Termination of term rewriting by semantic labelling

Termination of term rewriting by semantic labelling

H. Zantema

RUU-CS-93-24
July 1993

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

Termination of term rewriting by semantic labelling

H. Zantema

Technical Report RUU-CS-93-24
July 1993

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924-3275

Termination of term rewriting by semantic labelling *

H. Zantema
Utrecht University, Department of Computer Science
P.O. box 80.089, 3508 TB Utrecht, The Netherlands
phone +31-30-534116, e-mail: hansz@cs.ruu.nl

Abstract

A new kind of transformation of TRS's is proposed, depending on a choice for a model for the TRS. The labelled TRS is obtained from the original one by labelling operation symbols, possibly creating extra copies of some rules. This construction has the remarkable property that the labelled TRS is terminating if and only if the original TRS is terminating. Although the labelled version has more operation symbols and may have more rules (sometimes infinitely many), termination is often easier to prove for the labelled TRS than for the original one. This provides a new technique for proving termination, making classical techniques like RPO and polynomial interpretations applicable for non-simplifying TRS's.

1 Introduction

The well-known quicksort algorithm can be described as a term rewriting system (TRS) as follows:

$$\begin{array}{lcl}
\mathsf{qsort}(\mathsf{nil}) & \to & \mathsf{nil} \\
\mathsf{qsort}(x:y) & \to & \mathsf{qsort}(\mathsf{low}(x,y)) \circ (x:\mathsf{qsort}(\mathsf{high}(x,y))) \\
\mathsf{low}(x,\mathsf{nil}) & \to & \mathsf{nil} \\
\mathsf{low}(x,y:z) & \to & \mathsf{if}(y \leq x, y:\mathsf{low}(x,z), \mathsf{low}(x,z)) \\
\mathsf{high}(x,\mathsf{nil}) & \to & \mathsf{nil} \\
\mathsf{high}(x,y:z) & \to & \mathsf{if}(y \leq x, \mathsf{high}(x,z), y:\mathsf{high}(x,z)) \\
0 \leq x & \to & \mathsf{true} \\
s(x) \leq 0 & \to & \mathsf{false} \\
s(x) \leq s(y) & \to & x \leq y \\
\mathsf{if}(\mathsf{true},x,y) & \to & x \\
\mathsf{if}(\mathsf{false},x,y) & \to & y.
\end{array}$$

Here $x:y$ can be interpreted as the list obtained by putting the element x in front of the list y, '$\circ$' can be interpreted as list concatenation, $\mathsf{low}(x,y)$ removes the elements from y that are greater than x, and $\mathsf{high}(x,y)$ removes the elements from y that are less or equal

*This is an extended and revised version of technical report RUU-CS-92-38, appeared in December 1992.

than x. This TRS corresponds to a functional program implementing quicksort on natural numbers. Termination of this program is not difficult to see: for each recursive call of low and high the length of the right argument strictly decreases. Further the lengths of $\mathsf{low}(x, y)$ and $\mathsf{high}(x, y)$ are less or equal than the length of y, and hence for each recursive call of qsort the length of the argument strictly decreases.

However, if we forget about the semantics of the terms being lists and numbers, then proving termination of the TRS is not that easy any more. Standard techniques like recursive path order (RPO) fail. We should like to have a technique for proving termination of a TRS making use of the semantics of the TRS. One technique doing so is semantic path order ([7, 3]). It can be seen as a generalization of RPO and is discussed in section 7.

In this paper we describe another technique: given a TRS having some semantics, we introduce a labelling of the operation symbols in the TRS depending on the semantics of their arguments. We do this in such a way that termination of the original TRS is equivalent to termination of the labelled TRS. The labelled TRS has more operation symbols than the original TRS, and often more rules, sometimes even infinitely many. The original TRS can be obtained from the labelled TRS by removing all labels and removing multiple copies of rules. Although the labelled TRS is greater in some sense than the original one, in many cases termination of the labelled version is easier to prove than termination of the original one. We propose proving termination of a TRS by proving termination of a particular labelled version as a new method. This method we call *semantic labelling.*

For instance, in the quicksort system we can label every symbol 'qsort' by the length of the list interpretation of its argument. We obtain infinitely many distinct operation symbols 'qsort_i' instead of one symbol 'qsort'; the other operation symbols do not change. The labelled TRS is obtained from the original one by replacing the first two rules by the rule

$$\mathsf{qsort}_0(\mathsf{nil}) \;\to\; \mathsf{nil}$$

and infinitely many rules

$$\mathsf{qsort}_i(x : y) \;\to\; \mathsf{qsort}_j(\mathsf{low}(x, y)) \circ (x : \mathsf{qsort}_k(\mathsf{high}(x, y)))$$

for natural numbers i, j, k satisfying $j < i$ and $k < i$. Since the labels occurring in the left hand sides are all strictly greater than the labels occurring in the corresponding right hand sides, it is easy to prove termination of the labelled system by a recursive path order on a precedence satisfying $\mathsf{qsort}_{i+1} > \mathsf{qsort}_i$ for all i.

Our method is helpful for TRS's that are not simply terminating. The very simplest example is the system

$$f(f(x)) \;\to\; f(g(f(x))).$$

We can choose a model of two elements and obtain the labelled system

$$\begin{aligned} f_2(f_1(x)) &\;\to\; f_1(g(f_1(x))) \\ f_2(f_2(x)) &\;\to\; f_1(g(f_2(x))) \end{aligned}$$

of which simple termination is very easily proved. Less artificial is the factorial example:

$$\begin{array}{lll} \mathsf{fact}(s(x)) & \to & \mathsf{fact}(p(s(x))) * s(x) \\ p(s(0)) & \to & 0 \\ p(s(s(x))) & \to & s(p(s(x))). \end{array}$$

This system is not simply terminating. However, by semantic labelling it transforms to another system that is easily proved to be simply terminating by standard techniques as we shall see in section 3. A nice source of examples is [15].

Semantic labelling is also helpful for proving termination of TRS's that don't have obvious semantics, but for which particular patterns can be recognized in the rewrite rules. The system $f(f(x)) \to f(g(f(x)))$ can be considered of this type; we shall give more interesting examples. This approach is closely related to typing the operation symbols and proving termination of the resulting order-sorted system as discussed in [6]. Other approaches of proving termination of non-simply terminating systems in a syntactic way can be found in [13, 12, 1, 10].

The technique of semantic labelling does not restrict to plain TRS's. In section 4 we show that the same construction and the preservation of termination behaviour also holds for term rewriting modulo equations. Further semantic labelling serves well for completion of an equational specification: if the original equations hold in the model we want to use, the same holds for all critical pairs emerging during the completion process, and all these critical pairs can be labelled and oriented using a termination order we have for labelled terms.

In section 5 we present an extension of the theory in which the requirement of having a model is weakened. In a model the left hand side of any rule has to be equal to the corresponding right hand side; in this extension the left hand side is allowed to be greater than the corresponding right hand side.

Semantic labelling does not only provide termination proofs; it can also be used for proving bounds on reduction lengths. By labelling the length of a reduction does not change. So if we have a bound on the reduction lengths in the labelled version, such a bound can be used to prove a bound for the unlabelled version. Semantic labelling also holds for other properties like confluence, in the sense that confluence of a TRS follows from confluence of its labelled version. However, we do not know examples of confluence proofs that are simplified by this observation.

In sections 6 and 7 we compare semantic labelling with existing techniques and characterizations of TRS termination. In section 8 we sketch how labelling leads to a generalization of Kruskal's theorem, and can be a starting point for purely syntactical RPO-like orderings having the power to prove termination of systems that are not simply terminating.

2 The basic theorem

Let $\mathcal{F}$ be a set of operation symbols, each having a fixed arity ≥ 0. We define an $\mathcal{F}$*-algebra* $\mathcal{M}$ to consist of a set M (the carrier set) and for every $f \in \mathcal{F}$ of arity n a function $f_{\mathcal{M}} : M^n \to M$. In the following we fix an $\mathcal{F}$-algebra $\mathcal{M}$.

Let $\mathcal{X}$ be a set of variable symbols. Let $M^{\mathcal{X}} = \{\sigma : \mathcal{X} \to M\}$. We define $\phi_{\mathcal{M}} : \mathcal{T}(\mathcal{F}, \mathcal{X}) \times M^{\mathcal{X}} \to M$ inductively by

$$\begin{aligned}\phi_{\mathcal{M}}(x, \sigma) &= \sigma(x), \\ \phi_{\mathcal{M}}(f(t_1, \ldots, t_n), \sigma) &= f_{\mathcal{M}}(\phi_{\mathcal{M}}(t_1, \sigma), \ldots, \phi_{\mathcal{M}}(t_n, \sigma))\end{aligned}$$

for $x \in \mathcal{X}, \sigma : \mathcal{X} \to M, f \in \mathcal{F}, t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. This means that $\phi_{\mathcal{M}}(-, \sigma)$ is the homomorphic extension of σ to general terms. If it is clear which model is involved, we write simply ϕ instead of $\phi_{\mathcal{M}}$. The function ϕ satisfies the following useful property.

Lemma 1 *Let $\sigma : \mathcal{X} \to M$ and let $\tau : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$. Define $\sigma' : \mathcal{X} \to M$ by $\sigma'(x) = \phi(\tau(x), \sigma)$. Then*

$$\phi(t^\tau, \sigma) = \phi(t, \sigma').$$

Proof: By induction on the structure of t. $\square$

Next we introduce labelling of operation symbols: choose for every $f \in \mathcal{F}$ a corresponding non-empty set S_f of labels. Now the new signature $\overline{\mathcal{F}}$ is defined by

$$\overline{\mathcal{F}} = \{f_s | f \in \mathcal{F}, s \in S_f\},$$

where the arity of f_s is defined to be the arity of f. An operation symbol f is called *labelled* if S_f contains more than one element. For unlabelled f the set S_f containing only one element can be left implicit; in that case we shall often write f instead of f_s.

Choose for every $f \in \mathcal{F}$ a map $\pi_f : M^n \to S_f$, where n is the arity of f. This map describes how a function symbol is labelled depending on the values of its arguments as interpreted in $\mathcal{M}$. For unlabelled f this function π_f can be left implicit. We extend the labelling of operation symbols to a labelling of terms by defining $\mathsf{lab} : \mathcal{T}(\mathcal{F}, \mathcal{X}) \times M^{\mathcal{X}} \to \mathcal{T}(\overline{\mathcal{F}}, \mathcal{X})$ inductively by

$$\begin{aligned}\mathsf{lab}(x, \sigma) &= x, \\ \mathsf{lab}(f(t_1, \ldots, t_n), \sigma) &= f_{\pi_f(\phi(t_1,\sigma),\ldots,\phi(t_n,\sigma))}(\mathsf{lab}(t_1, \sigma), \ldots, \mathsf{lab}(t_n, \sigma))\end{aligned}$$

for $x \in \mathcal{X}, \sigma : \mathcal{X} \to M, f \in \mathcal{F}, t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. This labelling of terms satisfies the following property.

Lemma 2 *Let $\sigma : \mathcal{X} \to M$ and let $\tau : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$. Define $\sigma' : \mathcal{X} \to M$ by $\sigma'(x) = \phi(\tau(x), \sigma)$, and define $\overline{\tau} : \mathcal{X} \to \mathcal{T}(\overline{\mathcal{F}}, \mathcal{X})$ by $\overline{\tau}(x) = \mathsf{lab}(\tau(x), \sigma)$. Then*

$$\mathsf{lab}(t^\tau, \sigma) = \mathsf{lab}(t, \sigma')^{\overline{\tau}}.$$

Proof: By induction on the structure of t. If t is a variable the lemma follows from the definition of $\overline{\tau}$. If $t = f(t_1, \ldots, t_n)$ we obtain

$$\mathsf{lab}(t^\tau, \sigma) = \mathsf{lab}(f(t_1^\tau, \ldots, t_n^\tau), \sigma) = f_{\pi_f(\phi(t_1^\tau,\sigma),\ldots,\phi(t_n^\tau,\sigma))}(\mathsf{lab}(t_1^\tau, \sigma), \ldots, \mathsf{lab}(t_n^\tau, \sigma))$$

and

$$\mathsf{lab}(t, \sigma')^{\overline{\tau}} = \mathsf{lab}(f(t_1, \ldots, t_n), \sigma')^{\overline{\tau}} = f_{\pi_f(\phi(t_1,\sigma'),\ldots,\phi(t_n,\sigma'))}(\mathsf{lab}(t_1, \sigma')^{\overline{\tau}}, \ldots, \mathsf{lab}(t_n, \sigma')^{\overline{\tau}}).$$

The labels of f are equal due to lemma 1 and the arguments are equal due to the induction hypothesis. Hence both terms are equal. $\square$

Let R be a TRS over $\mathcal{F}$. We say that an $\mathcal{F}$-algebra $\mathcal{M}$ is a *model* for R if

$$\phi_{\mathcal{M}}(l,\sigma) = \phi_{\mathcal{M}}(r,\sigma)$$

for all $\sigma : \mathcal{X} \to M$ and all rules $l \to r$ of R.

Fix an $\mathcal{F}$-algebra $\mathcal{M}$ together with corresponding sets S_f and functions π_f. For any TRS R over $\mathcal{F}$ we define $\overline{R}$ to be the TRS over $\overline{\mathcal{F}}$ consisting of the rules

$$\mathsf{lab}(l,\sigma) \to \mathsf{lab}(r,\sigma)$$

for all $\sigma : \mathcal{X} \to M$ and all rules $l \to r$ of R. Note that if R and all S_f are finite, then $\overline{R}$ is finite too. The following lemma states how reduction over R can be transformed to reduction over $\overline{R}$.

Lemma 3 *Let $\mathcal{M}$ be a model for R. Let $t, t' \in \mathcal{T}(\mathcal{F},\mathcal{X})$ satisfy $t \to_R t'$. Then*

$$\mathsf{lab}(t,\sigma) \ \to_{\overline{R}} \ \mathsf{lab}(t',\sigma)$$

for all $\sigma : \mathcal{X} \to M$.

Proof: If $t = l^\tau$ and $t' = r^\tau$ for some rule $l \to r$ of R and some $\tau : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$ we obtain from lemma 2

$$\mathsf{lab}(t,\sigma) = \mathsf{lab}(l,\sigma')^{\overline{\tau}} \ \to_{\overline{R}} \ \mathsf{lab}(r,\sigma')^{\overline{\tau}} = \mathsf{lab}(t',\sigma),$$

since $\mathsf{lab}(l,\sigma') \ \to \ \mathsf{lab}(r,\sigma')$ is a rule of $\overline{R}$.

Let $t \to_R t'$ and $\mathsf{lab}(t,\sigma) \ \to_{\overline{R}} \ \mathsf{lab}(t',\sigma)$. We still have to prove that

$$\mathsf{lab}(f(\ldots,t,\ldots),\sigma) \ \to_{\overline{R}} \ \mathsf{lab}(f(\ldots,t',\ldots),\sigma).$$

Since $\mathcal{M}$ is a model for R we know that $\phi(t,\sigma) = \phi(t',\sigma)$. We obtain

$$\begin{array}{rcl}
\mathsf{lab}(f(\ldots,t,\ldots),\sigma) & = & f_{\pi_f(\ldots,\phi(t,\sigma),\ldots)}(\ldots,\mathsf{lab}(t,\sigma),\ldots) \\
& = & f_{\pi_f(\ldots,\phi(t',\sigma),\ldots)}(\ldots,\mathsf{lab}(t,\sigma),\ldots) \\
& \to_{\overline{R}} & f_{\pi_f(\ldots,\phi(t',\sigma),\ldots)}(\ldots,\mathsf{lab}(t',\sigma),\ldots) \\
& = & \mathsf{lab}(f(\ldots,t',\ldots),\sigma).
\end{array}$$

$\square$

As usual, a TRS R is defined to be *terminating* if it does not admit infinite reductions

$$t_1 \ \to_R \ t_2 \ \to_R \ t_3 \ \to_R \ \cdots.$$

In the literature a terminating TRS is also called *strongly normalizing* or *noetherian.* Now we arrive at the main theorem of this paper.

Theorem 4 *Let $\mathcal{M}$ be a model for a TRS R over $\mathcal{F}$. Choose for every $f \in \mathcal{F}$ a non-empty set S_f of labels and a map $\pi_f : M^n \to S_f$, where n is the arity of f. Define $\overline{R}$ as above. Then R is terminating if and only if $\overline{R}$ is terminating.*

Proof: Assume $\overline{R}$ allows an infinite reduction. Then removing all labels yields an infinite reduction in R.

On the other hand assume R allows an infinite reduction

$$t_1 \to_R t_2 \to_R t_3 \to_R \cdots.$$

Choose $\sigma : \mathcal{X} \to M$ arbitrarily. Then according to lemma 3 $\overline{R}$ allows an infinite reduction

$$\mathsf{lab}(t_1, \sigma) \to_{\overline{R}} \mathsf{lab}(t_2, \sigma) \to_{\overline{R}} \mathsf{lab}(t_3, \sigma) \to_{\overline{R}} \cdots.$$

$\square$

In section 6 an alternative proof of this theorem is proposed. One can wonder whether similar theorems hold for other interesting properties like confluence, weak confluence and weak normalization. Due to lemma 3 and the trivial counterpart (removing labels in an $\overline{R}$-reduction yields an R-reduction) it is not difficult to prove that if $\overline{R}$ is confluent, weakly confluent or weakly normalizing, then R satisfies the same property. However, we do not know examples in which these observations are helpful for proving these properties; in the typical case the proof obligations for $\overline{R}$ are similar or more complicated than for R.

Before giving a list of examples of termination proofs using theorem 4 we briefly discuss the notion of *simple termination*. For a set $\mathcal{F}$ of operation symbols define $Emb(\mathcal{F})$ to be the TRS consisting of all the rules

$$f(x_1, \ldots, x_n) \to x_i$$

with $f \in \mathcal{F}$ and $i \in \{1, \ldots, n\}$. A TRS R over $\mathcal{F}$ is defined to be *simply terminating* if $R \cup Emb(\mathcal{F})$ is terminating. In the literature ([9, 11, 17]) some other equivalent definitions appear. If $\mathcal{F}$ is finite it is also equivalent to the notion of a *simplifying* TRS ([8]); if $\mathcal{F}$ is infinite there is a slight difference (see [11]). However, for the scope of this paper it suffices to see that some terminating TRS's are *not* simply terminating using our definition, and to know that standard techniques like RPO and KBO, both with status (see e.g. [14]), and polynomial interpretations, all fail for TRS's that are not simply terminating.

3 Examples

Example 1.

The simplest example R of a terminating TRS that is not simply terminating is

$$f(f(x)) \to f(g(f(x))).$$

Intuitively termination of this system is not difficult: at every step the number of operation symbols f of which the argument is again a term with head symbol f decreases. This idea can be transformed directly to a semantic labelling: define the model $\mathcal{M}$ with $M = \{1, 2\}$, and $f_{\mathcal{M}}(x) = 2$ and $g_{\mathcal{M}}(x) = 1$ for $x = 1, 2$. Choose $S_f = \{1, 2\}$ and π_f is the identity; choose g to be unlabelled. Then $\overline{R}$ is

$$\begin{array}{lcl} f_2(f_1(x)) & \to & f_1(g(f_1(x))) \\ f_2(f_2(x)) & \to & f_1(g(f_2(x))); \end{array}$$

the first rule is obtained by choosing $\sigma(x) = 1$, the second by choosing $\sigma(x) = 2$. Termination of $\overline{R}$ is easily proved by counting the number of f_2 symbols. Also recursive path order and polynomial interpretations ($[f_1](x) = [g](x) = x$, $[f_2](x) = x + 1$) suffice for proving termination. Using theorem 4 we conclude that the original system R is terminating too.

Example 2.
Consider the TRS

$$f(0, 1, x) \to f(x, x, x)$$

from [16]. This system is not simply terminating. For proving termination we want to use the observation that in the left hand side the first and the second argument of f are distinct while in the right hand side they are equal. This distinction is made by choosing $S_f = \{1, 2\}$ and $\pi_f(x, y, z) = 1$ if $x = y$ and $\pi_f(x, y, z) = 2$ if $x \neq y$. We still need any model in which 0 and 1 are indeed distinct; a simple one is $M = \{0, 1\}$ with $0_{\mathcal{M}} = 0$, $1_{\mathcal{M}} = 1$, and $f_{\mathcal{M}}(x, y, z) = 0$ for $x, y, z = 0, 1$. Now we obtain the labelled system

$$f_2(0, 1, x) \to f_1(x, x, x)$$

which is easily proved to be terminating by any standard technique.

Example 3.
A valid definition of the function max to compute the maximum of two natural numbers is the following: if $x \geq y$ then $\mathsf{max}(x, y) = x$, otherwise $\mathsf{max}(x, y) = \mathsf{max}(y, x)$. This definition can be transformed to the following TRS MAX:

$$\begin{array}{lcl} \mathsf{max}(x, y) & \to & c(x, y, x \geq y) \\ x \geq 0 & \to & \mathsf{true} \\ 0 \geq s(x) & \to & \mathsf{false} \\ s(x) \geq s(y) & \to & x \geq y \\ c(x, y, \mathsf{true}) & \to & x \\ c(x, y, \mathsf{false}) & \to & \mathsf{max}(y, x). \end{array}$$

This system is not simply terminating since by adding the rule $x \geq y \to x$ which is in $Emb(\mathcal{F})$ we obtain the infinite reduction

$$\mathsf{max}(\mathsf{false}, \mathsf{false}) \to c(\mathsf{false}, \mathsf{false}, \mathsf{false} \geq \mathsf{false})$$

$$\to c(\mathsf{false}, \mathsf{false}, \mathsf{false}) \to \mathsf{max}(\mathsf{false}, \mathsf{false}) \to \cdots.$$

However, MAX can be proved to be terminating by semantic labelling. As a model $\mathcal{M}$ we choose the natural numbers in which we identify true and false by 1 and 0, respectively. More precisely: $M = \mathbb{N}$, $\mathsf{max}_{\mathcal{M}}(x, y) = \max(x, y)$, $\mathsf{true}_{\mathcal{M}} = 1$, $\mathsf{false}_{\mathcal{M}} = 0$, $0_{\mathcal{M}} = 0$, $s_{\mathcal{M}}(x) = x + 1$,

$$c_{\mathcal{M}}(x, y, z) = \begin{cases} x & \text{if } z > 0 \\ \max(x, y) & \text{if } z = 0 \end{cases}, \quad x \geq_{\mathcal{M}} y = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if } x < y. \end{cases}$$

One easily checks that $\mathcal{M}$ is indeed a model for MAX. We still have to find an appropriate labelling; consider the reduction

$$c(s(0), 0, \mathsf{false}) \to \mathsf{max}(0, s(0)) \to^+ c(0, s(0), \mathsf{false}) \to \mathsf{max}(s(0), 0) \to^+ c(s(0), 0, \mathsf{true}).$$

We shall label max and c in such a way that the three occurrences of c and the two occurrences of max in this sequence get distinct labels. A possible choice is $S_{\mathsf{max}} = \{1, 2\}$ and $S_c = \{1, 2, 3\}$ and

$$\pi_{\mathsf{max}}(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 2 & \text{if } x < y \end{cases} \qquad \pi_c(x, y, z) = \begin{cases} 1 & \text{if } z > 0 \\ 2 & \text{if } z = 0 \wedge x < y \\ 3 & \text{if } z = 0 \wedge x \geq y. \end{cases}$$

Now $\overline{MAX}$ is

$$\begin{array}{lcl} \mathsf{max}_1(x, y) & \to & c_1(x, y, x \geq y) \\ \mathsf{max}_2(x, y) & \to & c_2(x, y, x \geq y) \\ x \geq 0 & \to & \mathsf{true} \\ 0 \geq s(x) & \to & \mathsf{false} \\ s(x) \geq s(y) & \to & x \geq y \\ c_1(x, y, \mathsf{true}) & \to & x \\ c_2(x, y, \mathsf{false}) & \to & \mathsf{max}_1(y, x) \\ c_3(x, y, \mathsf{false}) & \to & \mathsf{max}_1(y, x) \\ c_3(x, y, \mathsf{false}) & \to & \mathsf{max}_2(y, x) \end{array}$$

and can be proved to be terminating by RPO using the precedence

$$c_3 \; > \; \mathsf{max}_2 \; > \; c_2 \; > \; \mathsf{max}_1 \; > \; c_1 \; > \; \geq \; > \; \mathsf{true} \; > \; \mathsf{false}.$$

Example 4.
In the system

$$\begin{array}{lcl} (x * y) * z & \to & x * (y * z) \\ (x + y) * z & \to & (x * z) + (y * z) \\ x * (y + f(z)) & \to & g(x, z) * (y + a) \end{array}$$

from [3] we can force that the symbols '$*$' in the last rule get distinct labels by choosing the model $\{1, 2\}$ and defining $a_{\mathcal{M}} = 1, f_{\mathcal{M}}(x) = 2, \pi_*(x, y) = x +_{\mathcal{M}} y = y, x *_{\mathcal{M}} y = 1$ for all $x, y = 1, 2$. The labelled system is

$$\begin{array}{lcl} (x *_1 y) *_1 z & \to & x *_1 (y *_1 z) \\ (x *_1 y) *_2 z & \to & x *_1 (y *_2 z) \\ (x *_2 y) *_1 z & \to & x *_1 (y *_1 z) \\ (x *_2 y) *_2 z & \to & x *_1 (y *_2 z) \\ (x + y) *_1 z & \to & (x *_1 z) + (y *_1 z) \\ (x + y) *_2 z & \to & (x *_2 z) + (y *_2 z) \\ x *_2 (y + f(z)) & \to & g(x, z) *_1 (y + a) \end{array}$$

and is proved terminating using RPO: give $*_1$ a lexicographic status, choose $*_2$ to be greater than all the other symbols and choose $*_1 > +$.

Example 5.
In the 'fact' system from the introduction choose $M = \mathbb{N}, 0_{\mathcal{M}} = 0, s_{\mathcal{M}}(x) = x+1, p_{\mathcal{M}}(0) = 0$, and $p_{\mathcal{M}}(x) = x-1$ for $x > 0$. Further choose $x *_{\mathcal{M}} y = x * y$ and $\mathsf{fact}_{\mathcal{M}}(x) = x!$. Clearly $\mathcal{M}$ is a model for the system; by labelling **fact** with the naturals and choosing $\pi_{\mathsf{fact}}(x) = x$ we get the labelled version

$$\begin{array}{lcl} \mathsf{fact}_{i+1}(s(x)) & \to & \mathsf{fact}_i(p(s(x))) * s(x) \\ p(s(0)) & \to & 0 \\ p(s(s(x))) & \to & s(p(s(x))) \end{array}$$

in which the first line stands for infinitely many rules, one for every $i \in \mathbb{N}$. An interpretation in $\mathbb{N}$ proving termination is $[0] = 0, [s](x) = x + 1, [p](x) = 2x, x[*]y = x + y, [\mathsf{fact}_i](x) = 4^i * x$.

4 Rewriting modulo equations

In this section we show how theorem 4 extends to rewriting modulo equations.

Theorem 5 *Let $\mathcal{M}$ be a model for a TRS R over $\mathcal{F}$. Choose for every $f \in \mathcal{F}$ a non-empty set S_f of labels and a map $\pi_f : M^n \to S_f$, where n is the arity of f. Define $\overline{R}$ as in section 2. Let $\mathcal{F}_u = \{f \in \mathcal{F} | \#S_f = 1\}$. Let $\mathcal{E}$ be any set of equations over $\mathcal{F}_u$ that hold in $\mathcal{M}$. Then R is terminating modulo $\mathcal{E}$ if and only if $\overline{R}$ is terminating modulo $\mathcal{E}$.*

Proof: Assume $\overline{R}$ allows an infinite reduction modulo $\mathcal{E}$:

$$t_1 \to_{\overline{R}} t_2 \equiv_{\mathcal{E}} t_3 \to_{\overline{R}} t_4 \equiv_{\mathcal{E}} t_5 \to_{\overline{R}} t_6 \cdots.$$

Then removing all labels yields an infinite reduction in R modulo $\mathcal{E}$.

On the other hand assume R allows an infinite reduction modulo $\mathcal{E}$:

$$t_1 \to_R t_2 \equiv_{\mathcal{E}} t_3 \to_R t_4 \equiv_{\mathcal{E}} t_5 \to_R t_6 \cdots.$$

Choose $\sigma : \mathcal{X} \to M$ arbitrarily. Similar to the proof of lemma 3 one proves that

$$\mathsf{lab}(t, \sigma) \equiv_{\mathcal{E}} \mathsf{lab}(t', \sigma)$$

for any t, t' satisfying $t \equiv_{\mathcal{E}} t'$. From this observation and lemma 3 we conclude that $\overline{R}$ allows an infinite reduction modulo $\mathcal{E}$:

$$\mathsf{lab}(t_1, \sigma) \to_{\overline{R}} \mathsf{lab}(t_2, \sigma) \equiv_{\mathcal{E}} \mathsf{lab}(t_3, \sigma) \to_{\overline{R}} \mathsf{lab}(t_4, \sigma) \equiv_{\mathcal{E}} \mathsf{lab}(t_5, \sigma) \to_{\overline{R}} \cdots.$$

□

In section 7 we present an application of this theorem. Note that all operation symbols in $\mathcal{E}$ are required to be unlabelled. This restriction is essential: otherwise the theorem does not hold without introducing extra restrictions. For instance, for the system

$$(x + y) + z \to x + (y + z)$$

we can choose the model of positive integers in which $+$ is interpreted as addition, which is commutative. If we choose $\pi_+(x,y) = x$, then the infinite labelled system is easily proved to be terminating modulo commutativity by the polynomial interpretation $x[+_i]y = x+y+i$. However, the original system is not terminating modulo commutativity.

Theorem 5 can be extended to allow $\mathcal{E}$ to contain commutativity of labelled symbols if π_f is required to be symmetric for these symbols. For other equations on labelled symbols it is not clear how it can be extended.

5 Quasi-models

In this section we give an extension of theorem 4 in the sense that $\mathcal{M}$ is not required to be a model for R any more. As a motivation consider the following TRS introduced in [4] for showing that completeness is not a modular property:

$$\begin{array}{lcl} f(a,b,x) & \to & f(x,x,x) \\ f(x,y,z) & \to & c \\ a & \to & c \\ b & \to & c. \end{array}$$

Clearly this system is closely related to example 2 of section 3. However, it does not allow any non-trivial model since in all models any term has the same interpretation as c. So theorem 4 is not helpful for proving termination of this system; using the extension presented in this section it is easily proved.

Until now the model M and label sets S_f were sets. Here we require them to be well-founded posets. The maps f_M and π_f have to be weakly monotone in all coordinates. Until now M was required to be a model for the TRS, meaning that the interpretation of a left hand side of a rule is always equal to the interpretation of the corresponding right hand side. Here M is only required to be a quasi-model for the TRS, meaning that the interpretation of a left hand side of a rule is $\geq$ the interpretation of the corresponding right hand side. Before presenting the theorem we give some definitions and lemmas.

Let $\mathcal{M}$ be an $\mathcal{F}$-algebra provided with a well-founded partial order $\geq$ for which each algebra operation is weakly monotone in all coordinates, more precisely: for all operation symbols $f \in \mathcal{F}$ and all $a_1, \ldots, a_n, b_1, \ldots, b_n \in M$ satisfying $a_i \geq b_i$ for all i, we have

$$f_{\mathcal{M}}(a_1, \ldots, a_n) \geq f_{\mathcal{M}}(b_1, \ldots, b_n).$$

For all $f \in \mathcal{F}$ let S_f be any set, again provided with a well-founded partial order $\geq$. For all $f \in \mathcal{F}$ of arity n let $\pi_f : M^n \to S_f$ any map that is weakly monotone in all coordinates. Define ϕ, lab and $\overline{\mathcal{F}}$ as in section 2. Let R be a TRS over $\mathcal{F}$. We say that the $\mathcal{F}$-algebra $\mathcal{M}$ is a *quasi-model* for R if

$$\phi_{\mathcal{M}}(l,\sigma) \geq \phi_{\mathcal{M}}(r,\sigma)$$

for all $\sigma : \mathcal{X} \to M$ and all rules $l \to r$ of R. As in section 2 we define $\overline{R}$ to be the TRS over $\overline{\mathcal{F}}$ consisting of the rules

$$\mathsf{lab}(l,\sigma) \to \mathsf{lab}(r,\sigma)$$

for all $\sigma : \mathcal{X} \to M$ and all rules $l \to r$ of R. Further the TRS Decr over $\overline{\mathcal{F}}$ is defined to consist of the rules

$$f_s(x_1, \ldots, x_n) \to f_{s'}(x_1, \ldots, x_n)$$

for all $f \in \mathcal{F}$ and all $s, s' \in S_f$ satisfying $s > s'$. Here $>$ denotes the strict part of $\geq$.

Lemma 6 *Let $\mathcal{M}$ be a quasi-model for R. Let $t, t' \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ satisfy $t \to_R t'$. Then $\phi(t, \sigma) \geq \phi(t', \sigma)$ for all $\sigma : \mathcal{X} \to M$.*

Proof: If $t = l^\tau$ and $t' = r^\tau$ for some rule $l \to r$ of R and some $\tau : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ the assertion follows from lemma 1 and the definition of quasi-model.

Let $t \to_R t'$ and $\phi(t, \sigma) \geq \phi(t', \sigma)$; we still have to prove that

$$\phi(f(\ldots, t, \ldots), \sigma) \geq \phi(f(\ldots, t', \ldots), \sigma)$$

for all $f \in \mathcal{F}$ and all $\sigma : \mathcal{X} \to M$. This follows from the definition of ϕ and the fact that $f_{\mathcal{M}}$ is weakly monotone in all coordinates. □

Lemma 7 *Let $\mathcal{M}$ be a quasi-model for R. Let $t, t' \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ satisfy $t \to_R t'$. Then for all $\sigma : \mathcal{X} \to M$ there is a term u over $\overline{\mathcal{F}}$ such that*

$$\mathsf{lab}(t, \sigma) \ \to^*_{\mathsf{Decr}} \ u \ \to_{\overline{R}} \ \mathsf{lab}(t', \sigma).$$

Proof: If $t = l^\tau$ and $t' = r^\tau$ for some rule $l \to r$ of R and some $\tau : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ we obtain from lemma 2

$$\mathsf{lab}(t, \sigma) = \mathsf{lab}(l, \sigma')^{\overline{\tau}} \ \to_{\overline{R}} \ \mathsf{lab}(r, \sigma')^{\overline{\tau}} = \mathsf{lab}(t', \sigma),$$

hence the assertion holds.

Write $\rightsquigarrow$ for the composition of $\to^*_{\mathsf{Decr}}$ and $\to_{\overline{R}}$. Let $t \to_R t'$ and $\mathsf{lab}(t, \sigma) \ \rightsquigarrow \ \mathsf{lab}(t', \sigma)$. We still have to prove that

$$\mathsf{lab}(f(\ldots, t, \ldots), \sigma) \ \rightsquigarrow \ \mathsf{lab}(f(\ldots, t', \ldots), \sigma).$$

According to lemma 6 and the fact that π_f is weakly monotone in all coordinates, we obtain

$$\pi_f(\ldots, \phi(t, \sigma), \ldots) \ \geq \ \pi_f(\ldots, \phi(t', \sigma), \ldots).$$

Hence

$$\begin{array}{rcl}
\mathsf{lab}(f(\ldots, t, \ldots), \sigma) & = & f_{\pi_f(\ldots, \phi(t, \sigma), \ldots)}(\ldots, \mathsf{lab}(t, \sigma), \ldots) \\
& \to^*_{\mathsf{Decr}} & f_{\pi_f(\ldots, \phi(t', \sigma), \ldots)}(\ldots, \mathsf{lab}(t, \sigma), \ldots) \\
& \rightsquigarrow & f_{\pi_f(\ldots, \phi(t', \sigma), \ldots)}(\ldots, \mathsf{lab}(t', \sigma), \ldots) \\
& = & \mathsf{lab}(f(\ldots, t', \ldots), \sigma).
\end{array}$$

□

Theorem 8 *Let $\mathcal{M}$ be a quasi-model for a TRS R over $\mathcal{F}$. Let $\overline{R}$ and Decr be as above for any choice of S_f and π_f. Then R is terminating if and only if $\overline{R} \cup \mathsf{Decr}$ is terminating.*

Proof: Assume $\overline{R} \cup \mathsf{Decr}$ allows an infinite reduction. Since the order on S_f is well-founded for all $f \in \mathcal{F}$, the system Decr is terminating. So the infinite reduction of $\overline{R} \cup \mathsf{Decr}$ contains infinitely many $\overline{R}$-steps. Then removing all labels yields an infinite reduction of R.

On the other hand assume that R allows an infinite reduction. Then applying lab for a fixed substitution on this infinite reduction yields an infinite reduction of $\overline{R} \cup \mathsf{Decr}$ according to lemma 7. □

This proof is very similar to the proof of theorem 4. In fact theorem 4 can be considered as a special case of theorem 8 by choosing the discrete order (i.e., $x \geq y$ if and only if $x = y$) on both $\mathcal{M}$ and S_f. In this special case the requirements of weak monotonicity are trivially fulfilled, the notions of model and quasi-model coincide, and the TRS Decr is empty.

Again consider the TRS introduced at the beginning of this section. The constant c serves as a bottom element: anything can be rewritten to c, but not the other way around. The elements a and b are essentially distinct. So choose the model $\mathcal{M}$ to consist of three elements a, b and c with $a > c$ and $b > c$; a and b are incomparable. By choosing $a_\mathcal{M} = a$, $b_\mathcal{M} = b$, $c_\mathcal{M} = c$ and $f_\mathcal{M}(x, y, z) = c$ for all x, y, z we have a quasi-model. Define $S_f = \{0, 1\}$ with $1 > 0$, and

$$\pi_f(x, y, z) = \begin{cases} 1 & \text{if } x = a \wedge y = b \\ 0 & \text{otherwise.} \end{cases}$$

One easily checks that π_f is weakly monotone in all three coordinates. Now $\overline{R}$ consists of the rules

$$\begin{array}{lcl} f_1(a, b, x) & \to & f_0(x, x, x) \\ f_0(x, y, z) & \to & c \\ f_1(x, y, z) & \to & c \\ a & \to & c \\ b & \to & c \end{array}$$

and Decr consists of the rule

$$f_1(x, y, z) \to f_0(x, y, z).$$

The system $\overline{R} \cup \mathsf{Decr}$ is easily proved to be terminating by choosing the interpretation

$$[a] = [b] = 2, \ [c] = 1, \ [f_0](x, y, z) = x + y + z, \ [f_1](x, y, z) = x + y + 3z$$

over the positive integers. Hence according to theorem 8 the original system is terminating.

In Appendix A of [2] termination of the TRS describing an algebra of communicating processes was proved by first transforming it to another TRS. This transformation is a particular case of our construction, and the proof of preservation of termination is a particular case of theorem 8.

One can wonder whether it is essential in theorem 8 to add the system Decr to the labelled system. It is indeed; consider the following example: R consists of one rule

$$f(g(x)) \to g(g(f(f(x)))).$$

Choose $M = S_f = \{0, 1\}$ with $0 < 1$, let $f_{\mathcal{M}}(x) = 1$ and $g_{\mathcal{M}}(x) = 0$ for all x. Clearly $\mathcal{M}$ is a quasi-model for R. Choose π_f to be the identity which is clearly monotone. Then the system $\overline{R}$ consists of the two rules

$$\begin{array}{rcl} f_0(g(x)) & \to & g(g(f_1(f_0(x)))) \\ f_0(g(x)) & \to & g(g(f_1(f_1(x)))) \end{array}$$

and is terminating: choose the interpretation

$$[f_0](x) = 3x,\ [f_1](x) = x,\ [g](x) = x + 1$$

over the positive integers. However, R is not terminating since it allows the infinite reduction

$$f(f(g(x))) \to f(g(g(f(f(x))))) \to g(g(\underbrace{f(f(g}(f(f(x))))))) \to \cdots.$$

By similar examples one can show that weak monotonicity of both $f_{\mathcal{M}}$ and π_f are essential.

6 Monotone algebras

A *well-founded monotone $\mathcal{F}$-algebra* $(\mathcal{A}, >)$ is defined to be an $\mathcal{F}$-algebra $\mathcal{A}$ for which the underlying set is provided with a well-founded strict partial order $>$ and each algebra operation is strictly monotone in all of its coordinates, more precisely: for each operation symbol $f \in \mathcal{F}$ and all $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ for which $a_i > b_i$ for some i and $a_j = b_j$ for all $j \neq i$ we have

$$f_{\mathcal{A}}(a_1, \ldots, a_n) > f_{\mathcal{A}}(b_1, \ldots, b_n).$$

Note the difference with the partial orders as they occurred in section 5: there operations were weakly monotone and here they are strictly monotone.

We define the partial order $>_{\mathcal{A}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ as follows:

$$t >_{\mathcal{A}} t' \iff (\forall \alpha \in A^{\mathcal{X}} : \phi_{\mathcal{A}}(t, \alpha) > \phi_{\mathcal{A}}(t', \alpha)),$$

where $\phi_{\mathcal{A}}$ is defined as in section 2. Intuitively: $t >_{\mathcal{A}} t'$ means that for each interpretation of the variables in A the interpreted value of t is greater than that of t'.

In [17] the following characterization of termination was given.

Theorem 9 *A TRS R over $\mathcal{F}$ is terminating if and only if there is a non-empty well-founded monotone $\mathcal{F}$-algebra $(\mathcal{A}, >)$ for which $l >_{\mathcal{A}} r$ for every rule $l \to r$ of R.*

If $l >_{\mathcal{A}} r$ for every rule $l \to r$ of R we say that $(\mathcal{A}, >)$ is *compatible* with R. Using this characterization we now sketch alternative proofs of theorems 4 and 8; in fact this was the line along which semantic labelling was discovered. Since theorem 4 is a special case of theorem 8 we concentrate on theorem 8. The interesting direction of the theorem is proving termination of R from termination of $\overline{R} \cup \mathsf{Decr}$. So assume that $\overline{R} \cup \mathsf{Decr}$ is terminating. Then it admits a compatible well-founded monotone $\overline{\mathcal{F}}$-algebra $(\overline{\mathcal{A}}, >)$. We define the well-founded monotone $\mathcal{F}$-algebra $(\mathcal{A}, >)$ by choosing $A = M \times \overline{A}$ as the carrier set, where M is the carrier set of the model $\mathcal{M}$ and $\overline{A}$ is the carrier set of $(\overline{\mathcal{A}}, >)$. As the order we define

$$(m, a) > (m', a') \iff m \geq m' \wedge a > a';$$

clearly it is well-founded. As operations we choose

$$f_A((m_1, a_1), \ldots, (m_n, a_n)) = (f_{\mathcal{M}}(m_1, \ldots, m_n), f_{s,\overline{\mathcal{A}}}(a_1, \ldots, a_n)),$$

where $s = \pi_f(m_1, \ldots, m_n)$. It can be checked straigtforwardly that $(\mathcal{A}, >)$ is compatible with R, so R is terminating.

A similar proof of theorem 5 using theorem 9 can be given, even of a "quasi-model" version of theorem 5, generalizing both theorem 8 and theorem 5.

7 Semantic path order

Let $\succeq$ be any quasi-ordering on terms, i.e., $\succeq$ is reflexive and transitive. Write $t \succ u$ for $t \succeq u$ and not $u \succeq t$, and write $t \approx u$ for $t \succeq u$ and $u \succeq t$. The quasi-ordering $\succeq$ is called well-founded if the strict partial order $\succ$ is well-founded. The *semantic path order* $\succeq_{spo}$ on terms is defined recursively as follows: $s = f(s_1, \ldots, s_m) \succeq_{spo} g(t_1, \ldots, t_n) = t$ if and only if one of the following conditions holds

- $s_i \succeq_{spo} t$ for some $i = 1, \ldots, m$,
- $s \succ t$ and $s \succ_{spo} t_j$ for all $j = 1, \ldots, n$,
- $s \approx t$ and $\{s_1, \ldots, s_m\} \succeq_{M,spo} \{t_1, \ldots, t_n\}$,

where $u \succ_{spo} u'$ means $u \succeq_{spo} u'$ and not $u' \succeq_{spo} u$, and $\succeq_{M,spo}$ is the multiset ordering induced by $\succeq_{spo}$. The basic theorem ([7, 3]) motivating this order is the following:

Theorem 10 *A TRS R is terminating if and only if there is a well-founded quasi-ordering $\succeq$ on terms such that $t \to_R u \Rightarrow f(\ldots, t, \ldots) \succeq f(\ldots, u, \ldots)$ holds for all terms and $l^\sigma \succ_{spo} r^\sigma$ holds for all rules $l \to r$ in R and all substitutions σ.*

If $\geq$ is a well-founded quasi-ordering on the set $\mathcal{F}$ of operation symbols and $\succeq$ is defined by

$$f(s_1, \ldots, s_m) \succeq g(t_1, \ldots, t_n) \iff f \geq g$$

then the corresponding semantic path order is called *recursive path order* (RPO). A recent extension of semantic path order is given in [5].

For practical applications the following observations are useful. Define the subterm relation $\subseteq$ recursively by $s \subseteq t = f(t_1, \ldots, t_n)$ if and only if $s = t$ or $\exists i : s \subseteq t_i$. Write $s \subset t$ for $s \subseteq t \wedge s \neq t$. If $t \subset s$ then we may conclude $s \succ_{spo} t$. Further if for all $u \subseteq t$ we have either $s \succ u$ or $u \subset s$ we also may conclude that $s \succ_{spo} t$. The 'only if' part of the theorem easily follows from this observation by defining

$$s \succeq t \iff \exists u : s \to^* u \wedge t \subseteq u.$$

A typical example of a termination proof by semantic path order is found in [3]:

$$\begin{array}{lcl} x*(y+1) & \to & (x*(y+(1*0)))+x \\ x*1 & \to & x \\ x+0 & \to & x \\ x*0 & \to & 0 \end{array}$$

which is not simply terminating. The semantic path order is defined as follows. First choose the obvious model $\mathcal{M}$ in which M consists of the natural numbers and $0,1,+,*$ are interpreted as $0,1,+,*$. Next define $s \succeq t$ if and only if either the head symbol of t is not '$*$', or

$$s = s_1 * s_2 \wedge t = t_1 * t_2 \wedge \forall \sigma : \phi(s_2, \sigma) \geq \phi(t_2, \sigma).$$

Here ϕ is defined as in section 2. Now one can check all proof obligations of theorem 10, concluding that the system is terminating.

Using similar ingredients we can give a termination proof of the same system by semantic labelling: choose the same $\mathcal{M}$, label '$*$' by the naturals and define $\pi_*(x,y) = y$. The resulting labelled system is

$$\begin{array}{lcl} x *_{i+1} (y+1) & \to & (x *_i (y+(1 *_0 0)))+x \\ x *_1 1 & \to & x \\ x+0 & \to & x \\ x *_0 0 & \to & 0 \end{array}$$

for all $i \geq 0$. We can give the termination proof of this labelled system by RPO. Then the structure of the complete termination proof is essentially the same as that of Dershowitz; labelling is only used to split up the definition of $\succeq$ in two layers.

However, we are not forced to use a path order like approach to prove termination of the labelled system, for example the interpretation in the naturals ≥ 2 defined by $[0] = [1] = 2, x[+]y = x+y, x[*_i]y = x*(y+4i)$ provides another termination proof. In this latter approach the symbol '$+$' is interpreted by a commutative and associative operation, so the labelled system is even terminating modulo commutativity and associativity of '$+$'. Also in the model $\mathcal{M}$ the operation $+$ is commutative and associative. According to theorem 5 we conclude that the original system is terminating modulo commutativity and associativity of '$+$'.

Finally, using the latter approach one easily proves by induction on the depth that a term of depth d can not have reductions of length greater then $2^{2^{C*d}}$ for some constant C. Semantic path order does not provide tools for deriving such bounds.

8 Conclusions and further research

We introduced semantic labelling as a new technique for proving termination of term rewriting systems. The starting point is a model for a TRS, i.e., a model in which each left hand side of a rewrite rule has the same value as the corresponding right hand side. An operation symbol in a term can now be labelled in a way depending on the interpretation of its arguments in the model. This is applied to all rewrite rules. We proved that the labelled TRS is terminating if and only if the original TRS is terminating. We illustrated this new technique for proving termination by several examples. In the typical case the TRS whose termination has to be proved is not simply terminating, while the labelled TRS is proved terminating by RPO or by an interpretation in the natural numbers.

Globally we distinguish two ways of using this technique. In the first way we choose a model which reflects the original semantics of the TRS, e.g., for a system describing quicksort we choose lists and in a system describing the factorial function we choose the natural numbers. In the second way we choose an artificial finite model reflecting syntactic properties we recognize in the rewrite rules. For example, in a rule

$$\cdots f(g(\cdots))\cdots \rightarrow \cdots f(h(\cdots))\cdots$$

the f's can be forced to obtain distinct labels by choosing the images of g and h in the model to be distinct. In section 5 we saw that the requirement of having a model for the TRS can essentially be weakened. This technique also works for termination modulo equations.

The technique of semantic labelling is hard to automate since it depends on either the knowledge of a semantic model or on heuristics for choosing a model in a syntactic way. A promising approach of using labelling without any model to avoid this drawback is the following. Choose the labelling in which every operation symbol in a term is labelled by the head symbols of its direct subterms. If the original signature is finite then the labelled signature is still finite. By applying the basic version of Kruskal's theorem to this labelled signature, the following generalization of Kruskal's theorem over finite signatures can be derived:

> Let E consist of all rewrite rules
>
> $$f(y_1,\ldots,y_{k-1},C[f(x_1,\ldots,x_n)],y_{k+1},\ldots,y_n) \rightarrow f(x_1,\ldots,x_n)$$
>
> for all operation symbols f and all contexts C. Then $\rightarrow_E^*$ is a well-quasi order.

If we replace E by the system $Emb(\mathcal{F})$ as introduced in section 2 we obtain the basic version of Kruskal's theorem. However, E is more restrictive than $Emb(\mathcal{F})$, so this theorem is more powerful than the basic version. For example, it succeeds in ordering $f(f(x)) > f(g(f(x)))$ (as in the approach of [13, 12]) and even $f(0,1,x) > f(x,x,x)$.

References

[1] Bellegarde, F., and Lescanne, P. Termination by completion. *Applicable Algebra in Engineering, Communication and Computing 1*, 2 (1990), 79–96.

[2] Bergstra, J. A., and Klop, J. W. Algebra of communicating processes with abstraction. *Theoretical Computer Science 37*, 1 (1985), 77–121.

[3] Dershowitz, N. Termination of rewriting. *Journal of Symbolic Computation 3*, 1 and 2 (1987), 69–116.

[4] Drosten, K. *Termersetzungssysteme*, vol. 210 of *Informatik-Fachberichte*. Springer, 1989.

[5] GESER, A. On a monotonic semantic path order. Tech. Rep. 92-13, University of Ulm, 1992.

[6] GNAEDIG, I. Termination of order-sorted rewriting. In *Algebraic and Logic Programming* (1992), H. Kirchner and G. Levi, Eds., vol. 632 of *Lecture Notes in Computer Science*, Springer, pp. 37 – 52.

[7] KAMIN, S., AND LÉVY, J. J. Two generalizations of the recursive path ordering. University of Illinois, 1980.

[8] KAPLAN, S. Simplifying conditional term rewriting systems: unification, termination and confluence. *Journal of Symbolic Computation 4*, 3 (1987), 295–334.

[9] KURIHARA, M., AND OHUCHI, A. Modularity of simple termination of term rewriting systems. *Journal of IPS Japan 31*, 5 (1990), 633–642.

[10] LESCANNE, P. Well rewrite orderings and well quasi-orderings. *Journal of Symbolic Computation 14* (1992), 419–435.

[11] OHLEBUSCH, E. A note on simple termination of infinite term rewriting systems. Tech. Rep. 7, Universität Bielefeld, 1992.

[12] PUEL, L. Embedding with patterns and associated recursive path ordering. In *Proceedings of the 3rd Conference on Rewriting Techniques an Applications* (1989), N. Dershowitz, Ed., vol. 355 of *Lecture Notes in Computer Science*, Springer, pp. 371–387.

[13] PUEL, L. Using unavoidable sets of trees to generalize Kruskal's theorem. *Journal of Symbolic Computation 8* (1989), 335–382.

[14] STEINBACH, J. Extensions and comparison of simplification orderings. In *Proceedings of the 3rd Conference on Rewriting Techniques an Applications* (1989), N. Dershowitz, Ed., vol. 355 of *Lecture Notes in Computer Science*, Springer, pp. 434–448.

[15] STEINBACH, J., AND KÜHLER, U. Check your ordering — termination proofs and open problems. Tech. Rep. SR-90-25, University of Kaiserslautern, 1990.

[16] TOYAMA, Y. Counterexamples to termination for the direct sum of term rewriting systems. *Information Processing Letters 25* (1987), 141–143.

[17] 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. Full version appeared as report RUU-CS-92-14, Utrecht University.