}(N(\bar{a}(t))) = \text{rev}(\bar{a}(N(t))) = \bar{a}_n(\bar{a}_{n-1}(\dots\bar{a}_1(\bar{a}(\text{rev}(t'))\cdot))) >_{lpo}$
 $\bar{a}_n(\bar{a}_{n-1}(\dots\bar{a}_1(\text{rev}(t'))\cdot)) = \text{rev}(N(t))$.

- In $N(f(t, s, s', s''))$ write $s = \bar{c}_1(\bar{c}_2(\dots\bar{c}_k(\tilde{s})\cdot))$ where the root symbol of \tilde{s} is not of the shape \bar{a} for $a \in \Sigma$. We obtain

$$\begin{aligned} \text{rev}(N(f(t, s, s', s''))) &= \text{rev}(f(c_k(\dots(c_1(N(t))))\cdot, N(\tilde{s}), \dots, \dots)) \\ &= f(c_k(\dots(c_1(\text{rev}(N(t))))\cdot, \text{rev}(N(\tilde{s})), \dots, \dots) \\ &>_{lpo} \text{rev}(N(t)) \end{aligned}$$

Hence $N(f(t, s, s', s'')) \sqsupset N(t)$. The proof of $N(f(s, s', t, s'')) \sqsupset N(t)$ is similar.

- Since $f \gg \bar{a}$ for all $a \in \Sigma$, we obtain

$$\begin{aligned} \text{rev}(N(f(s, t, s', s''))) &= \text{rev}(N(f(s, N(t), s', s''))) \\ &= \text{rev}(N(f(s, \bar{a}_1(\bar{a}_2(\dots\bar{a}_n(t')\cdot)), s', s''))) \\ &= \text{rev}(f(a_n(a_{n-1}(\dots a_1(N(s))))\cdot, t', \dots, \dots)) \\ &= f(a_n(a_{n-1}(\dots a_1(\text{rev}(N(s))))\cdot), \text{rev}(t'), \dots, \dots) \\ &>_{lpo} \bar{a}_n(\bar{a}_{n-1}(\dots\bar{a}_1(\text{rev}(t'))\cdot)) \\ &= \text{rev}(\bar{a}_1(\bar{a}_2(\dots\bar{a}_n(t')\cdot))) \\ &= \text{rev}(N(t)). \end{aligned}$$

Hence $N(f(s, t, s', s'')) \sqsupset N(t)$. The proof of $N(f(s, s', s'', t)) \sqsupset N(t)$ is similar.

□

We conclude this section by showing that for the validity of theorem 7 it is essential to distinguish between barred and unbarred symbols. Let S' consist of the rules

$$\begin{aligned} f(x, a(y), z, w) &\rightarrow f(a(x), y, z, w) \\ f(x, y, z, a(w)) &\rightarrow f(x, y, a(z), w) \end{aligned}$$

for all $a \in \Sigma$. Assume that $N(t)$ was defined to be the normal form of t with respect to S' instead of S . Then we show that theorem 7 does not hold any more. Assume that \sqsupset satisfies all the assertions of theorem 7. If \sqsupset satisfies $1(0(c)) = N(1(0(c))) \sqsupset N(0(1(c))) = 0(1(c))$, then we get the contradiction

$$\begin{aligned} f(0(1(c)), c, c, c) &= N(f(c, 1(0(c)), c, c) \sqsupset N(f(c, 0(1(c)), c, c) = \\ &f(1(0(c)), c, c, c) = N(f(1(0(c)), c, c, c)) \sqsupset \\ &N(f(0(1(c)), c, c, c)) = f(0(1(c)), c, c, c). \end{aligned}$$

On the other hand if \sqsupset satisfies $0(1(c)) = N(0(1(c))) \sqsupset N(1(0(c))) = 1(0(c))$, then we get the contradiction

$$\begin{aligned} f(1(0(c)), c, c, c) &= N(f(c, 0(1(c)), c, c) \sqsupset N(f(c, 1(0(c)), c, c) = \\ &f(0(1(c)), c, c, c) = N(f(0(1(c)), c, c, c)) \sqsupset \\ &N(f(1(0(c)), c, c, c)) = f(1(0(c)), c, c, c). \end{aligned}$$

We conclude that the distinct terms $1(0(c))$ and $0(1(c))$ can not be compared by the order \sqsupset , contradicting totality as required in theorem 7.

5 The final order

To construct the final order we need a total order \succ extending the rewrite relation \rightarrow_{R_P} . Since we assume that R_P is terminating, we know that $\rightarrow_{R_P}^+$ is a well-founded order. The existence of \succ is now immediate from the following well-known lemma, which is equivalent to the axiom of choice.

Lemma 12 *Every order on a fixed set extends to a total order.*

Proof: Apply Zorn's lemma to the set of all orders on the fixed set extending the given order, ordered with the usual set inclusion. □

Using a more delicate construction one can prove that every well-founded order extends to a total well-founded order. However, this total well-founded order is usually not monotonic. Even more, if R is any terminating TRS which

is not totally terminating, we know that any total well-founded order extending \rightarrow_R^+ is not monotonic. For our purposes we do not need to worry about well-foundedness at this moment since the final order will satisfy the subterm property by which well-foundedness will follow from proposition 2. From now on we assume that \succ is any total order on ground terms for which

$$\text{if } t \rightarrow_{R_P} u \text{ then } t \succ u.$$

To achieve monotonicity we only want to apply the order \succ on ground terms having f as its root symbol, and not containing other f symbols. We introduce another order \triangleright extending \succ to ground terms containing several f symbols. We still need some definitions. We do not need to distinguish between barred and unbarred symbols any more, we write $\bar{\cdot}$ for the set of all a and \bar{a} for $a \in \bar{\cdot}$. As before we write $\alpha(t) = a_1(a_2(\dots(a_n(t))))$ for $\alpha = a_1 a_2 \dots a_n \in \bar{\cdot}^*$ and any term t . Any ground term t can uniquely be written as $t = \alpha(t')$ with either $t' = c$ or the root symbol of t' is f . We define $\text{cap}(t) = \alpha$. Further we define

$$\text{trunc}(\alpha(f(t_1, t_2, t_3, t_4))) = f(\text{cap}(t_1), \text{cap}(t_2), \text{cap}(t_3), \text{cap}(t_4))$$

for all $\alpha \in \bar{\cdot}^*$ and all terms t_1, t_2, t_3, t_4 . Now we recursively define

$$t = \alpha(f(t_1, t_2, t_3, t_4)) \triangleright \beta(f(u_1, u_2, u_3, u_4)) = u$$

if and only if $\alpha = \beta$ and either

- $\text{trunc}(t) \succ \text{trunc}(u)$, or
- $\text{trunc}(t) = \text{trunc}(u)$ and $(t_1, t_2, t_3, t_4) \triangleright_{lex} (u_1, u_2, u_3, u_4)$.

Terms not of the shape $\alpha(f(t_1, t_2, t_3, t_4))$ are not related by the relation \triangleright .

Lemma 13 *The relation \triangleright is irreflexive, transitive and monotonic.*

Proof: By induction on the depth of the terms. For monotonicity we need the observation that $t \triangleright u$ implies $\text{trunc}(f(t, s, s', s'')) = \text{trunc}(f(u, s, s', s''))$, and similar for the second, third and fourth arguments of f . \square

Lemma 14 *Let $l \rightarrow r$ be a rule of R_P and let σ be a ground substitution. Then $l^\sigma \triangleright r^\sigma$.*

Proof: Let σ' be the ground substitution defined by $x^{\sigma'} = \text{cap}(x^\sigma)$ for all variables x . Let $t = f(t_1, t_2, t_3, t_4)$ be any term for which f does not occur in t_1, t_2, t_3, t_4 . Then $\text{cap}(t_i^\sigma) = t_i^{\sigma'}$ for $i = 1, 2, 3, 4$, hence

$$\begin{aligned} \text{trunc}(t^\sigma) &= f(\text{cap}(t_1^\sigma), \text{cap}(t_2^\sigma), \text{cap}(t_3^\sigma), \text{cap}(t_4^\sigma)) \\ &= f(t_1^{\sigma'}, t_2^{\sigma'}, t_3^{\sigma'}, t_4^{\sigma'}) \\ &= t^{\sigma'}. \end{aligned}$$

Since both l and r satisfy the conditions we assumed for t , and we have $t \succ u$ if $t \rightarrow_{R_P} u$, we obtain

$$\mathbf{trunc}(l^\sigma) = l^{\sigma'} \succ r^{\sigma'} = \mathbf{trunc}(r^\sigma),$$

proving $l^\sigma \triangleright r^\sigma$. \square

The following lemma follows directly from the definition of N .

Lemma 15 *Let $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in ,'^*$ and let*

$$N(f(\alpha_1(c), \alpha_2(c), \alpha_3(c), \alpha_4(c))) = f(\beta_1(c), \beta_2(c), \beta_3(c), \beta_4(c)).$$

Let t_1, t_2, t_3, t_4 be ground terms that are either equal to c or have f as its root symbol. Then

$$\begin{aligned} N(f(\alpha_1(t_1), \alpha_2(t_2), \alpha_3(t_3), \alpha_4(t_4))) &= \\ f(\beta_1(N(t_1)), \beta_2(N(t_2)), \beta_3(N(t_3)), \beta_4(N(t_4))). \end{aligned}$$

Lemma 16 *Let t, u be two ground terms satisfying $t \neq u$ and $N(t) = N(u)$. Then either $t \triangleright u$ or $u \triangleright t$.*

Proof: We proceed by induction on the depth of the terms.

From $t \neq u$ and $N(t) = N(u)$ follows that both t and u contain at least one symbol f . Hence we can write $t = \alpha(f(t_1, t_2, t_3, t_4))$ and $u = \beta(f(u_1, u_2, u_3, u_4))$ for $\alpha, \beta \in ,'^*$. From $N(t) = N(u)$ follows that $\alpha = \beta$. If $\mathbf{trunc}(t) \neq \mathbf{trunc}(u)$ then the lemma follows from totality of \succ .

In the remaining case we have $\mathbf{trunc}(t) = \mathbf{trunc}(u)$. So we can write $t_i = \alpha_i(t'_i)$ and $u_i = \alpha_i(u'_i)$ for t'_i, u'_i be ground terms that are either equal to c or have f as its root symbol, for $i = 1, 2, 3, 4$. From lemma 15 we obtain $\beta_1, \beta_2, \beta_3, \beta_4 \in ,'^*$ such that

$$\begin{aligned} \alpha(f(\beta_1(N(t'_1)), \beta_2(N(t'_2)), \beta_3(N(t'_3)), \beta_4(N(t'_4)))) & \\ = \alpha(N(f(\alpha_1(t'_1), \alpha_2(t'_2), \alpha_3(t'_3), \alpha_4(t'_4)))) & \\ = N(\alpha(f(\alpha_1(t'_1), \alpha_2(t'_2), \alpha_3(t'_3), \alpha_4(t'_4)))) & \\ = N(t) & \\ = N(u) & \\ = N(\alpha(f(\alpha_1(u'_1), \alpha_2(u'_2), \alpha_3(u'_3), \alpha_4(u'_4)))) & \\ = \alpha(N(f(\alpha_1(u'_1), \alpha_2(u'_2), \alpha_3(u'_3), \alpha_4(u'_4)))) & \\ = \alpha(f(\beta_1(N(u'_1)), \beta_2(N(u'_2)), \beta_3(N(u'_3)), \beta_4(N(u'_4)))) & \end{aligned}$$

Hence $N(t'_i) = N(u'_i)$ and consequently $N(t_i) = \alpha_i(N(t'_i)) = \alpha_i(N(u'_i)) = N(u_i)$ for $i = 1, 2, 3, 4$. From $t \neq u$ and $\alpha = \beta$ we conclude that $t_i \neq u_i$ for some $i = 1, 2, 3, 4$. Applying the induction hypothesis yields that either $(t_1, t_2, t_3, t_4) \triangleright_{lex} (u_1, u_2, u_3, u_4)$.

(u_1, u_2, u_3, u_4) or $(u_1, u_2, u_3, u_4) \triangleright_{lex} (t_1, t_2, t_3, t_4)$. Consequently either $t \triangleright u$ or $u \triangleright t$, which we had to prove. \square

Now we can define the final order $>$ on ground terms:

$$t > u \iff N(t) \sqsupset N(u) \vee (N(t) = N(u) \wedge t \triangleright u).$$

Clearly $>$ is indeed irreflexive and transitive. Moreover, it satisfies the following properties.

- $>$ is total. This follows from totality of \sqsupset (theorem 7) and lemma 16.
- $>$ is monotonic. This follows from theorem 7 and lemma 13.
- $>$ satisfies the subterm property. This follows from proposition 11.
- $>$ is well-founded. This follows from the above results and proposition 2.
- $l^\sigma > r^\sigma$ for any rule $l \rightarrow r$ of R_P and any ground substitution σ . This follows from proposition 6 and lemma 14.

We conclude that R_P is compatible with the monotonic well-founded total order $>$, hence R_P is totally terminating. This concludes the proof of theorem 3.

6 Conclusions

We proved that total termination is an undecidable property of finite term rewriting systems. We conjecture that total termination is even undecidable for a single rewrite rule.

Acknowledgements: We should like to thank Alfons Geser and Maria Ferreira for their comments on an earlier version of this paper.

References

- [1] CARON, A. C. Linear bounded automata and rewrite systems: influence of initial configurations on decision properties. In *Proceedings of the Colloquium on Trees in Algebra and Programming* (1991), vol. 493 of *Lecture Notes in Computer Science*, Springer, pp. 74–89.
- [2] DAUCHET, M. Simulation of Turing machines by a regular rewrite rule. *Theoretical Computer Science* 103, 2 (1992), 409–420. Appeared before in *Proceedings of RTA89, Lecture Notes in Computer Science* 355, Springer, 1989.

- [3] FERREIRA, M. C. F., AND ZANTEMA, H. Well-foundedness of term orderings. Fourth International Workshop on Conditional Term Rewriting Systems, Jerusalem, July 1994, to appear in *Lecture Notes in Computer Science*, Springer.
- [4] FERREIRA, M. C. F., AND ZANTEMA, H. Total termination of term rewriting. In *Proceedings of the 5th Conference on Rewriting Techniques and Applications* (1993), C. Kirchner, Ed., vol. 690 of *Lecture Notes in Computer Science*, Springer, pp. 213–227.
- [5] FERREIRA, M. C. F., AND ZANTEMA, H. Syntactical analysis of total termination. In *Proceedings of the 4th International Conference on Algebraic and Logic Programming* (1994), G. Levi and M. Rodríguez-Artalejo, Eds., vol. 850 of *Lecture Notes in Computer Science*, Springer, pp. 204–222.
- [6] FERREIRA, M. C. F., AND ZANTEMA, H. Total termination of term rewriting. *Journal of Applicable Algebra in Engineering, Communication and Computing* (1995). To appear.
- [7] GALLIER, J. What’s so special about Kruskal’s theorem and the ordinal ω_1 ? A survey of some results in proof theory. *Annals of Pure and Applied Logic* 53 (1991), 199–260.
- [8] HUET, G., AND LANKFORD, D. S. On the uniform halting problem for term rewriting systems. Rapport Laboria 283, INRIA, 1978.
- [9] KAMIN, S., AND LÉVY, J. J. Two generalizations of the recursive path ordering. University of Illinois, 1980.
- [10] KRUSKAL, J. Well-quasi-ordering, the tree theorem, and Vazsonyi’s conjecture. *Trans. American Mathematical Society* 95 (1960), 210–225.
- [11] LESCANNE, P. On termination of one rule rewrite systems. *Theoretical Computer Science* 132 (1994), 395–401.
- [12] MIDDELDORP, A., AND GRAMLICH, B. Simple termination is difficult. In *Proceedings of the 5th Conference on Rewriting Techniques and Applications* (1993), C. Kirchner, Ed., *Lecture Notes in Computer Science*, Springer.
- [13] MIDDELDORP, A., AND ZANTEMA, H. Simple termination revisited. In *Proceedings of the 12th International Conference on Automated Deduction (CADE12)* (1994), A. Bundy, Ed., vol. 814 of *Lecture Notes in Computer Science*, Springer, pp. 451–465.
- [14] POST, E. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society* 52 (1946).

- [15] ROZENBERG, G., AND SALOMAA, A. *Cornerstones of Undecidability*. Prentice Hall, 1994.
- [16] RUBIO, A. *Automated Deduction with Constrained Clauses*. PhD thesis, Polytechnical University Catalony, Barcelona, 1994.
- [17] STEINBACH, J. Extensions and comparison of simplification orderings. In *Proceedings of the 3rd Conference on Rewriting Techniques and Applications* (1989), N. Dershowitz, Ed., vol. 355 of *Lecture Notes in Computer Science*, Springer, pp. 434–448.
- [18] ZANTEMA, H. Termination of term rewriting by interpretation. In *Conditional Term Rewriting Systems, proceedings third international workshop CTRS-92* (1993), M. Rusinowitch and J. Rémy, Eds., vol. 656 of *Lecture Notes in Computer Science*, Springer, pp. 155–167.
- [19] ZANTEMA, H. Type removal in term rewriting. In *Conditional Term Rewriting Systems, proceedings third international workshop CTRS-92* (1993), M. Rusinowitch and J. Rémy, Eds., vol. 656 of *Lecture Notes in Computer Science*, Springer, pp. 148–154.
- [20] ZANTEMA, H. Termination of term rewriting: interpretation and type elimination. *Journal of Symbolic Computation* 17 (1994), 23–50.