Classifying termination of term rewriting

H. Zantema

RUU-CS-91-42
November 1991

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

Classifying termination of term rewriting

H. Zantema

Technical Report RUU-CS-91-42
November 1991

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

ISSN: 0924-3275

Classifying termination of term rewriting

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

Abstract

A classification of types of termination of term rewriting systems is proposed, built on properties in the semantic level in which terms are interpreted. It can be considered as a generalization of polynomial interpretations.

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 ([8, 2]). 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* ([4]). 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 [14].

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* ([11, 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. Furthermore 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.

A survey of the theory of term rewriting systems can be found in [5]. Overviews of existing techniques for termination detection of term rewriting systems can be found in [4, 14]. 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 term rewriting system if $l > r$ for each rewrite rule $l \to r$. This terminology is motivated by the following proposition.

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

Proof: Assume the term rewriting system 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 [13]. 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 ([11, 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 $x \in \mathcal{X}, \alpha : \mathcal{X} \to A, f \in \mathcal{F}, t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. This function ϕ 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 1 *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 2 *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 the lemma 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 term rewriting system if the corresponding reduction order $>_A$ normalizes the term rewriting system. This terminology is motivated by the following proposition.

Proposition 3 *A term rewriting system is terminating if and only if it is normalized by a non-empty well-founded monotone algebra.*

Proof: Assume the term rewriting system 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 term rewriting system 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 $\phi(l) >_A \phi(r)$ for all rewrite rules $l \to r$. Then according to the above proposition the term rewriting system is terminating.

For the case that $(A, >) = (\mathbb{N}, >)$, or, by translation equivalently,

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

for some natural number N, and the operations in the algebra A are polynomials, this corresponds to polynomial interpretations.

In the following examples we consider $\mathbb{N}$ to be defined as the set of integers strictly greater than zero.

Example 1: Consider the term rewriting system consisting of one rule, corresponding to associativity:

$$f(f(x, y), z) \to f(x, f(y, z)).$$

Define $(A, >) = (\mathbb{N}, >)$ and $f_A(x, y) = 2x + y$. Clearly it is strictly monotone in both coordinates, so $(A, >, f_A)$ is a well-founded monotone algebra. Further

$$\phi(f(f(x, y), z), \alpha) = 2(2x^\alpha + y^\alpha) + z^\alpha = 4x^\alpha + 2y^\alpha + z^\alpha$$

and

$$\phi(f(x, f(y,z)), \alpha) = 2x^\alpha + (2y^\alpha + z^\alpha) = 2x^\alpha + 2y^\alpha + z^\alpha.$$

Since the former is greater than the latter for all $\alpha : \{x,y,z\} \to \mathbb{N}$, we conclude that the term rewriting system is normalized by the well-founded monotone algebra, and so is terminating.

....

Example 2: Consider the term rewriting system consisting of the rule:

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

We give two termination proofs. First define $A = \mathbb{N} \times \mathbb{N}$ and define

$$(a,b) > (c,d) \iff a \geq c \wedge b \geq d \wedge (a,b) \neq (c,d).$$

Now $(A, >)$ is a well-founded poset; note that it is not total. Define

$$f_A(a,b) = (a+b, a) \quad \text{and} \quad g_A(a,b) = (b,a)$$

for all $(a,b) \in A$. Clearly both f_A and g_A are strictly monotone, so $(A, >, f_A, g_A)$ is a well-founded monotone algebra. Let $x^\alpha = (a,b)$, then

$$\phi(f(f(x)), \alpha) = f_A(f_A(a,b)) = f_A(a+b, a) = (2a+b, a+b)$$

and

$$\phi(f(g(f(x))), \alpha) = f_A(g_A(f_A(a,b))) = (2a+b, a).$$

The former is greater than the latter in the poset $(A, >)$ for all $(a,b) \in A$. So the term rewriting system is normalized by the well-founded monotone algebra, and so is terminating.

For the second proof define $A = \{0,1\} \times \mathbb{N}$ and define

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

Again $(A, >)$ is a non-total well-founded poset. Define

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

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

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

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

....

Example 3: Consider the term rewriting system consisting of the two rules:

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

Again define $A = \{0,1\} \times \mathbb{N}$ and $a,n) > (b,m) \Leftrightarrow a = b \wedge n < m$. Define

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

Both f_A and g_A are strictly monotone, while

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

So the term rewriting system is normalized by the well-founded monotone algebra, and so is terminating.

....

If no confusion is possible, we shall sometimes remove subscripts and superscripts, so we write $f, g, \ldots$ instead of $f_A, g_A, \ldots$, and write $x, y, \ldots$ instead of $x^\alpha, y^\alpha, \ldots$.

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 [5]. These definitions are motivated by the following two propositions.

Proposition 4 *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 ([6]), which is a special case of Kruskal's tree theorem ([10]); the relevance for termination of term rewriting systems is explained in [5]. 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 5 *Let $\mathcal{F}$ be finite and let $(A, >)$ be a non-empty simple monotone $\mathcal{F}$-algebra. Let R be a term rewriting system such that $l >_A r$ for all rewrite rules $l \to r$ of R. Then R is terminating.*

Proof: Apply proposition 4: 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 term rewriting system 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 6 *Let R be a term rewriting system 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: By definition all rules $l \to r$ of $Emb(\mathcal{F})$ satisfy $l_A >_A r_A$ for any simple monotone $\mathcal{F}$-algebra $(A, >)$. This proves (1) $\Leftrightarrow$ (2).

The implication (2) $\Rightarrow$ (3) is trivial. Finally, assume that (3) holds. Then according to proposition 3 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.

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 we shall 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 immediate from the following proposition.

Proposition 7 *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 $(A, >)$. □

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

Proposition 8 *The term rewriting system*

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

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 ([12]) 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. However, an example of this is expected to be more complicated than the system of proposition 8. In [12, 7] an example is given that the bound of $\exp(\exp(Cn))$ is sharp for polynomial termination. A smaller example with the same behaviour is

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

polynomial termination is shown by the interpretation $a(x,y) = 2x + y$ and $f(x) = x^2$, while for every n large enough the term $f^n(a(x,a(y,z)))$ allows a reduction of which the length exceeds $\exp(\exp(Cn))$ for some $C > 0$.

Proposition 9 *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

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

Proposition 11 *The term rewriting system*

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

is terminating but not simply terminating.

Proof: The proof of termination was given in example 2 of section 3. Assume it is simply terminating. According to proposition 6 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

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

□

6 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.

One of the common tools for proving termination of term rewriting systems is the recursive path order with status ([9, 3]). 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.

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] 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.

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

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

[5] 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.