Termination of term rewriting by interpretation

H. Zantema

RUU-CS-92-14
April 1992

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 interpretation

H. Zantema

Technical Report RUU-CS-92-14
April 1992

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

ISSN: 0924-3275

Termination of term rewriting by interpretation

H. Zantema
Utrecht University, Department of Computer Science
P.O. box 80.089
3508 TB Utrecht
The Netherlands

Abstract

We investigate how to prove termination of term rewriting systems by interpretation of terms. This can be considered as a generalization of polynomial interpretations. A classification of types of termination is proposed built on properties in the semantic level. A transformation on term rewriting systems eliminating distributive rules is introduced. Using this distribution elimination a new termination proof of SUBST from [9] is given.

1 Introduction

One of the main problems in the theory of term rewriting systems is the detection of termination: for a fixed system of rewrite rules, detect whether there exist infinite rewrite chains or not. In general this problem is undecidable ([12, 3]). However, there are several methods for deciding termination that are successful for many special cases. Roughly these methods can be divided into two main types: *syntactical* methods and *semantical* methods. In a syntactical method terms are ordered by a careful analysis of the term structure. A well-known representative of this type is the *recursive path order* ([5]). All of these orderings are simplification orderings, i.e., a term is always greater than its proper subterms. An overview and comparison of simplification orderings is given in [22].

In a semantical method terms are interpreted in some well-known well-founded ordered set in such a way that each rewrite chain will map to a descending chain, and hence will terminate. Until now most semantical methods have focussed on choosing the natural numbers as the well-founded ordered set. The method of *polynomial interpretations* ([17, 1]) can be seen as a particular case of a semantical method on natural numbers. In this paper we introduce the notion of a *monotone algebra* as the natural concept for semantical methods. Though we focus on 'pure' term rewriting systems, the ideas are easily extended to conditional TRS, typed TRS and TRS modulo equations. We propose a classification of types of termination based upon the types of orderings of the underlying monotone algebras. A lot of remarks and examples are not claimed to be new but are included for completeness and for illustrating the setting of monotone algebras.

We present a transformation of term rewriting systems eliminating a particular operation symbol. Using the framework of monotone algebras we prove that under some restrictions termination of the original system follows from termination of the eliminated system. Since in this construction distributive rules are removed completely it is called ***distribution elimination.***

As an application of monotone algebras and distribution elimination we give a new termination proof for the systems studied in [9] and [2]. Our proof is simpler than the existing proofs and gives a stronger result: we prove that the system is even simply terminating.

A survey of the theory of term rewriting systems can be found in [6]. Overviews of existing techniques for termination detection of term rewriting systems can be found in [5, 22]. In the literature termination is also called *strong normalization.*

2 Term rewriting and termination

First we give some standard terminology. Let $\mathcal{F}$ be a set of operation symbols, each having a fixed arity ≥ 0, and let $\mathcal{X}$ be a set of variables. Let $\mathcal{T}(\mathcal{F}, \mathcal{X})$ be the set of terms over $\mathcal{F}$ and $\mathcal{X}$.

An *term rewriting system* (TRS) is defined to be a set $R \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X}) \times \mathcal{T}(\mathcal{F}, \mathcal{X})$. Elements (l, r) of R are called *rules* and are often written as $l \to r$. The *reduction relation* of a TRS R is the relation $\to_R$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ inductively defined by

- $l^\sigma \to_R r^\sigma$ for every $(l, r) \in R$ and every substitution σ;
- $f(t_1, \ldots, t_n) \to_R f(t_1, \ldots, t'_k, \ldots, t_n)$ (only t_k replaced by t'_k) for every $f \in \mathcal{F}$ with arity n and all terms $t_1, \ldots, t_n$ and t'_k with $t_k \to_R t'_k$.

A TRS R is called *terminating* (or strongly normalizing or noetherian) if there exist no infinite reductions of the reduction relation $\to_R$.

A partial order on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is called a *reduction order* if it is well-founded and closed under substitution and context. We say that a reduction order $>$ *normalizes* a TRS if $l > r$ for each rewrite rule $l \to r$. This terminology is motivated by the following proposition.

Proposition 1 *A TRS is terminating if and only if it is normalized by a reduction order.*

Proof: Assume the TRS is normalized by a reduction order. Then any infinite reduction chain is an infinite descending chain. Since a reduction order is well-founded, such chains do not exist, so the system is terminating.

On the other hand, if the system is terminating then the transitive closure of the rewrite relation satisfies all requirements of a normalizing reduction order. □

3 Monotone algebras

In this paper we consider orderings on terms induced by interpretations. The idea is that each term is interpreted in some well-founded set in such a way that at each rewrite step the corresponding value decreases. Well-foundedness of the set then implies termination of the rewrite system. This idea already appears in [20]. It is convenient not to check decreasing for all (infinitely many) possible rewrite steps, but only for the rewrite rules. As we saw above, this holds if the implied order on terms is a reduction order. We shall see that if the interpretation is an algebra, i.e., it can be defined in a compositional way, and it satisfies some monotonicity condition, then the corresponding order is indeed a reduction order.

The same requirements already emerged in the particular case of polynomial interpretations ([17, 1]). We shall extend this concept in such a way that it covers all types of termination.

We define a *well-founded monotone $\mathcal{F}$-algebra* $(A, >)$ to be an $\mathcal{F}$-algebra A for which the underlying set is provided with a well-founded 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

$$f_A(a_1, \ldots, a_n) > f_A(b_1, \ldots, b_n).$$

Let $(A, >)$ be a well-founded monotone $\mathcal{F}$-algebra. Let $A^{\mathcal{X}} = \{\alpha : \mathcal{X} \to A\}$. We define

$$\phi : \mathcal{T}(\mathcal{F}, \mathcal{X}) \times A^{\mathcal{X}} \to A$$

inductively by

$$\begin{aligned} \phi(x, \alpha) &= x^\alpha, \\ \phi(f(t_1, \ldots, t_n), \alpha) &= f_A(\phi(t_1, \alpha), \ldots, \phi(t_n, \alpha)) \end{aligned}$$

for $x \in \mathcal{X}, \alpha : \mathcal{X} \to A, f \in \mathcal{F}, t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If confusion is possible to which algebra the function ϕ corresponds we write ϕ_A instead of ϕ. This function induces a relation $>_A$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ as follows:

$$t >_A t' \iff (\forall \alpha \in A^{\mathcal{X}} : \phi(t, \alpha) > \phi(t', \alpha)).$$

We shall prove that $>_A$ is a reduction order; first we need a lemma.

Lemma 2 *Let $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ be any substitution and let $\alpha : \mathcal{X} \to A$. Define $\beta : \mathcal{X} \to A$ by $\beta(x) = \phi(x^\sigma, \alpha)$ for $x \in \mathcal{X}$. Then*

$$\phi(t^\sigma, \alpha) = \phi(t, \beta)$$

for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$.

Proof: Induction on the structure of t. □

Proposition 3 *Let $(A, >)$ be a non-empty well-founded monotone $\mathcal{F}$-algebra. Then $>_A$ is a reduction order on $\mathcal{T}(\mathcal{F}, \mathcal{X})$.*

Proof: Irreflexivity, transitivity and well-foundedness of $>_A$ follow from the corresponding properties of $>$. We still have to prove the closedness under substitution and context of $>_A$.

Let $t >_A t'$ for $t, t' \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and let $\sigma : X \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ be any substitution. Let $\alpha : \mathcal{X} \to A$. From lemma 2 we obtain

$$\phi(t^\sigma, \alpha) \;=\; \phi(t, \beta) \;>\; \phi(t', \beta) \;=\; \phi(t'^\sigma, \alpha).$$

The key point here is that β does not depend on t. This holds for all $\alpha : \mathcal{X} \to A$, so $t^\sigma >_A t'^\sigma$. Hence $>_A$ is closed under substitution.

For proving closedness under context let $t >_A t'$ for $t, t' \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, and let $f \in \mathcal{F}$. Since $t >_A t'$ we have $\phi(t, \alpha) > \phi(t', \alpha)$ for all $\alpha : \mathcal{X} \to A$. Applying the monotonicity condition of f_A we obtain

$$\phi(f(\ldots, t, \ldots), \alpha) = f_A(\ldots, \phi(t, \alpha), \ldots) > f_A(\ldots, \phi(t', \alpha), \ldots) = \phi(f(\ldots, t', \ldots), \alpha).$$

This holds for all $\alpha : \mathcal{X} \to A$, so

$$f(\ldots, t, \ldots) >_A f(\ldots, t', \ldots),$$

which we had to prove. □

We say that a non-empty well-founded monotone algebra $(A, >)$ *normalizes* a TRS if the corresponding reduction order $>_A$ normalizes the TRS. This terminology is motivated by the following proposition.

Proposition 4 *A TRS is terminating if and only if it is normalized by a non-empty well-founded monotone algebra.*

Proof: Assume the TRS is normalized by a non-empty well-founded monotone algebra. Then it is normalized by a reduction order. From proposition 1 we conclude that it is terminating.

On the other hand, assume the system is terminating. Define $A = \mathcal{T}(\mathcal{F}, \mathcal{X})$, and define $>$ to be the transitive closure of the rewrite relation. One easily verifies that $(A, >)$ is a non-empty well-founded 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

$$\phi(t, \alpha) \;=\; t^\alpha$$

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

$$\phi(l, \alpha) \;=\; l^\alpha \;>\; r^\alpha \;=\; \phi(r, \alpha).$$

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

The way of proving termination of a TRS is now as follows: choose a well-founded poset A, define for each operation symbol a corresponding operation that is strictly monotone in all of its coordinates, and prove that $\phi(l, \alpha) >_A \phi(r, \alpha)$ for all rewrite rules $l \to r$ and all $\alpha : \mathcal{X} \rightarrow A$. Then according to the above proposition the TRS is terminating. Often we choose A to be $\mathbb{N}$, which is defined to be the set of strictly positive integers. For example, the system

$$f(f(x,y),z) \rightarrow f(x,f(y,z))$$

is proven to be terminating as follows. Choose $(A,>) = (\mathbb{N},>)$ and $f_A(x,y) = 2x+y$. Clearly f_A is strictly monotone in both coordinates, and

$$f_A(f_A(x,y),z) = 4x+2y+z > 2x+2y+z = f_A(x,f_A(y,z))$$

for all $x,y,z \in A$. Hence $f(f(x,y),z) >_A f(x,f(y,z))$, proving termination.

By translation the case of

$$(A,>) = (\{n \in \mathbb{N} \mid n > N\} >)$$

for some natural number N is equivalent to $(A,>) = (\mathbb{N},>)$. If the operations in this algebra are polynomials, this corresponds to polynomial interpretations.

Some examples

Next we give some examples in which we choose A to be two copies of the naturals: define $A = \{0,1\} \times \mathbb{N}$ and

$$(a,n) > (b,m) \iff a = b \wedge n > m.$$

Note that $(A,>)$ is well-founded poset which is not total. All three examples will be referred to later in this paper; none of the three can be proved tot be terminating using a total well-founded monotone algebra..

1. Consider the TRS consisting of the rule:

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

   Define

$$f_A(0,n) = (0,n+1), \quad f_A(1,n) = (0,n),$$
$$g_A(0,n) = g_A(1,n) = (1,n)$$

   for all $n \in \mathbb{N}$. Both f_A and g_A are strictly monotone, while

$$\begin{array}{rclclcl} f_A(f_A(0,n)) & = & (0,n+2) & > & (0,n+1) & = & f_A(g_A(f_A(0,n))), \\ f_A(f_A(1,n)) & = & (0,n+1) & > & (0,n) & = & f_A(g_A(f_A(1,n))) \end{array}$$

   for all $n \in \mathbb{N}$, proving termination.

2. Consider the TRS with the two rules:

$$\begin{array}{rcl} f(g(x)) & \rightarrow & f(f(x)), \\ g(f(x)) & \rightarrow & g(g(x)). \end{array}$$

   Define

$$f_A(0,n) = (1,2n), \quad f_A(1,n) = (1,n+1),$$

$$g_A(0,n) = (0, n+1), \quad g_A(1,n) = (0, 2n).$$

Both f_A and g_A are strictly monotone, while

$$\begin{array}{rclclcl} f_A(g_A(0,n)) & = & (1,2n+2) & > & (1,2n+1) & = & f_A(f_A(0,n)), \\ f_A(g_A(1,n)) & = & (1,4n) & > & (1,n+2) & = & f_A(f_A(1,n)), \\ g_A(f_A(0,n)) & = & (0,4n) & > & (0,n+2) & = & g_A(g_A(0,n)), \\ g_A(f_A(1,n)) & = & (0,2n+2) & > & (0,2n+1) & = & g_A(g_A(1,n)). \end{array}$$

for all $n \in \mathbb{N}$, proving termination.

3. Let the TRS consist of the rule:

$$f(0,1,x) \rightarrow f(x,x,x).$$

Define

$$0_A = (0,1), \quad 1_A = (1,1),$$

$$f_A((a,n),(b,m),(c,k)) = \begin{cases} (0, n+m+k) & \text{if } a = b \\ (0, n+m+3k) & \text{if } a \neq b \end{cases}$$

The function f_A is strictly monotone in all three coordinates. For all $(a,n) \in A$ we have

$$f_A(0_A, 1_A, (a,n)) = (0, 3n+2) > (0, 3n) = f_A((a,n),(a,n),(a,n)),$$

proving termination.

If no confusion is possible, we shall sometimes remove subscripts, so we write $f, g, \ldots$ instead of $f_A, g_A, \ldots$.

4 Simple termination

If $\mathcal{F}$ is finite it is sometimes convenient to replace the well-foundedness condition in the definition of a well-founded monotone algebra by a simplicity condition as follows. A *simple monotone $\mathcal{F}$-algebra* $(A, >)$ is defined to be an $\mathcal{F}$-algebra A for which the underlying set is provided with a partial order $>$ such that each algebra operation is strictly monotone in all of its coordinates, and

$$f_A(a_1, \ldots, a_n) \geq a_i$$

for each $f \in \mathcal{F}$, $a_1, \ldots, a_n \in A$, and $i \in \{1, \ldots, n\}$. The corresponding reduction order $>_A$ is called a *simplification ordering*. This definition coincides with that in [6]. These definitions are motivated by the following two propositions.

Proposition 5 *Let $\mathcal{F}$ be finite and let $(A, >)$ be a simple monotone $\mathcal{F}$-algebra. Let A' be the smallest subalgebra of A, i.e., A' is the homomorphic image of the ground terms. Then $(A', >)$ is a well-founded monotone $\mathcal{F}$-algebra.*

Proof: The only property to prove is well-foundedness. Assume the restriction of $>$ to A' is not well-founded. Then there is an infinite chain

$$h(t_0) > h(t_1) > h(t_2) > h(t_3) > \cdots,$$

where h is the homomorphism from ground terms to A. The key argument is Higman's lemma ([10]), which is a special case of Kruskal's tree theorem ([14]); the relevance for termination of term rewriting systems is explained in [6]. Higman's lemma states that there is some $i < j$ such that t_i can be homeomorphically embedded in t_j. Since $(A, >)$ is a simple monotone algebra and h is a homomorphism, we conclude that $h(t_j) \geq h(t_i)$, contradicting irreflexivity and transitivity of $>$. □

Proposition 6 *Let $\mathcal{F}$ be finite and let $(A, >)$ be a non-empty simple monotone $\mathcal{F}$-algebra. Let R be a TRS such that $l >_A r$ for all rewrite rules $l \to r$ of R. Then R is terminating.*

Proof: Apply proposition 5: A' is a well-founded monotone algebra normalizing R. In the case that $\mathcal{F}$ does not contain constants, add one dummy constant symbol forcing $A' \neq \emptyset$. □

For a set $\mathcal{F}$ of operation symbols we define $Emb(\mathcal{F})$ to be the TRS consisting of all the rules

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

with $f \in \mathcal{F}$ and $i \in \{1, \ldots, n\}$.

Proposition 7 *Let R be a TRS over a set $\mathcal{F}$ of operation symbols. Then the following assertions are equivalent:*

(1) *R is simply terminating;*
(2) *$R \cup Emb(\mathcal{F})$ is simply terminating;*
(3) *$R \cup Emb(\mathcal{F})$ is terminating.*

Proof: The implication (2) $\Rightarrow$ (1) is trivial. For proving (1) $\Rightarrow$ (2) let $(A, >)$ be a simple monotone $\mathcal{F}$-algebra normalizing R. Since we allow equality in the definition of simplicity, we have to modify A in order to normalize $R \cup Emb(\mathcal{F})$. Choose

$$B = A \times \mathbb{N}$$

having the lexicographic order

$$(a, k) > (a', k') \iff a > a' \vee (a = a' \wedge k > k').$$

Define

$$f_B((a_1, k_1), \ldots, (a_n, k_n)) = (f_A(a_1, \ldots, a_n), 1 + \sum_{i=1}^{n} k_i).$$

Now $(B, >)$ is a simple monotone algebra normalizing both R and $Emb(\mathcal{F})$, proving (2).

The implication (2) $\Rightarrow$ (3) is trivial. Finally, assume that (3) holds. Then according to proposition 4 there is a non-empty well-founded monotone $\mathcal{F}$-algebra $(A, >)$ normalizing $R \cup Emb(\mathcal{F})$. Since it normalizes $Emb(\mathcal{F})$ it is also a simple monotone $\mathcal{F}$-algebra. This implies (2). □

5 The hierarchy

Let $(A, >)$ be a monotone algebra. Depending on its properties we propose a hierarchy of types of termination. If $A = \mathbb{N}$ and $>$ is the ordinary order on $\mathbb{N}$ and f_A is a polynomial for all $f \in \mathcal{F}$, we speak about *polynomial termination*. If $A = \mathbb{N}$ and $>$ is the ordinary order on $\mathbb{N}$, we speak about *ω-termination*. In these cases we may have $\{n \in \mathbb{N} \mid n > N\}$ instead of $\mathbb{N}$, which gives equivalent definitions due to linear transformation. An implementation based on polynomial termination is described in [1]; a recent extension to elementary functions in which also exponents may occur is given in [19].

If the order $>$ on A is total and well-founded, we speak about *total termination*. If $(A, >)$ is a simple monotone algebra, we speak about *simple termination*.

The following implications hold, and in this section we prove that none of the implications holds in the reverse direction:

polynomial termination
$\Longrightarrow$ ω-termination
$\Longrightarrow$ total termination
$\Longrightarrow$ simple termination
$\Longrightarrow$ termination.

The only non-trivial implication is the implication of simple termination from total termination. This follows immediately from the following proposition.

Proposition 8 *Let $(A, >)$ be a well-founded monotone $\mathcal{F}$-algebra for which the order $>$ is total on A. Then $(A, >)$ is a simple monotone $\mathcal{F}$-algebra.*

Proof: Assume it is not simple. Then there exist $f \in \mathcal{F}, a_1, \ldots, a_n \in A$ and $i \in \{1, \ldots, n\}$ such that

$$a_i > f_A(a_1, \ldots, a_n).$$

Define $g : A \to A$ by $g(x) = f_A(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_n)$, then g is strictly monotone. We obtain an infinite chain

$$a_i > g(a_i) > g(g(a_i)) > g(g(g(a_i))) > \cdots,$$

contradicting the well-foundedness of $(A, >)$. $\Box$

To prove that none of the implications holds in the reverse direction we prove properties of particular examples.

Proposition 9 *The TRS*

$$a(f(x), y) \quad \to \quad f(a(x, a(x, y)))$$

is ω-terminating but not polynomially terminating.

Proof: Define $a(x, y) = y^x$ and $f(x) = x^3$. Then

$$a(f(x), y) = y^{x^3} > y^{3x^2} = f(a(x, a(x, y)))$$

for all $x, y > 3$, so the system is ω-terminating.

Assume the system is polynomially terminating. Then there exist polynomials a and f, strictly monotone in all coordinates, such that

$$a(f(x), y) > f(a(x, a(x, y))) \tag{1}$$

for all $x, y \in \mathbb{N}$. There exist polynomials p, q, r such that

$$a(x, y) = p(x) + q(y) + xy * r(x, y).$$

If $r \neq 0$ then the degree in x of the left hand side of (1) is smaller than the degree in x of the right hand side of (1), contradiction, so $r = 0$. Now (1) yields

$$p(f(x)) + q(y) > f(p(x) + q(p(x) + q(y))). \tag{2}$$

Due to monotonicity f, p and q all have degree ≥ 1. Considering the degree in y now yields that both f and q are linear. Due to monotonicity the leading coefficients of f and q are both ≥ 1, due to (2) they are not > 1. So

$$f(x) = x + c \quad \text{and} \quad q(x) = x + d$$

for constants c and d. Now (2) yields

$$p(x + c) > 2p(x) + d + c,$$

which is impossible considering degree and leading coefficient. □

Another approach for proving the non-equivalence of polynomial and ω-termination is the following. For a polynomially terminating term rewriting system R on finite $\mathcal{F}$ it is easy to prove ([18]) that there is a constant C only depending on R such that the length of a reduction of a term consisting of n operation symbols is bounded by $\exp(\exp(Cn))$. For ω-terminating term rewriting systems this property does not hold, as is shown by the next system due to V.C.S. Meeussen:

$$\begin{array}{rcl} b(a(x)) & \to & a(a(b(x))) \\ c(b(x)) & \to & b(b(c(x))) \\ c(a(x)) & \to & a(x) \\ d(c(x)) & \to & c(c(d(x))) \\ d(b(x)) & \to & b(x), \end{array}$$

having the reduction

$$d^n(c(b(a(x)))) \to^* c^{2^n}(d^n(b(a(x)))) \to^* c^{2^n}(b(a(x))) \to^*$$

$$b^{2^{2^n}}(c^{2^n}(a(x))) \to^* b^{2^{2^n}}(a(x)) \to^* a^{2^{2^{2^n}}}(b^{2^{2^n}}(x))$$

of length strictly exceeding the above bound. Hence this system is not polynomially terminating; ω-termination is easily shown by choosing $a(x) = x + 1, b(x) = 3x, c(x) = x^3$ and $d(x) = 2^x$. In [18, 11] an example is given that the bound of $\exp(\exp(Cn))$ is sharp for polynomial termination. A smaller example with the same behaviour is given by the first three rules of the above example; outermost reduction of $c^n(b(a(x)))$ gives a reduction length exceeding $\exp(\exp(Cn))$ for some $C > 0$.

Proposition 10 *The term rewriting system*

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

is totally terminating but not ω-terminating.

Proof: For proving total termination choose $A = \mathbb{N} \times \mathbb{N}$ with the lexicographic order

$$(n, n') > (m, m') \iff n > m \vee (n = m \wedge n' > m').$$

Further define

$$f(n, n') = (n, n + n') \quad \text{and} \quad g(n, n') = (2n + 1, n').$$

Monotonicity of f and g is easily verified; for the monotonicity of f it is essential to choose this lexicographic order and not the reversed one. Now we have

$$f(g(n, n')) = (2n + 1, 2n + n' + 1) > (2n + 1, 2n + n') = g(f(f(n, n')))$$

for all $(n, n') \in A$, so the system is totally terminating.

On the other hand assume that the system is ω-terminating. Then there exist strictly monotonic $f, g : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$\forall n \in \mathbb{N} : f(g(n)) > g(f(f(n))). \tag{3}$$

Using monotonicity one easily proves by induction on n that

$$\forall n \in \mathbb{N} : f(n) \geq n \wedge g(n) \geq n. \tag{4}$$

Since f is monotonic we have

$$\forall n, m \in \mathbb{N} : (f(n) > f(m) \Rightarrow n > m). \tag{5}$$

According to (4) the assertion $g(n) \geq f^k(n)$ holds for $k = 0$ and all $n \in \mathbb{N}$. Assume that the assertion $g(n) \geq f^k(n)$ holds for some $k \in \mathbb{N}$ and all $n \in \mathbb{N}$. Then

$$\forall n \in \mathbb{N} : f(g(n)) > g(f(f(n))) \geq f^k(f(f(n))) = f(f^{k+1}(n));$$

from (5) we conclude that

$$\forall n \in \mathbb{N} : g(n) > f^{k+1}(n).$$

Now we have proved by induction on k that

$$\forall n, k \in \mathbb{N} : g(n) \geq f^k(n). \tag{6}$$

If there is some $n \in \mathbb{N}$ with $f(n) > n$ then

$$n < f(n) < f(f(n)) < f(f(f(n))) < \cdots$$

contradicting (6), otherwise $f(n) = n$ for all $n \in \mathbb{N}$, contradicting (3). □

After a question on the electronic newsnet, the proof of the impossibility of (3) has been given independently by several people.

Proposition 11 *The term rewriting system*

$$\begin{array}{rcl} f(a) & \rightarrow & f(b) \\ g(b) & \rightarrow & g(a) \end{array}$$

is simply terminating but not totally terminating.

Proof: Choose $A = \mathcal{T}(\mathcal{F})$ and define

$$t > t' \iff \psi(t) > \psi(t'),$$

where $\psi : \mathcal{T}(\mathcal{F}) \rightarrow \mathbb{N}$ is defined by

$$\psi(a) = \psi(b) = 0,$$

$$\psi(f(a)) = \psi(g(b)) = 3,$$

$$\psi(f(t)) = \psi(t) + 2 \quad \text{for all} \quad t \neq a,$$

$$\psi(g(t)) = \psi(t) + 2 \quad \text{for all} \quad t \neq b.$$

One easily verifies that $(A, >)$ is a simple monotone algebra, proving simple termination.

Next assume that the system is totally terminating. Then it is normalized by a well-founded monotone algebra $(A, >)$ such that $>$ is a total order on A. Totality implies $a_A \geq b_A \vee b_A \geq a_A$. Since f_A and g_A are monotone this implies $g_A(a_A) \geq g_A(b_A) \vee f_A(b_A) \geq f_A(a_A)$, contradicting the assumption that $(A, >)$ normalizes the TRS. □

The same example is given in [5, 6]. Another way of proving simple termination can be given by applying proposition 7. Example 2 of section 3 is also simply terminating and not totally terminating.

Proposition 12 *The term rewriting system*

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

is terminating but not simply terminating.

Proof: A termination proof was given in example 1 of section 3. Assume it is simply terminating. According to proposition 7 then the system extended by the rules $f(x) \rightarrow x$ and $g(x) \rightarrow x$ is terminating, which is not true since there is an infinite cyclic reduction

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

□

One of the common tools for proving termination of term rewriting systems is the recursive path order with status ([13, 4]). It can be shown that every TRS proved terminating using this ordering is totally terminating as follows. Consider the equivalence relation on terms generated by permuting arguments of operation symbols of multiset status. Now the set of terms up to this equivalence is a total monotone algebra in a natural way.

6 Basic constructions

In order to be able find termination proofs for term rewriting systems by monotone algebras, it is useful to investigate some basic constructions. In particular for some basic well-founded sets we should like to have lists of unary and binary functions that are monotone in all coordinates. By *monotone* we shall always mean *strictly monotone* in all coordinates. In this section we restrict to total orders. The simplest unary function satisfying the monotonicity condition is the identity.

If a is a binary monotone function, f and g are unary monotone functions and c is some constant, then

$$x \mapsto a(x,x), \quad x \mapsto a(x,c), \quad x \mapsto a(c,x), \quad f \circ g$$

are unary monotone functions, and

$$(x,y) \mapsto a(f(x),g(y)), \quad (x,y) \mapsto a(y,x)$$

are binary monotone functions.

Natural numbers

Basic unary functions on the natural numbers are: addition by a non-negative constant (often: 1), multiplication by a positive constant, $x \mapsto x^c$ for a positive constant c, and $x \mapsto c^x$ for a constant $c > 1$.

Basic binary functions on the natural numbers are: addition, multiplication, and $(x,y) \mapsto x^y$. Multiplication is only monotone for positive natural numbers, exponentiation only for numbers > 1. Note that some monotone polynomials, like $x \mapsto x^2 - x + 1$ can not be obtained as compositions of these basic monotone functions.

Lexicographic order

If $(A, >)$ and $(B, >)$ are well-founded, then so is $A \times B$ with the lexicographic order

$$(a,b) > (a',b') \iff a > a' \vee (a = a' \wedge b > b').$$

The lexicographic order on $A \times B$ is total if and only if the orders $(A, >)$ and $(B, >)$ are both total.

Basic unary monotone functions on $A \times B$ are

$$(a,b) \mapsto (\chi(a), \psi(a,b)),$$

where χ is a monotone function on A and $x \mapsto \psi(a,x)$ is a monotone function on B for each $a \in A$; monotonicity in the first coordinate of ψ is not required.

Basic binary monotone functions one $A \times B$ are

$$((a,b),(a',b')) \mapsto (\chi(a,a'), \psi(a,a',b,b')),$$

where χ is a binary monotone function on A and $(x,y) \mapsto \psi(a,a',x,y)$ is a binary monotone function on B for each $a, a' \in A$; monotonicity in the first and second coordinate of ψ is not required.

Multisets

For any set A define

$$M(A) = \{X : A \to \mathbb{N} | \#\{a \in A | X(a) \neq 0\} < \infty\},$$

i.e., $M(A)$ is the set of finite multisets over A. If $>$ is an order on A then an order on $M(A)$ can be defined as follows:

$$X > Y \iff X \neq Y \wedge (\forall a \in A : X(a) \geq Y(a) \vee (\exists a' \in A : a' > a \wedge X(a') > Y(a'));$$

this order is the same as in [7]. The order $(M(A), >)$ is total if and only if $(A, >)$ is total; the order $(M(A), >)$ is well-founded if and only if $(A, >)$ is well-founded. This construction corresponds to exponentiation in ordinal arithmetic as discussed in [15]: if $(A, >)$ corresponds to the ordinal α, then $(M(A), >)$ corresponds to the ordinal ω^α.

A basic binary monotone function is the multiset union, defined by

$$(X \cup Y)(a) = X(a) + Y(a).$$

Multiset union is associative and commutative. For every $a \in A$ the singleton $[a]$ is defined by $[a](a) = 1$ and $[a](x) = 0$ for $x \neq a$. Every non-empty finite multiset can be obtained as a finite multiset union of singletons. The smallest multiset is the empty multiset $[]$, it is defined by $[](a) = 0$ for all $a \in A$.

Unary monotone functions are obtained by taking union by a constant or union by itself. Further if $f : A \to A$ is monotone, then so is $f^* : M(A) \to M(A)$, where

$$f^*([]) = [], \quad f^*([a]) = [f(a)], \quad f^*(X \cup Y) = f^*(X) \cup f^*(Y).$$

Not only unary monotone functions can be lifted to monotone functions on multisets. Write $M'(A) = M(A) \setminus \{[]\}$, the set of finite non-empty multisets over A. For every $f : A^k \to A$ the function $\bar{f} : (M'(A))^k \to M'(A)$ is defined as follows:

$$\bar{f}([a_1], [a_2], \ldots [a_k]) = [f(a_1, a_2, \ldots, a_k)],$$

$$\bar{f}(X_1, \ldots X_{i-1}, Y \cup Z, X_{i+1}, \ldots X_k) =$$

$$\bar{f}(X_1, \ldots X_{i-1}, Y, X_{i+1}, \ldots X_k) \cup \bar{f}(X_1, \ldots X_{i-1}, Z, X_{i+1}, \ldots X_k).$$

Intuitively: to compute $\bar{f}(X_1, \ldots X_k)$, apply f to all possible choices of elements of $X_1, \ldots X_k$, and collect all results in one multiset. One easily shows that if f is monotone in every coordinate then also $\bar{f}$ is monotone in every coordinate; here for $k > 1$ it is essential to restrict to non-empty multisets. For unary f the function $\bar{f}$ is equal to the restriction of f^* to non-empty multisets; for a constant c we have $\bar{c} = [c]$.

In the next section this lifting of f to $\bar{f}$ plays an essential role.

7 Distribution elimination

In this section we introduce a transformation of TRS's in which a particular operation symbol is eliminated, and prove that if the eliminated TRS is terminating then the original TRS is also terminating.

Let a be any fixed operation symbol of arity $n \geq 1$. A rewrite rule is called a *distribution rule* for a if it can be written as

$$C[a(x_1, \ldots, x_n)] \to a(C[x_1], \ldots, C[x_n])$$

for some non-trivial context $C[]$ in which the symbol a does not occur. For example,

$$b(z, f(a(x,y)) \to a(b(z, f(x)), b(z, f(y)))$$

is a distribution rule for a. Problems with distribution rules have been recognized before; for example in [8] a particular ordering for proving termination of AC rewriting systems is introduced for systems containing distribution rules.

Write $\mathcal{P}$ for powerset. The function $E_a : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathcal{P}(\mathcal{T}(\mathcal{F}, \mathcal{X}))$ is defined inductively as follows:

$$\begin{array}{lcll} E_a(x) & = & \{x\} & \text{for all } x \in \mathcal{X}, \\ E_a(f(t_1, \ldots, t_k)) & = & \{f(u_1, \ldots, u_k) \mid \forall i : u_i \in E_a(t_i)\} & \text{for all } f \in \mathcal{F}, f \neq a \\ E_a(a(t_1, \ldots, t_n)) & = & \bigcup_{i=1}^n E_a(t_i). & \end{array}$$

Let R be a TRS for which each rule is either a distribution rule for a or a rule in which a does not occur in the left hand side. Then the TRS $E_a(R)$ is defined by

$$E_a(R) = \{l \to u \mid l \to r \text{ is a non-distribution rule of } R \text{ for } a \text{ and } u \in E_a(r)\}.$$

For example, if R is defined by

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

then $E_a(R)$ consists only of the rule $f(f(x)) \to f(g(f(x)))$. This system is known to be terminating; the next proposition states that we can conclude that also R is terminating. As usual a term is defined to be linear if no variable occurs more than once, and a TRS is defined to be right-linear if for every rule the right hand side is linear.

Theorem 13 *Let R be a TRS for which each rule is either a distribution rule for a or a rule in which a does not occur in the left hand side. Then*

- *if $E_a(R)$ is totally terminating then R is totally terminating;*
- *if $E_a(R)$ is simply terminating and right-linear then R is simply terminating;*
- *if $E_a(R)$ is terminating and right-linear then R is terminating.*

Before giving the proof we give examples showing that the converse does not hold and the right-linearity requirement is essential. First choose R to be $f(f(x)) \to f(a(f(x)))$. Then R is terminating and satisfies the conditions (there are no distribution rules), while $E_a(R)$ consists of $f(f(x)) \to f(f(x))$ which is not terminating.

Next choose R to be:

$$\begin{array}{rcl} f(0,1,x) & \to & f(x,x,x) \\ f(a(x,y),z,w) & \to & a(f(x,z,w),f(y,z,w)) \\ f(x,a(y,z),w) & \to & a(f(x,y,w),f(x,z,w)) \end{array}$$

The second and third rule are distribution rules for a, the first rule does not contain a. So $E_a(R)$ consists only of the first rule, which we proved to be terminating in example 3 of section 3. However, in R we have the reduction

$$\begin{array}{rcl} f(0,1,a(0,1)) & \to & f(a(0,1),a(0,1),a(0,1)) \\ & \to & a(f(0,a(0,1),a(0,1)),f(1,a(0,1),a(0,1))) \\ & \to & a(a(f(0,0,a(0,1)),\underbrace{f(0,1,a(0,1))}),f(1,a(0,1),a(0,1))) \end{array}$$

in which the starting term occurs as a subterm. This can be expanded to an infinite reduction. The original idea of this example is due to Toyama ([23]) in the context of direct sum modularity; it is known that this non-modular behaviour only occurs in a system that has both collapsing and duplicating rules ([21]) and is not simply terminating ([16]).

The requirement of right-linearity in the third assertion of the theorem may not be weakened to absence of duplicating rules: the system

$$\begin{array}{rcl} f(0,1,x,x) & \to & f(x,x,a(0,1),a(0,1)) \\ f(a(x,y),z,v,w) & \to & a(f(x,z,v,w),f(y,z,v,w)) \\ f(x,a(y,z),v,w) & \to & a(f(x,y,v,w),f(x,z,v,w)) \end{array}$$

allows a similar inifinite reduction, while the eliminated version

$$\begin{array}{rcl} f(0,1,x,x) & \to & f(x,x,0,0) \\ f(0,1,x,x) & \to & f(x,x,0,1) \\ f(0,1,x,x) & \to & f(x,x,1,0) \\ f(0,1,x,x) & \to & f(x,x,1,1) \end{array}$$

has no duplicating rules and is terminating.

We do not know whether the right-linearity requirement may be removed in the second assertion of the theorem.

Now we prove theorem 13.

Proof: Let $(A,>)$ be a well-founded monotone algebra for $E_a(R)$. Again write $M'(A)$ for the set of finite non-empty multisets over A.

We define the well-founded monotone algebra for R to be

$$B = M'(A) \times \mathbb{N}$$

where $\mathbb{N}$ consists of the strictly positive integers, with the lexicographic order

$$(X,k) > (Y,m) \iff X > Y \vee (X = Y \wedge k > m).$$

As operations we define

$$a_B((X_1,m_1),\ldots,(X_n,m_n)) = (\bigcup_{i=1}^{n} X_i, 1+\sum_{i=1}^{n} m_i)$$

and

$$f_B((X_1,m_1),\ldots,(X_k,m_k)) = (\bar{f}_A(X_1,\ldots,X_k), \prod_{i=1}^{k} m_i^2)$$

for all $f \in \mathcal{F}, f \neq a$. Note that these operations are strictly monotone in all coordinates. Further if $(A,>)$ is a simple monotone algebra then $(B,>)$ is also simple; if $(A,>)$ is total then $(B,>)$ is also total. We have to prove that for every rule of R the left hand side is greater than the right hand side, interpreted in B.

Let $\beta : \mathcal{X} \to B$ be arbitrary, and write $\phi_B(t,\beta) = (\phi_1(t),\phi_2(t)) \in M'(A)\times \mathbb{N}$ for $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, where ϕ_B is from the definition of well-founded monotone algebra. By definition we have

$$\begin{aligned}
\phi_1(x) &= \pi(x^\beta) \quad \text{for } x \in \mathcal{X}, \text{ where } \pi : B \to M'(A) \text{ is the} \\
&\qquad\qquad\qquad \text{projection on the first coordinate} \\
\phi_1(a(t_1,\ldots,t_n)) &= \bigcup_{i=1}^{n} \phi_1(t_i) \\
\phi_1(f(t_1,\ldots,t_k)) &= \bar{f}_A(\phi_1(t_1),\ldots,\phi_1(t_k)) \quad \text{for } f \in \mathcal{F}, f \neq a.
\end{aligned}$$

Let $C[]$ be any non-trivial context in which a does not occur. Since

$$\bar{f}_A(X_1,\ldots X_{i-1},Y\cup Z,X_{i+1},\ldots X_k) =$$
$$\bar{f}_A(X_1,\ldots X_{i-1},Y,X_{i+1},\ldots X_k)\cup \bar{f}_A(X_1,\ldots X_{i-1},Z,X_{i+1},\ldots X_k)$$

for all operation symbols f occurring in $C[]$, we see that

$$\phi_1(C[a(x_1,\ldots,x_n)] = \bigcup_{i=1}^{n} \phi_1(C[x_i]) = \phi_1(a(C[x_1],\ldots,C[x_n])).$$

Since a does not occur in $C[]$, for every term t we have $\phi_2(C[t]) = c * \phi_2(t)^p$ for some $c \geq 1$ and $p > 1$. As a consequence we have

$$\begin{aligned}
\phi_2(C[a(x_1,\ldots,x_n)] &= c*(1+\sum_{i=1}^{n}\phi_2(x_i))^p \\
&> 1+\sum_{i=1}^{n} c*\phi_2(x_i)^p \\
&= \phi_2(a(C[x_1],\ldots,C[x_n])).
\end{aligned}$$

As a consequence, for any distribution rule

$$C[a(x_1,\ldots,x_n)] \to a(C[x_1],\ldots,C[x_n])$$

in R we have

$$C[a(x_1,\ldots,x_n)] >_B a(C[x_1],\ldots,C[x_n]).$$

Every other rule in R is of the shape $l \to r$ in which a does not occur in l. Choose $s \in \phi_1(r)$ arbitrarily. We shall prove that in any case $\phi_1(l)$ contains an element strictly greater than s.

From the construction of E_a we obtain

$$\phi_1(r) = \bigcup_{u \in E_a(r)} \phi_1(u),$$

which is proved by induction on the structure of r. As a consequence, there is some $u \in E_a(r)$ such that $s \in \phi_1(u)$. We shall construct $\alpha : \mathcal{X} \to A$ such that $s \leq \phi_A(u, \alpha)$ and $x^\alpha \in \phi_1(x)$; here we need to distinguish between the cases of right-linearity and totality. For the first case we need to prove the following fact:

> **Fact.** Let t be a linear term in which the symbol a does not occur and let $e \in \phi_1(t)$. Define $\mathcal{X}_t$ to be the set of variables occurring in t. Then there exists $\alpha : \mathcal{X}_t \to A$ such that $x^\alpha \in \phi_1(x)$ for all $x \in \mathcal{X}_t$ and $e = \phi_A(t, \alpha)$.

We prove this fact by induction on the structure of t. For the basis of the induction the term t is a variable x and we define $x^\alpha = e$. For the induction step we have $t = f(t_1, \ldots, t_k)$; then we have

$$e \in \phi_1(t) = \phi_1(f(t_1, \ldots, t_k)) = \bar{f}_A(\phi_1(t_1), \ldots, \phi_1(t_k)).$$

From the definition of $\bar{f}_A$ follows that there are $e_i \in \phi_1(t_i)$ for $i = 1, \ldots, k$ such that $e = f_A(e_1, \ldots, e_k)$. From the induction hypothesis we obtain $\alpha_i : \mathcal{X}_{t_i} \to A$ with $e_i = \phi_A(t_i, \alpha_i)$ for $i = 1, \ldots, k$. Since the term t is linear all $\mathcal{X}_{t_i}$ are disjoint. So there exists $\alpha : \mathcal{X}_t \to A$ such that $x^\alpha = x^{\alpha_i}$ if $x \in \mathcal{X}_{t_i}$. This gives $x^\alpha \in \phi_1(x)$; further we obtain

$$e = f_A(e_1, \ldots, e_k) = f_A(\phi_A(t_1, \alpha), \ldots, \phi_A(t_k, \alpha)) = \phi_A(t, \alpha),$$

proving the fact.

If $E_a(R)$ is right-linear, the term u is linear and we can apply the fact giving a particular $\alpha : \mathcal{X}_u \to A$. By choosing x^α arbitrarily in $\phi_1(x)$ for $x \in \mathcal{X} \setminus \mathcal{X}_u$ we obtain $\alpha : \mathcal{X} \to A$ for which $x^\alpha \in \phi_1(x)$ for all $x \in \mathcal{X}$ and $s = \phi_A(u, \alpha)$.

In the other case we assumed that $(A, >)$ is total. Then for every $x \in \mathcal{X}$ the finite non-empty multiset $\phi_1(x)$ has a unique maximum. Define $\alpha : \mathcal{X} \to A$ by choosing x^α to be this maximum for every $x \in \mathcal{X}$. Then one easily shows by induction on t that for this α we have $e \leq \phi_A(t, \alpha)$ for all terms t not containing a and all $e \in \phi_1(t)$; here we use again that every $e \in \phi_1(f(t_1, \ldots, t_k))$ can be written as $f_A(e_1, \ldots, e_k)$ for some $e_i \in \phi_1(t_i)$.

In all cases we have constructed some $\alpha : \mathcal{X} \to A$ for which $x^\alpha \in \phi_1(x)$ for all $x \in \mathcal{X}$ and $s \leq \phi_A(u, \alpha)$. Since $(A, >)$ is a well-founded monotone algebra for $E_a(R)$ and $l \to u$ is a rule of $E_a(R)$, we obtain

$$\phi_A(l, \alpha) > \phi_A(u, \alpha) \geq s.$$

One easily shows by induction on t that $\phi_A(t, \alpha) \in \phi_1(t)$ for all terms t not containing a. Since l is a term not containing a, we conclude that we have found an element $\phi_A(l, \alpha)$ in $\phi_1(l)$ which is strictly greater than s. Since this construction can be done for every

$s \in \phi_1(r)$, we conclude that the multiset $\phi_1(l)$ is strictly greater than the multiset $\phi_1(r)$. Hence

$$\phi_B(l,\beta) > \phi_B(r,\beta).$$

This holds for all $\beta : \mathcal{X} \to B$, so $l >_B r$, which concludes the proof of theorem 13. □

8 Application to SUBST

Let $\circ$ and $\cdot$ be binary symbols, λ a unary symbol, and 1 and $\uparrow$ constants. Consider the TRS:

$$\begin{array}{lcll} \lambda(x)\circ y & \to & \lambda(x\circ(1\cdot(y\circ\uparrow))) & \text{(Abs)} \\ (x\cdot y)\circ z & \to & (x\circ z)\cdot(y\circ z) & \text{(Map)} \\ (x\circ y)\circ z & \to & x\circ(y\circ z) & \text{(Ass).} \end{array}$$

In this section we prove that this system is totally terminating. According to proposition 8 then it is simply terminating, and according to proposition 7 every combination of this system and rules for which the right hand side can be embedded in the left hand side is also simply terminating. The system σ_0 in [2] is such a system. Hence this system is proven terminating by our method. The same holds for the original system SUBST in [9] if $\Lambda, F, S, \langle u,t\rangle$ is written instead of $\lambda, \uparrow, 1, t\cdot u$, respectively, and for the constant I the same value is chosen as for F. Both systems are developed for describing the rules for composition, pairing and Currying in the $\lambda\sigma$-calculus. This is a refinement of λ-calculus in which substitutions are manipulated explicitly. The rules are necessary for the propagation of substitution.

We see that the rule (Map) is a distribution rule for the operation '$\cdot$', and that '$\cdot$' does not occur in the left hand sides of the other rules. Hence according to theorem 13 it suffices to prove total termination of the eliminated system:

$$\begin{array}{lcll} \lambda(x)\circ y & \to & \lambda(x\circ 1) & \text{(Abs1)} \\ \lambda(x)\circ y & \to & \lambda(x\circ(y\circ\uparrow)) & \text{(Abs2)} \\ (x\circ y)\circ z & \to & x\circ(y\circ z) & \text{(Ass).} \end{array}$$

As a total well-founded monotone algebra we choose $\mathbb{N}\times\mathbb{N}\times\mathbb{N}$, where $\mathbb{N}$ consists of the integers ≥ 0, with the lexicographic order

$$(x_1,x_2,x_3) > (y_1,y_2,y_3) \iff x_1 > y_1 \vee (x_1 = y_1 \wedge (x_2 > y_2 \vee (x_2 = y_2 \wedge x_3 > y_3))).$$

We define the operations in this algebra as follows:

$$1 = \uparrow = (0,0,0),\quad \lambda(x_1,x_2,x_3) = (x_1+1,x_2,x_3),$$

$$(x_1,x_2,x_3)\circ(y_1,y_2,y_3) = (x_1+y_1, x_1(y_1+1)+x_2+y_2, 2x_3+y_3+1).$$

These operations are strictly monotone in all coordinates.

Let $(x_1,x_2,x_3),(y_1,y_2,y_3),(z_1,z_2,z_3)\in\mathbb{N}\times\mathbb{N}\times\mathbb{N}$ be arbitrary. For the rule (Abs1) we obtain

$$\begin{array}{rcl} \lambda(x_1,x_2,x_3)\circ(y_1,y_2,y_3) & = & (x_1+y_1+1,(x_1+1)(y_1+1)+x_2+y_2,\ldots) \\ & > & (x_1+1,x_1+x_2,\ldots) \\ & = & \lambda((x_1,x_2,x_3)\circ 1). \end{array}$$

For the rule (Abs2) we obtain

$$\begin{aligned}\lambda(x_1,x_2,x_3)\circ(y_1,y_2,y_3) &= (x_1+y_1+1,(x_1+1)(y_1+1)+x_2+y_2,\ldots)\\ &> (x_1+y_1+1,x_1y_1+x_1+x_2+y_1+y_2,\ldots)\\ &= \lambda((x_1,x_2,x_3)\circ((y_1,y_2,y_3)\circ\uparrow)).\end{aligned}$$

Finally, for the rule (Ass) we obtain

$$\begin{aligned}&((x_1,x_2,x_3)\circ(y_1,y_2,y_3))\circ(z_1,z_2,z_3)\\ &= (x_1+y_1+z_1,(x_1+y_1)(z_1+1)+x_1(y_1+1)+x_2+y_2+z_2,4x_3+2y_3+z_3+3)\\ &> (x_1+y_1+z_1,x_1(y_1+z_1+1)+y_1(z_1+1)+x_2+y_2+z_2,2x_3+2y_3+z_3+2)\\ &= (x_1,x_2,x_3)\circ((y_1,y_2,y_3)\circ(z_1,z_2,z_3)).\end{aligned}$$

This proves that the system is totally terminating. Note that the third coordinate of $\mathbb{N}\times\mathbb{N}\times\mathbb{N}$ is only essential for the rule (Ass) in the case of $x_1=0$.

9 Concluding remarks

We gave a classification of termination of term rewriting systems based upon types of orderings. The strongest type of termination we consider is polynomial termination: termination that can be proved by a polynomial interpretation. For the five proposed levels of termination we showed by very small examples that they are all distinct.

We gave some basic constructions for building domains and corresponding operations in which terms can be interpreted for proving termination.

We gave a construction of eliminating a particular operation symbol from a TRS and proved that under some restrictions termination of the original system can be derived from termination of the eliminated system. It is called distribution elimination since distributive rules are removed completely.

For a classical example of a TRS of which termination is very hard to prove ([9]) we gave a simple termination proof using distribution elimination.

References

[1] 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.

[2] CURIEN, P. L., HARDIN, T., AND RÍOS, A. Normalisation forte du calcul des substitutions. Tech. rep., LIENS Ecole Normale Supèrieure, 1991.

[3] DAUCHET, M. Simulation of Turing machines by a left-linear rewrite rule. 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. 109–120.

[4] DERSHOWITZ, N. Orderings for term rewriting systems. *Theoretical Computer Science 17*, 3 (1982), 279–301.

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

[6] DERSHOWITZ, N., AND JOUANNAUD, J.-P. Rewrite systems. In *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed., vol. B. Elsevier, 1990, ch. 6, pp. 243–320.

[7] DERSHOWITZ, N., AND MANNA, Z. Proving termination with multiset orderings. *Communications ACM 22*, 8 (1979), 465–476.

[8] GNAEDIG, I., AND LESCANNE, P. Proving termination of associative commutative rewriting systems by rewriting. In *Proceedings of the 8th Conference on Automated Deduction* (1986), J. H. Siekmann, Ed., vol. 230 of *Lecture Notes in Computer Science*, Springer, pp. 52–61.

[9] HARDIN, T., AND LAVILLE, A. Proof of termination of the rewriting system SUBST on CCL. *Theoretical Computer Science 46* (1986), 305–312.

[10] HIGMAN, G. Ordering by divisibility in abstract algebras. *Proc. London Mathematical Society 2*, 7 (1952), 326–336.

[11] HOFBAUER, D., AND LAUTEMANN, C. Termination proofs and the length of derivations (preliminary version). 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. 167–177.

[12] HUET, G., AND LANKFORD, D. S. On the uniform halting problem for term rewriting systems. Rapport Laboria 283, INRIA, 1978.

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

[14] KRUSKAL, J. Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture. *Trans. American Mathematical Society 95* (1960), 210–225.

[15] KURATOWSKI, K., AND MOSTOWSKI, A. *Set Theory.* North-Holland Publishing Company, 1968.

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

[17] LANKFORD, D. S. On proving term rewriting systems ar noetherian. Tech. Rep. MTP-3, Louisiana Technical University, Ruston, 1979.

[18] LAUTEMANN, C. A note on polynomial interpretation. *Bulletin of the EATCS 36* (1988), 129–131.

[19] LESCANNE, P. Termination of rewrite systems by elementary interpretations. Tech. Rep. 91-R-168, CRIN, 1991.

[20] MANNA, Z., AND NESS, S. On the termination of Markov algorithms. In *Proceedings of the Third Hawaii International Conference on System Science* (Honolulu, 1970), pp. 789–792.

[21] RUSINOWITCH, M. On termination of the direct sum of term rewriting systems. *Information Processing Letters 26* (1987), 65–70.

[22] 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.

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