Termination of context-sensitive rewriting

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

Abstract

Context-sensitive term rewriting is a kind of term rewriting in which reduction is not allowed inside some fixed arguments of some function symbols. We introduce two new techniques for proving termination of context-sensitive rewriting. The first one is a modification of the technique of interpretation in a well-founded order, the second one is implied by a transformation in which context-sensitive termination of the original system can be concluded from termination of the transformed one. In combination with purely automatic techniques for proving ordinary termination, the latter technique is purely automatic too.

1 Introduction

The function computing the factorial is usually defined as follows:

$$\mathit{fact}(x) = \mathit{if}(x = 0, 1, x * \mathit{fact}(x - 1)),$$

together with some standard rules like $\mathit{if}(\mathit{true}, x, y) = x$ and $\mathit{if}(\mathit{false}, x, y) = y$. Considered as a term rewriting system however, the rule

$$\mathit{fact}(x) \to \mathit{if}(x = 0, 1, x * \mathit{fact}(x - 1))$$

is not terminating. Apparently here general term rewriting does not reflect what is intended by the definition. In the definition the purpose is that first the first argument of *if* will be reduced to *false* or *true*, then the rules $\mathit{if}(\mathit{true}, x, y) = x$ and $\mathit{if}(\mathit{false}, x, y) = y$ will be used to eliminate the *if*-symbol. In describing this computation process in terms of rewriting, we see that in the intended computation never a redex is reduced inside the second or third argument of an *if*-symbol. At this point it turns out to be natural to define a kind of restricted rewriting corresponding to usual rewriting with the extra restriction that reduction inside the second or third argument of an *if*-symbol is not allowed. More general, for every symbol we can define inside which of its arguments reduction is allowed or not. This kind of rewriting is called *context-sensitive rewriting.* It is introduced and discussed by Salvador Lucas, [5, 7, 6]. For further motivation we refer to those papers, here the emphasis is on introduction and justification of new techniques for proving termination of context-sensitive rewriting, shortly denoted as context-sensitive termination.

First we extend the well-known notions of reduction orders and monotone algebras to generalize to the framework of context-sensitive rewriting, arriving at two if-and-only-if-characterizations of context-sensitive termination. The latter one about monotone algebras implies a practical technique for proving context-sensitive termination, quite similar to polynomial interpretations as used for proving ordinary termination.

A first investigation of proving context-sensitive termination was given in [7]. The main result was that context-sensitive termination of a rewrite system can be concluded from ordinary termination of a transformed system. Roughly speaking in the transformed system all arguments at forbidden positions are removed in the left and right hand sides of the rules. However, in most realistic examples the transformed system is not terminating, or is not even a well-defined rewrite system since the proposed right hand sides contain variables that are not in the corresponding left hand sides.

Our main result is the presentation and justification of a more involved transformation for which context-sensitive termination of a rewrite system also can be concluded from ordinary termination of the transformed system. In our transformation the transformed system is always well-defined as a rewrite system, while termination of the transformed system can often be proved fully automatic by means of recursive path order or Knuth-Bendix order. The definition of the transformation is very simple and fully constructive, by which the combination with well-known automatic techniques for proving ordinary termination yields a fully automatic technique for proving context-sensitive termination. Roughly speaking, compared with the technique of [7], in our transformation the arguments at forbidden positions are marked instead of removed. For correctness some extra rules have to be added for handling the unmarking needed when in a reduction forbidden positions change into allowed positions.

By means of our transformation we show how context-sensitive termination can be proved fully automatic for a number of non-terminating examples describing recursive programs with an if-then-else-construction or selecting arguments in infinite lists. For none of the examples the method from [7] is applicable.

2 Preliminaries

Let $\mathcal{T}(\mathcal{F},\mathcal{X})$ denote the set of terms over a signature $\mathcal{F}$ and a set of variables $\mathcal{X}$, where every $f \in \mathcal{F}$ has a fixed arity $ar(f)$. A *rewrite rule over* $\mathcal{F}$ is defined to be a pair of terms $l \to r$ with $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $l \notin \mathcal{X}$, and all variables in r also occur in l. A *term rewriting system* (TRS) over $\mathcal{F}$ is defined to be of a set of rewrite rules over $\mathcal{F}$. A map $\mu : \mathcal{F} \to \mathcal{P}(\mathbb{N})$ is called a *replacement map for* $\mathcal{F}$ if $1 \leq i \leq ar(f)$ for all $i \in \mu(f)$, for all $f \in \mathcal{F}$. For any constant c we have $\mu(c) = \emptyset$ for every replacement map μ, hence in defining a replacement map the images for constants may be left implicit.

If R is a TRS over $\mathcal{F}$ and μ is a replacement map for $\mathcal{F}$, then the corresponding context-sensitive rewrite relation $\hookrightarrow_{R,\mu}$ is defined inductively to be the least relation satisfying

- $l^\sigma \hookrightarrow_{R,\mu} r^\sigma$ for all rules $l \to r$ in R and all $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$;
- if $t \hookrightarrow_{R,\mu} u$, $f \in \mathcal{F}$ and $i \in \mu(f)$ then
  $$f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \hookrightarrow_{R,\mu} f(t_1, \ldots, t_{i-1}, u, t_{i+1}, \ldots, t_n)$$
  for all $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$.

Intuitively, the replacement map describes the arguments in which rewriting is allowed. The definition of $\hookrightarrow_{R,\mu}$ given in [5, 7, 6] looks slightly different: there it is based on the notation for positions as being strings of natural number. However, it is easy to check that both definitions are equivalent. We chose our definition since it directly reflects the inductive structure of $\hookrightarrow_{R,\mu}$ and we never need the position notation.

The usual rewrite relation $\to_R$ is a particular case of the context-sensitive rewrite relation, namely $\to_R = \hookrightarrow_{R,\mu}$ for the replacement map μ defined by $\mu(f) = \{1, \ldots, ar(f)\}$ for all $f \in \mathcal{F}$.

A binary relation $\to$ is called *terminating* if no infinite sequence of elements t_i exists for which $t_i \to t_{i+1}$ for all $i \in \mathbb{N}$. A TRS R is called terminating if $\to_R$ is terminating; for a replacement map μ it is called μ-terminating if $\hookrightarrow_{R,\mu}$ is terminating.

If $\mu(f) \subseteq \mu'(f)$ for all $f \in \mathcal{F}$ and two replacement maps μ, μ', then clearly every $\hookrightarrow_{R,\mu}$-reduction is a $\hookrightarrow_{R,\mu'}$-reduction too, as was already remarked in [5]. Hence μ-termination follows from μ'-termination, in particular μ-termination follows from termination, as was already remarked in [7].

For proving termination of TRSs many techniques have been developed. Standard techniques for automatic proving termination include recursive path order and Knuth-Bendix order; for overviews we refer to [4, 8]. A more recent approach for an automatic technique is proposed in [1]. Often stronger but less automatic techniques are polynomial interpretations ([3]), transformation order ([2]) and semantic labelling ([10]).

3 Context-sensitive reduction orders and interpretations

Let μ be a replacement map for a signature $\mathcal{F}$. A *μ-reduction order* on $\mathcal{T}(\mathcal{F},\mathcal{X})$ is defined to be a well-founded order $>$ on $\mathcal{T}(\mathcal{F},\mathcal{X})$ satisfying

- if $t > u$ and $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$ then $t^\sigma > u^\sigma$, and
- if $t > u$, $f \in \mathcal{F}$ and $i \in \mu(f)$ then
$$f(t_1,\ldots,t_{i-1},t,t_{i+1},\ldots,t_n) > f(t_1,\ldots,t_{i-1},u,t_{i+1},\ldots,t_n)$$
for all $t_1,\ldots,t_{i-1},t_{i+1},\ldots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{X})$.

A μ-reduction order $>$ on $\mathcal{T}(\mathcal{F},\mathcal{X})$ is called *compatible* with a TRS R over $\mathcal{F}$ if $l > r$ for every rewrite rule $l \to r$ in R. For the replacement map μ defined by $\mu(f) = \{1,\ldots,ar(f)\}$ for all $f \in \mathcal{F}$, the notion of μ-reduction order coincides with the usual notion of reduction order.

Proposition 1 *A TRS R is μ-terminating if and only if it admits a compatible μ-reduction order $>$ on $\mathcal{T}(\mathcal{F},\mathcal{X})$.*

Proof: If R is μ-terminating then choose $>$ to be $\hookrightarrow^+_{R,\mu}$. It is a well-founded order since $\hookrightarrow_{R,\mu}$ is terminating and it satisfies both requirements for being a μ-reduction order due to the definition of $\hookrightarrow_{R,\mu}$.

On the other hand, if $>$ is a μ-reduction order that is compatible with R, then one proves that $t > u$ for any t, u satisfying $t \hookrightarrow_{R,\mu} u$ by induction on the definition of $\hookrightarrow_{R,\mu}$. Since $>$ is well-founded, the relation $\hookrightarrow_{R,\mu}$ is terminating, hence R is μ-terminating. □

As usual, an $\mathcal{F}$-algebra is defined to consist of a set A, and for every $f \in \mathcal{F}$ a function $f_A : A^n \to A$, where $n = ar(f)$. Write $\mathcal{F}_A$ for the collection of all algebra operations f_A.

A *μ-monotone $\mathcal{F}$-algebra* $(A, \mathcal{F}_A, >)$ is defined to be an $\mathcal{F}$-algebra $(A, \mathcal{F}_A)$ provided with an order $>$ on A such that each algebra operation is strictly monotone in every μ-argument. More precisely, if $f \in \mathcal{F}$, $i \in \mu(f)$ and $a > b$ for $a, b \in A$, then
$$f_A(a_1,\ldots,a_{i-1},a,a_{i+1},\ldots,a_n) > f(a_1,\ldots,a_{i-1},b,a_{i+1},\ldots,a_n)$$
for all $a_1,\ldots,a_{i-1},a_{i+1},\ldots,a_n \in A$. A μ-monotone $\mathcal{F}$-algebra $(A, \mathcal{F}_A, >)$ is called *well-founded* if the order $>$ on A is well-founded.

For $\alpha : \mathcal{X} \to A$ we define the term evaluation $[\alpha] : \mathcal{T}(\mathcal{F},\mathcal{X}) \to A$ inductively by
$$\begin{array}{rcl}[\alpha](x) & = & \alpha(x), \\ [\alpha](f(t_1,\ldots,t_n)) & = & f_A([\alpha](t_1),\ldots,[\alpha](t_n))\end{array}$$

for $x \in \mathcal{X}, f \in \mathcal{F}, t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Write $t >_A t'$ if and only if $[\alpha](t) > [\alpha](t')$ for all $\alpha : \mathcal{X} \to A$. Intuitively, $t >_A t'$ means that for each interpretation of the variables in A the interpreted value of t is greater than that of t'.

Proposition 2 *Let μ be a replacement map and let $(A, \mathcal{F}_A, >)$ be a non-empty well-founded μ-monotone $\mathcal{F}$-algebra. Then the relation $>_A$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ defined above is a μ-reduction order.*

Proof: The relation $>_A$ is irreflexive since $A \neq \emptyset$ and $>$ is irreflexive. It is transitive and well-founded since $>$ is transitive and well-founded.

Next observe that $[\alpha](t^\sigma) = [[\alpha] \circ \sigma](t)$ for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}), \sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X}), \alpha : \mathcal{X} \to A$, which is easily proved by induction on the structure of t. Then

$$[\alpha](t^\sigma) = [[\alpha] \circ \sigma](t) > [[\alpha] \circ \sigma](u) = [\alpha](u^\sigma)$$

for $t >_A u$ and all $\alpha : \mathcal{X} \to A$, hence $t^\sigma >_A u^\sigma$.

The remaining property required in the definition of μ-reduction order follows from the fact that each algebra operation is strictly monotone in every μ-argument. □

Proposition 3 *Let μ be a replacement map over $\mathcal{F}$. A TRS R over $\mathcal{F}$ is μ-terminating if and only if a non-empty well-founded μ-monotone $\mathcal{F}$-algebra $(A, \mathcal{F}_A, >)$ exists for which $>_A$ is compatible with R.*

Proof: The 'if'-part is immediate from Propositions 1 and 2.

For the 'only if'-part, assume R is μ-terminating. Define $A = \mathcal{T}(\mathcal{F}, \mathcal{X})$, let $f_A(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$, and define $>$ to coincide with $\hookrightarrow^+_{R,\mu}$. One easily verifies that $(A, \mathcal{F}_A, >)$ is a non-empty well-founded μ-monotone algebra. We still have to prove that $l >_A r$ for each rewrite rule $l \to r$. Let $\alpha : \mathcal{X} \to A$. Since $A = \mathcal{T}(\mathcal{F}, \mathcal{X})$ we see that α is a substitution. Then $[\alpha](t) = t^\alpha$ for each term t, which is easily proved by induction on the structure of t. Since $l \to r$ is a rewrite rule, the term l^α can be reduced in one step to r^α. So

$$[\alpha](l) \;=\; l^\alpha \;>\; r^\alpha \;=\; [\alpha](r).$$

This holds for every $\alpha : \mathcal{X} \to A$, so $l >_A r$, which we had to prove. □

For the replacement map μ defined by $\mu(f) = \{1, \ldots, ar(f)\}$ for all $f \in \mathcal{F}$, the notion of well-founded μ-monotone algebra coincides with the notion of well-founded monotone algebra from [9]; in this way Proposition 1 in [9] is a special case of Proposition 3.

The way of proving μ-termination of a TRS is now as follows: choose a well-founded partially ordered set $(A, >)$, define for each operation symbol f a corresponding operation f_A that is strictly monotone in every μ-argument, and for which $[\alpha](l) > [\alpha](r)$ for all rewrite rules $l \to r$ and all $\alpha : \mathcal{X} \to A$. Then according to Proposition 3 the TRS is terminating.

The problem is how to choose the partially ordered set and the operations. The simplest useful choice for $(A, >)$ is $(\mathbb{N}^+, >)$, the set of strictly positive integers with the ordinary ordering. In many applications this is a fruitful choice.

Example 1. Let R consist of the rules

$$\begin{array}{rcl} g(x) & \to & h(x) \\ c & \to & d \\ h(d) & \to & g(c) \end{array}$$

The cyclic reduction $g(c) \to_R h(c) \to_R h(d) \to_R g(c)$ shows that R is not terminating, and not even μ-terminating if $1 \in \mu(h)$. The cyclic reduction $g(c) \to_R g(d) \to_R h(d) \to_R g(c)$ shows that R is not μ-terminating if $1 \in \mu(g)$. Now we prove that R is μ-terminating for μ defined by $\mu(g) = \mu(h) = \emptyset$. We choose the well-founded μ-monotone algebra A consisting of positive integers with the usual order, where we choose $c_A = 2, d_A = 1, g_A(1) = 4, h_A(1) = 3, g_A(x) = x, h_A(x) = x - 1$ for all $x > 1$. Indeed we have

$$\begin{array}{rcll} g_A(x) & > & h_A(x) & \qquad \text{for all } x \in A, \\ c_A & > & d_A, & \\ h_A(d_A) = 3 & > & 2 = g_A(c_A), & \end{array}$$

proving that R is μ-terminating by Proposition 3. Note that neither g_A nor h_A is strictly monotone in its argument.

Example 2. From [6] we take the following rewrite system

$$\begin{array}{rcl} \mathit{sel}(0, x : y) & \to & x \\ \mathit{sel}(s(x), y : z) & \to & \mathit{sel}(x, z) \\ \mathit{from}(x) & \to & x : \mathit{from}(s(x)). \end{array}$$

Here ':' is the cons function: $x : y$ means that the element x is put in front of the list y. The first two rules describe a general mechanism to select the n-th element from an arbitrary list. The last rule describes a definition of a particular infinite list. The combination can be used to compute the n-th element of this list, for instance

$$\mathit{sel}(s(s(0)), \mathit{from}(0)) \to \mathit{sel}(s(s(0)), 0 : \mathit{from}(s(0))) \to$$

$$\mathit{sel}(s(0), \mathit{from}(s(0))) \to \mathit{sel}(s(0), s(0) : \mathit{from}(s(s(0)))) \to$$

$$\mathit{sel}(0, \mathit{from}(s(s(0)))) \to \mathit{sel}(0, s(s(0)) : \mathit{from}(s(s(s(0))))) \to s(s(0)).$$

The strategy applied here corresponds to a lazy evalution strategy, which in this case means that no reduction is allowed inside an argument of the symbol

':'. In terms of context-sensitive rewriting this corresponds to choosing $\mu(:) = \emptyset$. For this system the essential point is that no reduction is allowed inside the right argument of ':'; this weakening of the restriction is given by defining $\mu(:) = \{1\}$. For the other symbols we do not have restrictions and we choose $\mu(s) = \mu(\mathit{from}) = \{1\}$, $\mu(\mathit{sel}) = \{1,2\}$. Clearly the system is not terminating, but it is μ-terminating as can be proved as follows. We choose the well-founded μ-monotone algebra A consisting of positive integers with the usual order, where we choose $0_A = 1, s_A(x) = x+1, x :_A y = x+(y \text{ div } 2), \mathit{sel}_A(x,y) = 2^x*y, \mathit{from}_A(x) = 2x+4$ for all $x, y \in A$. One easily verifies that s_A, sel_A and from_A are strictly monotone in all of their arguments, and that $:_A$ is strictly monotone in its first argument (not in its second). Further we have

$$\mathit{sel}_A(0_A, x :_A y) = 2*(x+(y \text{ div } 2)) > x$$

$$\mathit{sel}_A(s_A(x), y :_A z) = 2^{x+1}*(y+(z \text{ div } 2)) > 2^x * z = \mathit{sel}_A(x,z)$$

$$\mathit{from}_A(x) = 2x+4 > 2x+3 = x :_A \mathit{from}_A(s_A(x)),$$

for all $x, y, z > 0$, proving that R is μ-terminating by Proposition 3.

Proposition 3 is not only useful for proving context-sensitive termination of particular rewrite systems, it can also be used for obtaining theoretical results. In particular it implies a simpler proof of the main result of [7]. There for a TRS R and a replacement map μ a TRS R^μ is introduced obtained from R by removing all arguments at forbidden positions in all left and right hand sides of the rules of R. It may happen that the result is not a TRS, then it is left undefined. The result of [7] states that R is μ-terminating if R^μ is terminating. A simple proof of this statement can be given as follows: it is easily verified that R^μ is terminating if and only if a non-empty well-founded μ-monotone $\mathcal{F}$-algebra $(A, \mathcal{F}_A, >)$ exists in which $f_A(x_1, \ldots, x_n)$ does not depend on x_i for al $f \in \mathcal{F}$ and $i \notin \mu(f)$ and for which $>_A$ is compatible with R. This property immediately implies that R is μ-terminating by Proposition 3.

4 A transformational method

Throughout this section we fix a signature $\mathcal{F}$ and a corresponding replacement map μ. We extend the signature $\mathcal{F}$ to a signature $\mathcal{F}'$ by adding a fresh symbol $\overline{f}$ for every $f \in \mathcal{F}$, with $ar(\overline{f}) = ar(f)$ for every $f \in \mathcal{F}$. Further we assume a fresh unary symbol a in $\mathcal{F}'$. The overlining is extended to terms by defining

$$\begin{array}{rcll} \overline{x} & = & x & \text{for } x \in \mathcal{X} \\ \overline{f(t_1, \ldots, t_n)} & = & \overline{f}(t_1, \ldots, t_n) & \text{for } f \in \mathcal{F}, t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}', \mathcal{X}). \end{array}$$

Next we define a function $\Phi : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathcal{T}(\mathcal{F}', \mathcal{X})$ meant to overline root symbols of subterms for which reduction is not allowed. It is defined inductively as follows:

$$\begin{array}{rcll} \Phi(x) & = & x & \text{for } x \in \mathcal{X} \\ \Phi(f(t_1,\ldots,t_n)) & = & f(u_1,\ldots,u_n) & \text{for } f \in \mathcal{F}, \end{array}$$

where $u_i = \Phi(t_i)$ if $i \in \mu(f)$, and $u_i = \overline{\Phi(t_i)}$ if $i \notin \mu(f)$. We define the TRS $\mathsf{Bar}(\mathcal{F})$ over $\mathcal{F}'$ as follows:

$$\mathsf{Bar}(\mathcal{F}) = \left\{ \begin{array}{rcll} a(\overline{f}(x_1,\ldots,x_n)) & \to & f(x_1,\ldots,x_n) & \text{for all } f \in \mathcal{F} \\ f(x_1,\ldots,x_n) & \to & \overline{f}(x_1,\ldots,x_n) & \text{for all } f \in \mathcal{F} \\ a(x) & \to & x & \end{array} \right.$$

The purpose of the system $\mathsf{Bar}(\mathcal{F})$ is that overlining can freely be added, while overlining only may be removed in combination with an a symbol. The latter is needed to be able to remove overlining in case a rule is applied by which a forbidden position changes into an allowed position.

As usual for a term t the set $\mathsf{Var}(t)$ denotes the set of variables in t, inductively defined by $\mathsf{Var}(x) = \{x\}$ for $x \in \mathcal{X}$ and $\mathsf{Var}(f(t_1,\ldots,t_n)) = \bigcup_{i=1}^{n} \mathsf{Var}(t_i)$, for $f \in \mathcal{F}, t_1,\ldots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{X})$. We shall use the set $\mathsf{Forb}(t) \subseteq \mathsf{Var}(t)$ describing variables at forbidden positions, inductively defined by

$$\begin{array}{rcll} \mathsf{Forb}(x) & = & \emptyset & \text{for } x \in \mathcal{X} \\ \mathsf{Forb}(f(t_1,\ldots,t_n)) & = & (\bigcup_{i\in\mu(f)} \mathsf{Forb}(t_i)) \cup (\bigcup_{i\notin\mu(f)} \mathsf{Var}(t_i)) & \end{array}$$

for $f \in \mathcal{F}, t_1,\ldots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{X})$. This means that a variable x occurring in a term t is contained in $\mathsf{Forb}(t)$ if and only if x occurs in a forbidden argument of one of the operation symbols in t.

For any term t we define the substitution $\tau(t)$ as follows:

$$\begin{array}{rcll} x^{\tau(t)} & = & a(x) & \text{if } x \in \mathsf{Forb}(t) \\ x^{\tau(t)} & = & x & \text{if } x \notin \mathsf{Forb}(t). \end{array}$$

Let R be any TRS over $\mathcal{F}$. Then we define

$$\Phi(R) \;=\; \{\; \Phi(l) \to (\Phi(r))^{\tau(l)} \mid l \to r \in R \;\}.$$

For instance, if R is the TRS from Example 2 then $\Phi(R)$ consists of the rules

$$\begin{array}{rcl} \mathit{sel}(0, x : y) & \to & x \\ \mathit{sel}(s(x), y : z) & \to & \mathit{sel}(x, a(z)) \\ \mathit{from}(x) & \to & x : \overline{\mathit{from}}(s(x)). \end{array}$$

Now we can state a basic result; after proving it we will present our main result as an optimization of this proposition.

Proposition 4 *Let R be a TRS over $\mathcal{F}$ for which the TRS $\Phi(R) \cup \mathsf{Bar}(\mathcal{F})$ is terminating. Then R is μ-terminating.*

In order to prove this proposition we need a couple of lemmas.

Lemma 5 *Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then*

$$a(\overline{t}) \to_{\mathsf{Bar}(\mathcal{F})} t \to^*_{\mathsf{Bar}(\mathcal{F})} \overline{t}.$$

Proof: Straightforward. □

Define $\Phi(\sigma), \overline{\Phi(\sigma)} : \mathcal{X} \to \mathcal{T}(\mathcal{F}', \mathcal{X})$ by $x^{\Phi(\sigma)} = \Phi(x^\sigma)$ and $x^{\overline{\Phi(\sigma)}} = \overline{\Phi(x^\sigma)}$ for all $x \in \mathcal{X}$.

Lemma 6 *Let $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ and let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then*

$$\Phi(t^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(t)^{\overline{\Phi(\sigma)}} \quad \textit{and} \quad \overline{\Phi(t^\sigma)} \to^*_{\mathsf{Bar}(\mathcal{F})} \overline{\Phi(t)}^{\overline{\Phi(\sigma)}}.$$

Proof: We apply induction on t. For $t = x \in \mathcal{X}$ we have $\Phi(x^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} \overline{\Phi(x^\sigma)} = x^{\overline{\Phi(\sigma)}} = \Phi(x)^{\overline{\Phi(\sigma)}}$ by Lemma 5, and $\overline{\Phi(x^\sigma)} = x^{\overline{\Phi(\sigma)}} = \overline{\Phi(x)}^{\overline{\Phi(\sigma)}}$, and we are done.

For $t = f(t_1, \ldots, t_n)$ we write $u_i = \Phi(t_i)$ and $v_i = \Phi(t_i^\sigma)$ if $i \in \mu(f)$, and $u_i = \overline{\Phi(t_i)}$ and $v_i = \overline{\Phi(t_i^\sigma)}$ if $i \notin \mu(f)$. Then we have

$$\Phi(t^\sigma) = f(v_1, \ldots, v_n), \quad \text{and} \quad \Phi(t)^{\overline{\Phi(\sigma)}} = f(u_1^{\overline{\Phi(\sigma)}}, \ldots, u_n^{\overline{\Phi(\sigma)}}),$$

$$\overline{\Phi(t^\sigma)} = \overline{f}(v_1, \ldots, v_n), \quad \text{and} \quad \overline{\Phi(t)}^{\overline{\Phi(\sigma)}} = \overline{f}(u_1^{\overline{\Phi(\sigma)}}, \ldots, u_n^{\overline{\Phi(\sigma)}}),$$

hence it suffices to prove that $v_i \to^*_{\mathsf{Bar}(\mathcal{F})} u_i^{\overline{\Phi(\sigma)}}$ for $i = 1, \ldots, n$. In case of $i \in \mu(f)$ this follows from the first half of the induction hypothesis, in case of $i \notin \mu(f)$ this follows from the second half of the induction hypothesis. □

For a substitution σ and a term t the substitution $\rho(\sigma, t)$ is defined by

$$\begin{array}{rcll} x^{\rho(\sigma,t)} & = & \overline{\Phi(x^\sigma)} & \text{if } x \in \mathsf{Forb}(t) \\ x^{\rho(\sigma,t)} & = & \Phi(x^\sigma) & \text{if } x \notin \mathsf{Forb}(t). \end{array}$$

Lemma 7 *Let σ be a substitution and let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then*

$$\Phi(t^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} (\Phi(t))^{\rho(\sigma,t)}.$$

Proof: We apply induction on t. For $t = x \in \mathcal{X}$ we have $x \notin \mathsf{Forb}(x) = \emptyset$, and hence

$$\Phi(x^\sigma) = x^{\rho(\sigma,x)} = (\Phi(x))^{\rho(\sigma,x)}.$$

For $t = f(t_1, \ldots, t_n)$ we again write

$$\Phi(t^\sigma) = \Phi(f(t_1^\sigma, \ldots, t_n^\sigma)) = f(v_1, \ldots, v_n)$$

where $v_i = \Phi(t_i^\sigma)$ if $i \in \mu(f)$, and $v_i = \overline{\Phi(t_i^\sigma)}$ if $i \notin \mu(f)$. As before let $u_i = \Phi(t_i)$ if $i \in \mu(f)$, and $u_i = \overline{\Phi(t_i)}$ if $i \notin \mu(f)$. Since $\Phi(t) = f(u_1, \ldots, u_n)$, it suffices to prove $v_i \to^*_{\mathsf{Bar}(\mathcal{F})} u_i^{\rho(\sigma,t)}$ for $i = 1, \ldots, n$.

First assume $i \notin \mu(f)$. Since $\mathsf{Var}(t_i) \subseteq \mathsf{Forb}(t)$ we have $x^{\rho(\sigma,t)} = \overline{\Phi(x^\sigma)} = x^{\overline{\Phi(\sigma)}}$ for all $x \in \mathsf{Var}(t_i)$. From Lemma 6 we conclude

$$v_i = \overline{\Phi(t_i^\sigma)} \to^*_{\mathsf{Bar}(\mathcal{F})} \overline{\Phi(t_i)}^{\overline{\Phi(\sigma)}} = \overline{\Phi(t_i)}^{\rho(\sigma,t)} = u_i^{\rho(\sigma,t)}$$

which we had to prove.

For the remaining case assume $i \in \mu(f)$. Then $v_i = \Phi(t_i^\sigma)$. From the induction hypothesis we conclude $\Phi(t_i^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(t_i)^{\rho(\sigma,t_i)}$. Since $\mathsf{Forb}(t_i) \subseteq \mathsf{Forb}(t)$ we have $x^{\rho(\sigma,t_i)} = \overline{\Phi(x^\sigma)} = x^{\rho(\sigma,t)}$ for all $x \in \mathsf{Forb}(t_i)$. For $x \notin \mathsf{Forb}(t_i)$ we have $x^{\rho(\sigma,t_i)} = \Phi(x^\sigma)$, and either $x^{\rho(\sigma,t)} = \Phi(x^\sigma)$ or $x^{\rho(\sigma,t)} = \overline{\Phi(x^\sigma)}$. Since $\Phi(x^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} x^{\rho(\sigma,t)}$ by Lemma 5, we have $x^{\rho(\sigma,t_i)} \to^*_{\mathsf{Bar}(\mathcal{F})} x^{\rho(\sigma,t)}$ for all $x \in \mathsf{Var}(t_i)$. Hence

$$v_i = \Phi(t_i^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(t_i)^{\rho(\sigma,t_i)} \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(t_i)^{\rho(\sigma,t)} = u_i^{\rho(\sigma,t)},$$

concluding the proof. □

Lemma 8 *Let $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ and let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then*

$$\Phi(t)^{\Phi(\sigma)} \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(t^\sigma) \quad \textit{and} \quad \overline{\Phi(t)}^{\Phi(\sigma)} \to^*_{\mathsf{Bar}(\mathcal{F})} \overline{\Phi(t^\sigma)}.$$

Proof: Again we apply induction on t. For $t = x \in \mathcal{X}$ we have $\Phi(x)^{\Phi(\sigma)} = x^{\Phi(\sigma)} = \Phi(x^\sigma)$, and $\overline{\Phi(x)}^{\Phi(\sigma)} = x^{\Phi(\sigma)} = \Phi(x^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} \overline{\Phi(x^\sigma)}$ by Lemma 5, and we are done.

For $t = f(t_1, \ldots, t_n)$ we again write $u_i = \Phi(t_i)$ and $v_i = \Phi(t_i^\sigma)$ if $i \in \mu(f)$, and $u_i = \overline{\Phi(t_i)}$ and $v_i = \overline{\Phi(t_i^\sigma)}$ if $i \notin \mu(f)$. Then we have

$$\Phi(t)^{\Phi(\sigma)} = f(u_1^{\Phi(\sigma)}, \ldots, u_n^{\Phi(\sigma)}) \quad \text{and} \quad \Phi(t^\sigma) = f(v_1, \ldots, v_n),$$

$$\overline{\Phi(t)}^{\Phi(\sigma)} = \overline{f}(u_1^{\Phi(\sigma)}, \ldots, u_n^{\Phi(\sigma)}) \quad \text{and} \quad \overline{\Phi(t^\sigma)} = \overline{f}(v_1, \ldots, v_n),$$

hence it suffices to prove that $u_i^{\Phi(\sigma)} \to^*_{\mathsf{Bar}(\mathcal{F})} v_i$ for $i = 1, \ldots, n$. In case of $i \in \mu(f)$ this follows from the first half of the induction hypothesis, in case of $i \notin \mu(f)$ this follows from the second half of the induction hypothesis. □

Lemma 9 *Let $l \to r$ be a rule in R and let σ be a substitution. Then*

$$\Phi(l^\sigma) \to^+_{\Phi(R) \cup \mathsf{Bar}(\mathcal{F})} \Phi(r^\sigma).$$

Proof: From Lemma 7 we conclude $\Phi(l^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(l)^{\rho(\sigma,l)}$. From the definition of $\Phi(R)$ we conclude $\Phi(l)^{\rho(\sigma,l)} \to_{\Phi(R)} (\Phi(r)^{\tau(l)})^{\rho(\sigma,l)}$. For $x \in \mathsf{Var}(l)$ we have $(x^{\tau(l)})^{\rho(\sigma,l)} = a(\overline{\Phi(x^\sigma)})$ if $x \in \mathsf{Forb}(l)$, and $(x^{\tau(l)})^{\rho(\sigma,l)} = \Phi(x^\sigma)$ if $x \notin \mathsf{Forb}(l)$. So for all $x \in \mathsf{Var}(l)$ we have $(x^{\tau(l)})^{\rho(\sigma,l)} \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(x^\sigma) = x^{\Phi(\sigma)}$ by Lemma 5, hence $(\Phi(r)^{\tau(l)})^{\rho(\sigma,l)} \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(r)^{\Phi(\sigma)}$. Combining all these steps and Lemma 8 yields

$$\Phi(l^\sigma) \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(l)^{\rho(\sigma,l)} \to_{\Phi(R)} (\Phi(r)^{\tau(l)})^{\rho(\sigma,l)} \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(r)^{\Phi(\sigma)} \to^*_{\mathsf{Bar}(\mathcal{F})} \Phi(r^\sigma).$$

□

Lemma 10 *Let $t \hookrightarrow_{R,\mu} u$. Then $\Phi(t) \to^+_{\Phi(R)\cup\mathsf{Bar}(\mathcal{F})} \Phi(u)$.*

Proof: We apply induction on the definition of $\hookrightarrow_{R,\mu}$. The base step exactly coincides with Lemma 9. For the induction step assume $t \hookrightarrow_{R,\mu} u$, $\Phi(t) \to^+_{\Phi(R)\cup\mathsf{Bar}(\mathcal{F})} \Phi(u)$, $f \in \mathcal{F}$, $i \in \mu(f)$, $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. We have to prove that

$$\Phi(f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n)) \to^+_{\Phi(R)\cup\mathsf{Bar}(\mathcal{F})} \Phi(f(t_1, \ldots, t_{i-1}, u, t_{i+1}, \ldots, t_n)).$$

This is immediate since we can write

$$\Phi(f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n)) = f(u_1, \ldots, u_{i-1}, \Phi(t), u_{i+1}, \ldots, u_n)$$

and

$$\Phi(f(t_1, \ldots, t_{i-1}, u, t_{i+1}, \ldots, t_n)) = f(u_1, \ldots, u_{i-1}, \Phi(u), u_{i+1}, \ldots, u_n).$$

□

Now we can prove Proposition 4.

Proof: Assume that R is not μ-terminating. Then $\hookrightarrow_{R,\mu}$ admits an infinite reduction. By applying Φ to all terms in this infinite reduction this gives rise to an infinite reduction of $\Phi(R) \cup \mathsf{Bar}(\mathcal{F})$ according to Lemma 10. This contradicts the assumption that $\Phi(R) \cup \mathsf{Bar}(\mathcal{F})$ is terminating. This concludes the proof of Proposition 4. □

It turns out that the overlined symbols that do not occur in $\Phi(R)$ do not play an essential role. They can be eliminated by the following construction. For a TRS R over $\mathcal{F}$ let $\mathcal{F}_0$ consist of the symbols $f \in \mathcal{F}$ for which $\overline{f}$ occurs in $\Phi(l)$ or $\Phi(r)$ for some $l \to r$ in R. Remember that both $\Phi(R)$ and $\mathcal{F}_0$ depend on the choice of the replacement map μ; we fixed μ and omitted μ in our notation for saving space.

Define $\Psi(R, \mu) = \Phi(R) \cup \mathsf{Bar}(\mathcal{F}_0)$; without referring to $\Phi(R)$ and $\mathsf{Bar}(\mathcal{F}_0)$ this means that

$$\Psi(R,\mu) = \left\{ \begin{array}{rcll} \Phi(l) & \to & (\Phi(r))^{\tau(l)} & \text{for all } l \to r \in R, \\ a(\overline{f}(x_1, \ldots, x_n)) & \to & f(x_1, \ldots, x_n) & \text{for all } f \in \mathcal{F}_0, \\ f(x_1, \ldots, x_n) & \to & \overline{f}(x_1, \ldots, x_n) & \text{for all } f \in \mathcal{F}_0, \\ a(x) & \to & x & \end{array} \right.$$

Lemma 11 *The TRS $\Psi(R, \mu)$ is terminating if and only if $\Phi(R) \cup \mathsf{Bar}(\mathcal{F})$ is terminating.*

Proof: Since $\Psi(R, \mu) \subseteq \Phi(R) \cup \mathsf{Bar}(\mathcal{F})$, termination of $\Phi(R) \cup \mathsf{Bar}(\mathcal{F})$ implies termination of $\Psi(R, \mu)$.

For the converse we introduce a function λ removing redundant bars. It is defined inductively as follows.

$$\begin{array}{rcll} \lambda(x) & = & x & \text{for all } x \in \mathcal{X} \\ \lambda(f(t_1, \ldots, t_n)) & = & f(\lambda(t_1), \ldots, \lambda(t_n)) & \text{for all } f \in \mathcal{F} \\ \lambda(\overline{f}(t_1, \ldots, t_n)) & = & \overline{f}(\lambda(t_1), \ldots, \lambda(t_n)) & \text{for all } f \in \mathcal{F}_0 \\ \lambda(\overline{f}(t_1, \ldots, t_n)) & = & f(\lambda(t_1), \ldots, \lambda(t_n)) & \text{for all } f \in \mathcal{F} \setminus \mathcal{F}_0. \end{array}$$

Write $\Phi(R) \cup \mathsf{Bar}(\mathcal{F}) = S \cup T$ where S consists of the rules $f(x_1, \ldots, x_n) \to \overline{f}(x_1, \ldots, x_n)$ for $f \in \mathcal{F} \setminus \mathcal{F}_0$, and T consists of the other rules of $\Phi(R) \cup \mathsf{Bar}(\mathcal{F})$. One easily verifies that $\lambda(t) = \lambda(u)$ for $t \to_S u$ and $\lambda(t) \to_{\Psi(R,\mu)} \lambda(u)$ for $t \to_T u$. Assume that $\Phi(R) \cup \mathsf{Bar}(\mathcal{F}) = S \cup T$ admits an infinite reduction and that $\Psi(R, \mu)$ is terminating. Then applying λ on the terms in the infinite $S \cup T$-reduction yields by the above observation an infinite S-reduction, contradiction. □

Now we arrive at the main theorem.

Theorem 12 *Let R be a TRS over $\mathcal{F}$ and let μ be a replacement map for which the TRS $\Psi(R, \mu)$ is terminating. Then R is μ-terminating.*

Proof: Immediate from Proposition 4 and Lemma 11. □

We like to stress here that given a TRS R and a replacement map μ, the construction of $\Psi(R, \mu)$ is purely mechanical and very simple: R is copied, according to μ some symbols are overlined and in some right hand sides some a-symbols may be added, and for the symbols that are overlined the corresponding $\mathsf{Bar}(\mathcal{F})$-rules are added. Surprisingly, often termination of $\Psi(R, \mu)$ can be proved purely mechanically and simply too by recursive path order, hence automatic proving context-sensitive termination according to Theorem 12, even if R is not terminating itself.

Example 3. As in Example 2 let R consist of the rules

$$\begin{array}{rcl} \mathit{sel}(0, x : y) & \to & x \\ \mathit{sel}(s(x), y : z) & \to & \mathit{sel}(x, z) \\ \mathit{from}(x) & \to & x : \mathit{from}(s(x)), \end{array}$$

describing the computation of the n-th element of a particular infinite list. We choose the same μ, defined by $\mu(:) = \mu(s) = \mu(\mathit{from}) = \{1\}$, $\mu(\mathit{sel}) = \{1, 2\}$. In Example 3 we proved that R is indeed μ-terminating. However, in that proof we needed an ad hoc choice for interpretation in the naturals. Now we can prove μ-termination of R purely mechanically: according to Theorem 12 it suffices to prove termination of $\Psi(R,\mu)$ consisting of the rules

$$\begin{array}{rcl}
\mathit{sel}(0, x : y) & \to & x \\
\mathit{sel}(s(x), y : z) & \to & \mathit{sel}(x, a(z)) \\
\mathit{from}(x) & \to & x : \overline{\mathit{from}}(s(x)) \\
a(\overline{\mathit{from}}(x)) & \to & \mathit{from}(x) \\
\mathit{from}(x) & \to & \overline{\mathit{from}}(x) \\
a(x) & \to & x.
\end{array}$$

Termination of $\Psi(R,\mu)$ is proved by recursive path order: choose a precedence $>$ satisfying $\mathit{sel} > a > \mathit{from} > \overline{\mathit{from}}$, $\mathit{from} > s$ and $\mathit{from} >:$, and give sel the lexicographic status from left to right.

Example 4. The former example is easily extended. For instance by the rules

$$\begin{array}{rcl}
f(x) & \to & x : f(g(x)) \\
g(0) & \to & s(0) \\
g(s(x)) & \to & s(s(g(x)))
\end{array}$$

the term $f(0)$ describes the infinite list of numbers of the shape $2^k - 1$ for $k \geq 0$. By adding the first two rules of the former system we can compute the n-th element of this list. If we choose $\mu(g) = \{1\}$ and $\mu(:), \mu(s)$ and $\mu(\mathit{sel})$ as before, the system $\Psi(R,\mu)$ reads

$$\begin{array}{rclcrcl}
\mathit{sel}(0, x : y) & \to & x & \quad & \mathit{sel}(s(x), y : z) & \to & \mathit{sel}(x, a(z)) \\
f(x) & \to & x : \overline{f}(g(x)) & & g(0) & \to & s(0) \\
g(s(x)) & \to & s(s(g(x))) & & a(\overline{f}(x)) & \to & f(x) \\
f(x) & \to & \overline{f}(x) & & a(x) & \to & x.
\end{array}$$

which is again proved to be terminating by recursive path order: choose a precedence $>$ satisfying $\mathit{sel} > a > f > \overline{f}$, $f > g > s$ and $f >:$, and give sel the lexicographic status from left to right.

Example 5. Let R consist of the rules

$$\begin{array}{rcl}
f(x) & \to & \mathit{if}(x, c, f(\mathit{true}))) \\
\mathit{if}(\mathit{true}, x, y) & \to & x \\
\mathit{if}(\mathit{false}, x, y) & \to & y,
\end{array}$$

describing a simple recursive function with an *if*-construction. From the first rule we see that R is not terminating. In the unfolding of the recursion no reduction in the third argument of *if* is allowed. If we correspondingly choose $\mu(f) = \{1\}$,

$\mu(\mathit{if}) = \{1,2\}$ then indeed we can prove that R is μ-terminating by means of Theorem 12. The TRS $\Psi(R,\mu)$ consists of the rules

$$\begin{array}{rcl@{\qquad}rcl} f(x) & \to & \mathit{if}(x, c, \overline{f}(\mathit{true})) & \mathit{if}(\mathit{true}, x, y) & \to & x \\ \mathit{if}(\mathit{false}, x, y) & \to & a(y) & a(\overline{f}(x)) & \to & f(x) \\ f(x) & \to & \overline{f}(x) & a(x) & \to & x. \end{array}$$

Termination of $\Psi(R,\mu)$ is easily proved by means of a interpretation in positive integers (choose $c_A = \mathit{true}_A = 1, \mathit{false}_A = 4, f_A(x) = x+3, \overline{f}_A(x) = x, a_A(x) = x+4, \mathit{if}_A(x,y,z) = x+y+z$), or by a Knuth-Bendix order.

One can wonder whether the converse of Theorem 12 also holds: is $\Psi(R,\mu)$ terminating if R is μ-terminating? Unfortunately, this does not always hold, as is shown by the next example.

Example 6. Let R consist of the rules

$$\begin{array}{rcl} g(x) & \to & h(x) \\ c & \to & d \\ h(d) & \to & g(c) \end{array}$$

and choose $\mu(g) = \mu(h) = \emptyset$. In Example 1 we saw that R is not terminating but it is μ-terminating. On the other hand $\Psi(R,\mu)$ consists of the rules

$$\begin{array}{rcl@{\qquad}rcl} g(x) & \to & h(a(x)) & c & \to & d \\ h(\overline{d}) & \to & g(\overline{c}) & a(\overline{c}) & \to & c \\ a(\overline{d}) & \to & d & c & \to & \overline{c} \\ d & \to & \overline{d} & a(x) & \to & x \end{array}$$

and admits the infinite cyclic reduction

$$g(\overline{c}) \to h(a(\overline{c})) \to h(c) \to h(d) \to h(\overline{d}) \to g(\overline{c}) \to \cdots.$$

It seems to be possible to refine the definition of $\Psi(R,\mu)$ in such a way that Theorem 12 still holds, and in this example the rule $g(x) \to h(a(x))$ in $\Psi(R,\mu)$ is replaced by $g(x) \to h(x)$, by which $\Psi(R,\mu)$ is terminating. The idea is that in the refinement for a rule $l \to r$ the a-symbols are not plugged in all x-symbols in r for $x \in \mathsf{Forb}(l)$, but only at x-occurrences of r at forbidden positions. We did not elaborate this possible improvement: even then we do not arrive at a version of Theorem 12 for which also the converse holds.

5 Conclusions and further research

We generalized the well-known notions of reduction orders and monotone algebras for ordinary termination of rewriting to similar notions for context-sensitive termination. We presented a transformation by which context-sensitive termination of a given TRS follows from termination of a transformed TRS, by which well-known techniques for proving termination can be applied for proving context-sensitive termination. Possible directions for future research include:

- Directly defining μ-reduction orders in the style of recursive path order, by which automatic proofs of context-sensitive termination can be given without an intermediate transformation.

- Optimizing the transformation. For instance, R consisting of the rule $f(x) \to g(h(f(x)))$ is μ-terminating for $\mu(f) = \mu(h) = \{1\}$, $\mu(g) = \emptyset$ while $\Psi(R, \mu)$ contains the non-terminating rule $f(x) \to g(\overline{h}(f(x)))$. We also developed another correct transformation (not described in this paper) by which the transformed system contains the rule $f(x) \to g(\overline{h}(\overline{f}(x)))$ and is proved to be terminating by recursive path order. Although this looks more natural and powerful than the transformation described in this paper, it is not. In particular, for less artificial examples containing if-then-else-constructions or selection in infinite lists, standard techniques fail to prove termination of the alternative transformed system. Moreover, this alternative tranformation still allows μ-terminating TRSs for which the tranformed version is non-terminating.

- Treating more involved examples like the factorial example that we mentioned in the introduction, or the sieve of Eratosthenes.

References

[1] Arts, T., and Giesl, J. Termination of constructor systems. In *Proceedings of the 7th Conference on Rewriting Techniques and Applications* (1996), vol. 1103 of *Lecture Notes in Computer Science*, Springer, pp. 63–77.

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

[3] Ben-Cherifa, A., and Lescanne, P. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computing Programming 9*, 2 (1987), 137–159.

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

[5] Lucas, S. Fundamentals of context-sensitive rewriting. In *Proceedings of the 22nd Seminar on Current Trends in Theory and Practice of Informatics (SOFSEM95)* (1995), J. S. M. Bartosek and J. Wiedermann, Eds., vol. 1012 of *Lecture Notes in Computer Science*, Springer, pp. 405 – 412.

[6] Lucas, S. Context-sensitive computations in confluent programs. In *Proceedings of the 8th International Symposium on Programming Languages: Implementations, Logics and Programs (PLILP96)* (1996), H. Kuchen and S. D. Swierstra, Eds., vol. 1140 of *Lecture Notes in Computer Science*, Springer, pp. 408–422.

[7] Lucas, S. Termination of context-sensitive rewriting by rewriting. In *Proceedings of the 23rd International Colloquium on Automata, Languages and Programming (ICALP96)* (1996), F. Meyer auf der Heide and B. Monien, Eds., vol. 1099 of *Lecture Notes in Computer Science*, Springer, pp. 122 – 133.

[8] Steinbach, J. Simplification orderings: history of results. *Fundamenta Informaticae 24* (1995), 47–87.

[9] Zantema, H. Termination of term rewriting: interpretation and type elimination. *Journal of Symbolic Computation 17* (1994), 23–50.

[10] Zantema, H. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae 24* (1995), 89–105.