Dummy Elimination: making termination easier

M. C. F. Ferreira and H. Zantema

UU-CS-1994-47
October 1994

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

Dummy Elimination: making termination easier

M. C. F. Ferreira and H. Zantema

Technical Report UU-CS-1994-47
October 1994

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

ISSN: 0924–3275

Dummy Elimination: making termination easier

M. C. F. Ferreira*and H. Zantema

Department of Computer Science, Utrecht University,
P.O. box 80.089, 3508 TB Utrecht, The Netherlands
e-mail: {maria, hansz}@cs.ruu.nl, fax: +30-513791

Abstract

Proving termination of term rewriting systems is a difficult task. Here we investigate a technique whose goal is to simplify that task. The technique consist of a transformation of the term rewriting system which eliminates function symbols considered "useless" and simplifies the rewrite rules. We show that the transformation is sound, i. e., termination of the original system can be inferred from termination of the transformed one. For proving this result we use a new notion of lifting of orders that turns out to be a generalization of the multiset construction.

1 Introduction

Suppose we want to prove termination of the following system

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

Intuitively, the function symbol a is created but seems not to have any influence on the reductions. Taking that into account, we can eliminate it and transform the given rule into

$$\begin{array}{lcl} f(g(x)) & \rightarrow & f(\diamond) \\ f(g(x)) & \rightarrow & g(g(f(x))) \\ f(g(x)) & \rightarrow & g(f(x)) \end{array}$$

where $\diamond$ is a fresh constant. Termination of the first system is not easy to prove (since the system is self-embedding orders like *recursive path order (rpo)* cannot be used) while termination of the second system is trivially proven with *rpo* by choosing the precedence $\triangleright$ satisfying $f \triangleright g \triangleright \diamond$. Now if the transformation is sound, i. e., termination of the original system can be inferred from termination of the transformed one, our task is done. In this paper we formally describe this transformation and prove its soundness with respect to termination.

In general, we are interested on simplifying the process of proving termination of term rewriting systems (TRS's). A possible approach to this goal is to devise sound transformations on TRS's such that the transformed systems are somehow easier to deal with, with respect to termination proofs, than the original ones. As examples of such transformations we have *transformation orderings* [1], *semantic labelling* [9] and *distribution elimination* [10] [1].

*Supported by NWO, the Dutch Organization for Scientific Research, under grant 612-316-041.

[1] For an example of application of some of these techniques, including the one described in this paper, see [11].

Briefly, on *transformation orderings*, a mapping ϕ from terms to terms is devised such that when a term rewrites to another their images under ϕ are related under a certain well-founded order. Both the function ϕ and the well-founded relation are obtained via some other TRS's. Between these TRS's, a property similar to confluence (*cooperation*) is required. In *semantic labelling*, labels from a certain domain are associated with the function symbols of the TRS. The new system contains rules where the function symbols are labelled, being the number of rules related to the cardinality of the set of labels chosen. In *distribution elimination*, function symbols occurring only on the right-hand-side (rhs) of rewrite rules or in distribution rules, are eliminated in a particular way, together with the distribution rules. The remaining rules are transformed. If the resulting system is right-linear, the transformation is sound.

The technique we present falls within the same category as *distribution elimination*, namely function symbols occurring only on the rhs of rules are eliminated and the rules transformed. No distribution rules are allowed and the transformation used is different. Termination of the original system can be deduced from termination of the transformed one, without any linearity restriction. Moreover in many cases our technique is stronger. For instance in the example we started with, *distribution elimination* is not helpful since the resulting system is, though right-linear, not terminating for that transformation.

As a technical mean to prove our result we make use of trees labelled with terms and of a new construction that lifts an order on a set to an order on trees labelled with labels from that set. This construction is interesting per se and therefore treated separately on section 3.

The rest of the paper is organized as follows. In section 2 we give some basic definitions on TRS's and orders. In section 3 we present the tree lifting of an order. It turns out that this lifting is a generalization of the multiset construction, monotone with respect to the order lifted and well-foundedness preserving. This will be the essential tool to be used on the proof of the main result. In section 4 we present the transformation on TRS's and prove its soundeness. The proof is conceptually simple although the technical details may not seem so. Nevertheless we give those details for the sake of completeness. Finally in section 5 we make some final remarks.

2 Basic notions

Below we introduce some notation used on the rest of the paper and give some basic notions over orders and TRS's. For more information on TRS's the reader is referred to [4].

A *poset* $(S, >)$ is a set S together with a partial order, i. e., an irreflexive and transitive relation, $> \subseteq S \times S$. Given a poset $(S, >)$, $M(S)$ denotes the finite multisets over S (see [5]) and $>_{mul}$ denotes the multiset extension of $>$ to $M(S)$, given by

$$S >_{mul} T \iff T = (S \setminus X) \cup Y \text{ such that } \emptyset \neq X \subseteq S \text{ and } \forall y \in Y\ \exists x \in X : \ x > y$$

The multiset extension of a partial order is itself a partial order and is well-foundedness preserving. Furthermore the multiset extension is monotone with respect to the order extended, i. e., if $>, \gg$ are orders over a set S and $> \subseteq \gg$ then $>_{mul} \subseteq \gg_{mul}$ in $M(S)$. We use the parentheses $[\,]$ to denote multisets being $[\,]$ the empty multiset.

Given a non-empty set A, we consider non-empty trees over A, defined by the following data type: $Tr(A) \cong A \times M(Tr(A))$, i. e., if f is the function from sets to sets given by $f(X) = A \times M(X)$, then $Tr(A)$ is the least fixed point of f. Therefore a tree is either a root, represented by $(a, [\,])$, with $a \in A$ and $[\,]$ the empty multiset, or a tree with root $a \in A$ and subtrees $t_1, \ldots, t_n$, represented by $(a, [t_1, \ldots, t_n])$. Since we are not interested in the order of the subtrees, we choose the multiset representation for the subtrees instead of a sequence representation.

The *depth* of a tree is given by the function **depth** and is defined inductively as follows:

- $\mathsf{depth}((s, [])) = 1$
- $\mathsf{depth}((s, [t_1, \ldots, t_m])) = 1 + max\{\mathsf{depth}(t_1), \ldots, \mathsf{depth}(t_m)\}$

Let $\mathcal{F}$ denote a *signature*, i. e., a non-empty set of varyadic function symbols, possibly infinite. Let $\mathcal{X}$ denote a denumerable set of variables such that $\mathcal{F} \cap \mathcal{X} = \emptyset$. The function $\mathsf{arity} : \mathcal{F} \cup \mathcal{X} \to \mathcal{P}(\mathbb{N})$, where $\mathcal{P}(\mathbb{N})$ represents the powerset of the natural numbers, gives the possible arities a symbol can have. Constants and variables have as arity the set $\{0\}$. The set of terms over $\mathcal{F}$ and $\mathcal{X}$ is denoted by $\mathcal{T}(\mathcal{F}, \mathcal{X})$ and the set of ground terms over $\mathcal{F}$ by $\mathcal{T}(\mathcal{F})$, and they are defined in the usual way. Given any term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, the set $Var(t)$ contains the variables occurring in t.

A *term rewriting system* (TRS) is a tuple $(\mathcal{F}, \mathcal{X}, R)$, where R is a subset of $\mathcal{T}(\mathcal{F}, \mathcal{X}) \times \mathcal{T}(\mathcal{F}, \mathcal{X})$. The elements (l, r) of R are called the rules of the TRS and are usually denoted by $l \to r$. They obey the restriction that l must be a non-variable and every variable in r must also occur in l. In the following, unless otherwise specified, we identify the TRS with R, being $\mathcal{F}$ the set of function symbols occurring in R.

A TRS R induces a *rewrite relation* over $\mathcal{T}(\mathcal{F}, \mathcal{X})$, denoted by $\to_R$, as follows: $s \to_R t$ iff $s = C[l\sigma]$ and $t = C[r\sigma]$, for some context C, substitution σ and rule $l \to r \in R$. The transitive closure of $\to_R$ is denoted by $\to_R^+$ and its reflexive-transitive closure by $\to_R^*$. By $\to_R^n$, with $n \in \mathbb{N}$, we denote the composition of $\to_R$ with itself n times (if $n = 0$, then $\to_R^n$ is the identity).

A TRS is called *terminating* (strongly normalizing or noetherian) if there exists no infinite sequence of the form $t_0 \to_R t_1 \to_R \ldots$.

3 Ordering trees

In this section we describe how to lift a partial order on a set A to a partial order on $Tr(A)$ in such a way that well-foundedness is preserved. This lifting will be used later on in the context of term rewriting.

Definition 3.1 *Let $(A, >)$ be a partially ordered set and consider $Tr(A)$, the finite non-empty trees over A. In $Tr(A)$ we define the following relation $\succ$*

$$t = (a, M) \succ (b, M') \iff \begin{cases} a > b \quad and \quad \forall u \in M' : (t \succ u) \text{ or } (\exists v \in M : v \succeq u) \\ a = b \quad and \quad M \succ_{mul} M' \end{cases}$$

where $\succ_{mul}$ is the multiset extension of $\succ$ and $\succeq \, = \, \succ \cup =$. We call the relation $\succ$ the tree lifting of $>$.

We remark that the above definition can be easily modified to cover quasi-orders. We are however particularly interested on tree liftings of relations that are partial orders so in this presentation we restrict ourselves to this case.

Lemma 3.2 *The relation $\succ$ is a partial order on $Tr(A)$.*

Proof We have to see that $\succ$ is a transitive and irreflexive relation on $Tr(A)$. We check first irreflexivity and we proceed by induction on the depth of a tree. Suppose $(a,[\,]) \succ (a,[\,])$, then we must have that $[\,] \succ_{mul} [\,]$, which is a contradiction. Suppose that $t \not\succ t$ for any tree t with $\mathsf{depth}(t) \leq n$, for a certain n, and let (a, M) be a tree of depth $n+1$. If $(a, M) \succ (a, M)$, then we must have $M \succ_{mul} M$, but since the multiset extension respects irreflexivity and $\succ$ is irreflexive over M (by induction hypothesis), we get a contradiction.

We check now that $\succ$ is transitive. If s, t and u are any trees such that $s \succ t$ and $t \succ u$, we need to see that $s \succ u$. We proceed by induction over $k = \mathsf{depth}(s)+\mathsf{depth}(t)+\mathsf{depth}(u)$. If $k = 3$ then $s = (a,[\,])$, $t = (b,[\,])$ and $u = (c,[\,])$ and we must have $a > b > c$. Since $>$ is transitive we conclude $a > c$ and therefore $s \succ u$. Suppose the property holds for triples of trees with $k \leq n$, for a certain n. Suppose now that $s = (a, M_a)$, $t = (b, M_b)$, $u = (c, M_c)$, $s \succ t$ and $t \succ u$ and $k = n+1$. We need to do some case analysis:

- $a = b = c$; then we must have $M_a \succ_{mul} M_b \succ_{mul} M_c$. Since the multiset extension respects transitivity and by induction hypothesis $\succ$ is transitive when comparing elements of M_a, M_b and M_c, we conclude that $M_a \succ_{mul} M_c$ and so $s \succ u$.
- $a = b$ and $b > c$; then we have

$$M_a \succ_{mul} M_b \text{ and } (\forall x \in M_c : t \succ x \text{ or } (\exists t' \in M_b : t' \succeq x))$$

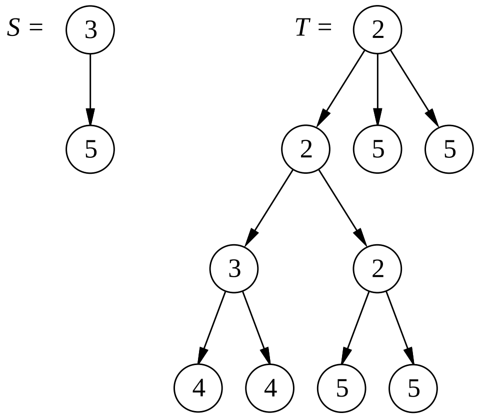
  Let $x \in M_c$, if $t \succ x$, since $s \succ t$ and $\mathsf{depth}(s) + \mathsf{depth}(t) + \mathsf{depth}(x) \leq k$, we can apply the induction hypothesis and conclude that $s \succ x$. If it is the case that there exists $t' \in M_b$ such that $t' \succeq x$, then, since $M_a \succ_{mul} M_b$, by definition of multiset extension, there exists $s' \in M_a$ such that $s' \succeq t'$. Again we can apply the induction hypothesis to conclude that $s' \succeq x$. Since $a > c$, by definition of $\succ$, we can conclude that $s \succ u$.
- $a > b$ and $b = c$; then we have

$$M_b \succ_{mul} M_c \text{ and } (\forall x \in M_b : s \succ x \text{ or } (\exists s' \in M_a : s' \succeq x))$$

  Let $x \in M_c$, then there is an element $t' \in M_b$ such that $t' \succeq x$ (since $M_b \succ_{mul} M_c$). But then either $s \succ t'$, and by induction hypothesis we get $s \succ x$, or there exists $s' \in M_a$ such that $s' \succeq t'$, and again by induction hypothesis we get $s' \succeq x$. Since $a > c$, by definition of $\succ$, we have $s \succ u$.
- $a > b > c$; then we also have $(\forall x \in M_b : s \succ x \text{ or } (\exists s' \in M_a : s' \succeq x))$ and $(\forall y \in M_c : t \succ y \text{ or } (\exists t' \in M_b : t' \succeq y))$. Take $u' \in M_c$. If $t \succ u'$ then by induction hypothesis and since $s \succ t$, we conclude that $s \succ u'$. If there is $t' \in M_b$ such that $t' \succeq u'$, then either
  - $s \succ t'$ and again by induction hypothesis we have $s \succ u'$; or
  - there exists $s' \in M_a$ such that $s' \succeq t'$, and by induction hypothesis we get $s' \succeq u'$. Since $a > c$, by definition of $\succ$, we have $s \succ u$.

□

Example 1
We present an example of this construction. Let $(A, >)$ be the natural numbers with the usual order. Let



From definition 3.1 it follows that $S \succ T$. Note that even though $S \succ T$, the depth of T is greater than the depth of S.

The construction presented in definition 3.1 has many interesting properties, as we show below. Namely it is monotonic with respect to the order lifted, preserves well-foundedness and is a proper generalization of the multiset construction.

Lemma 3.3 *Let A be a set and $>, \gg$ two partial orders in S such that $> \subseteq \gg$. Consider $Tr(A)$ with the partial orders $\succ^{>}$ and $\succ^{\gg}$, the tree liftings of respectively $>$ and $\gg$. Then $\succ^{>} \subseteq \succ^{\gg}$.*

Proof We need to see that given two trees $S, T \in Tr(A)$ if $S \succ^{>} T$ then $S \succ^{\gg} T$. We proceed by induction on $k = \mathsf{depth}(S) + \mathsf{depth}(T)$. If $k = 2$ then $S = (a, [])$ and $T = (b, [])$ and we must have $a > b$. Consequently $a \gg b$ and therefore $S \succ^{\gg} T$.

Suppose the result holds for trees U, V with $\mathsf{depth}(U) + \mathsf{depth}(V) \leq n$. Let $S = (a, M_a)$ and $T = (b, M_b)$ with $\mathsf{depth}(S) + \mathsf{depth}(T) = n + 1$ and $S \succ^{>} T$. We must have either

- $a > b$ and for all $u \in M_b$ either $S \succ^{>} u$ or there is a tree $v \in M_a$ such that $v \succeq^{>} u$. In this case also $a \gg b$ and by induction hypothesis either $S \succ^{\gg} u$ or $v \succeq^{\gg} u$, so $S \succ^{\gg} T$.
- $a = b$ and $M_a \succ^{>}_{mul} M_b$. Since the multiset extension is monotone with respect to the extended order, and comparisons between the elements of M_a and M_b envolve trees u, v with $\mathsf{depth}(u) + \mathsf{depth}(v) < n$, we can apply the induction hypothesis and conclude that $M_a \succ^{\gg}_{mul} M_b$ and therefore that $S \succ^{\gg} T$.

□

Lemma 3.4 *Let $(A, >)$ be a partially ordered set. Then there is an order-preserving injection from $(M(A), >_{mul})$ to $(Tr(A), \succ)$, where $(M(A), >_{mul})$ is the multiset extension of $(A, >)$.*

Proof (Sketch) Fix $r \in A$, arbitrarily chosen. The function $\phi_r : M(A) \to Tr(A)$ is given by:

- $\phi_r([]) = (r, [])$
- $\phi_r([s_1, \ldots, s_k]) = (r, [(s_1, []), \ldots, (s_k, [])])$

It is not difficult to see that ϕ_r is well-defined and provides such an injection. □

Essential for our purposes is the preservation of well-foundedness, stated in the next result.

Theorem 3.5 *Let $(A, >)$ be a poset. Then $>$ is well-founded on A if and only if $\succ$ is well-founded on $Tr(A)$.*

Proof For the "if" part, suppose that $>$ is not well-founded in A. Then there is an infinite descending chain $a_0 > a_1 > \cdots$. According to the definition of $\succ$ then $(a_0, [\,]) \succ (a_1, [\,]) \succ \ldots$, is an infinite descending chain on $Tr(A)$, contradicting well-foundedness of $\succ$.

For the "only-if" part we will use the *recursive path order*, $>_{rpo}$, on trees, based on $>$ and with multiset status (for a definition of $>_{rpo}$ see for example [2, 3]), given by

$$(a, M) >_{rpo} (b, N) \iff \begin{cases} \exists u \in M : \ u \geq_{rpo} (b, N), \text{ or} \\ (a > b) \text{ and } (\forall u \in N : \ (a, M) >_{rpo} u), \text{ or} \\ (a = b) \text{ and } (M >_{rpo,mul} N) \end{cases}$$

Since $>_{rpo}$ is well-founded whenever $>$ is well-founded (for a simple proof see [7]), we only need to check that $\succ \subseteq >_{rpo}$.

We will prove that for any trees $S, T \in Tr(A)$, $S \succ T \Rightarrow S >_{rpo} T$, by induction on $k = \mathsf{depth}(S) + \mathsf{depth}(T)$. If $k = 2$ then $S = (a, [\,])$ and $T = (b, [\,])$, with $a, b \in A$ and we must have $a > b$. By definition of $>_{rpo}$ we also conclude that $S >_{rpo} T$.

Suppose now that $S \succ T$ and $\mathsf{depth}(S) + \mathsf{depth}(T) \leq n$, for a fixed n, implies $S >_{rpo} T$. Let $S_1 = (a, M_a)$ and $T_1 = (b, M_b)$ with $S_1 \succ T_1$ and $\mathsf{depth}(S_1) + \mathsf{depth}(T_1) = n + 1$. If $a > b$ then, by definition of $\succ$, we have for all $u \in M_b$ either $S_1 \succ u$ or $v \succeq u$, for some $v \in M_a$. To conclude that $S_1 >_{rpo} T_1$ we have to see that $S_1 >_{rpo} u$, for all $u \in M_b$. Take $u \in M_b$ arbitrary. If $S_1 \succ u$, then by induction hypothesis we have $S_1 >_{rpo} u$. If there exists $v \in M_a$ such that $v \succeq u$ then again by induction hypothesis we conclude $v \geq_{rpo} u$. Since $v \in M_a$ and $>_{rpo}$ has the subtree property, we have $S_1 >_{rpo} v$. Transitivity of $>_{rpo}$ gives $S_1 >_{rpo} u$, as we wanted.

If $a = b$ then $M_a \succ_{mul} M_b$, i. e.,

$$M_b = (M_a \setminus X) \cup Y \text{ with } \emptyset \neq X \subseteq M_a \text{ and } \forall y \in Y \ \exists x \in X : x \succ y$$

Since $\mathsf{depth}(x) + \mathsf{depth}(y) \leq n$, by induction hypothesis we also have $x >_{rpo} y$ so $M_a >_{rpo,mul} M_b$ and consequently $S_1 >_{rpo} T_1$.

We have just seen that $\succ \subseteq >_{rpo}$ and since the last order is well-founded, so is the former. □

We give a sketch of an alternative proof for theorem 3.5. For each leaf in a tree T, consider the multiset of labels of nodes in the path from the root to the leaf. Define $\mathsf{Path}(T)$ to be the multiset of all these multisets (i. e., $\mathsf{Path}(T)$ is an element of $M(M(A))$). Now one can see that for any trees $S, T \in Tr(A)$, $S \succ T \Rightarrow \mathsf{Path}(S) \gg \mathsf{Path}(T)$, where $\gg = (>_{mul})_{mul}$ is the extension in $M(M(A))$ of $>$, i. e., $\gg$ is the multiset extension of the multiset extension of $>$. Since $>$ is well-founded on A and multiset extension preserves well-foundedness, $\gg$ is also well-founded on $M(M(A))$ and as a consequence $\succ$ is well-founded on $Tr(A)$.

A property not preserved by the tree lifting is totality. Again take $(A, >)$ to be the natural numbers with the usual order. Let

Then according to the definition of $\succ$ neither $S \succ T$ nor $T \succ S$. Since $S \neq T$, the order $\succ$ is obviously not total. Note that $T >_{rpo} S$.

4 Transforming the TRS

In this section we present the transformation on TRS's and we show how termination of the original system can be inferred from termination of the transformed system.

We establish first some terminology. Let $\mathcal{F}$ be a set of varyadic function symbols and $\mathcal{X}$ a set of variables with $\mathcal{F} \cap \mathcal{X} = \emptyset$. Let a be a function symbol with non-null arities, i. e., $N > 0$, for all $N \in \mathsf{arity}(a)$, and not occurring in $\mathcal{F}$. Let $\diamond$ be a constant also not occurring in $\mathcal{F}$. We denote by $\mathcal{F}_a$ and $\mathcal{F}_\diamond$ respectively the sets $\mathcal{F} \cup \{a\}$ and $\mathcal{F} \cup \{\diamond\}$.

We consider TRS's over $T(\mathcal{F}_a, X)$ such that the function symbol a only occurs (eventually) in the right-hand-side (rhs) of the rules of the TRS. The idea behind the transformation is that the fuction symbol a, not occurring in the left-hand-side (lhs) of rewrite rules, does not take a relevant role in the reductions and therefore should not influence the termination behaviour of the TRS. We consider the symbol a as a kind of blocker and given a term over $T(\mathcal{F}_a, \mathcal{X})$, we decompose it on its "components". Those components are terms over $T(\mathcal{F}_\diamond, \mathcal{X})$, more especifically, the subterms above and below the occurrences of the symbol a, with those occurrences being replaced by the constant $\diamond$. We make this more precise.

Definition 4.1 *Given a term $t \in T(\mathcal{F}_a, \mathcal{X})$, the cap of t, denoted by $\mathsf{cap}(t)$, is a term over $T(\mathcal{F}_\diamond, \mathcal{X})$ given by the function cap: $T(\mathcal{F} \cup \{a, \diamond\}, \mathcal{X}) \to T(\mathcal{F}_\diamond, \mathcal{X})$, defined inductively as follows:*

- $\mathsf{cap}(x) = x$, *for any* $x \in \mathcal{X}$
- $\mathsf{cap}(f(t_1, \ldots, t_m)) = f(\mathsf{cap}(t_1), \ldots, \mathsf{cap}(t_m))$, *if* $f \in \mathcal{F}$ *and* $m \in \mathsf{arity}(f)$ $(f \neq a)$
- $\mathsf{cap}(a(t_1, \ldots, t_N)) = \diamond$, *with* $N \in \mathsf{arity}(a)$

Note that strictly speaking the domain of cap need only be $T(\mathcal{F}_a, \mathcal{X})$, however to simplify the treatment later (basicly avoid defining an extension of cap only to include $\diamond$ in its domain), we use the extended signature $\mathcal{F}_a \cup \{\diamond\}$. This same observation applies to the next definition.

Since we are interested also in the subterms hanging under occurrences of the symbol a, we need another operation that collects all the caps of the subterms encapsulated between occurrences of the symbol a.

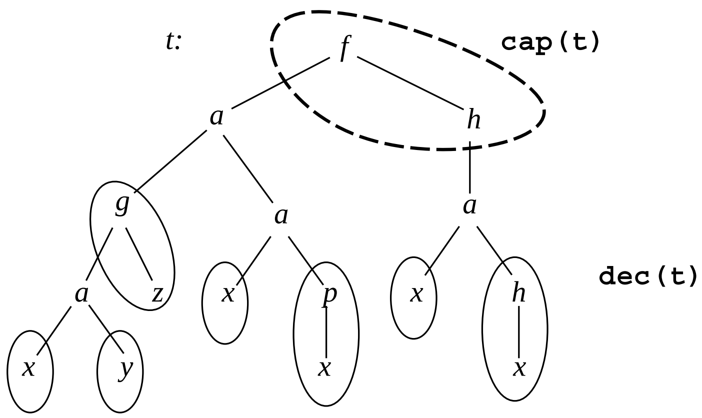
Definition 4.2 *For any term $t \in T(\mathcal{F}_a, \mathcal{X})$, its decomposition is denoted by $\mathsf{dec}(t)$, where $\mathsf{dec} : T(\mathcal{F} \cup \{a, \diamond\}, \mathcal{X}) \to M(T(\mathcal{F}_\diamond, \mathcal{X}))$ is defined inductively as follows:*

- $\mathsf{dec}(x) = [\,]$
- $\mathsf{dec}(f(t_1, \ldots, t_m)) = \bigcup_{i=1}^{m} \mathsf{dec}(t_i)$, *if* $f \in \mathcal{F}$ *and* $m \in \mathsf{arity}(f)$ $(f \neq a)$

- $\mathsf{dec}(a(t_1,\ldots,t_N)) = \bigcup_{i=1}^{N} ([\mathsf{cap}(t_i)] \cup \mathsf{dec}(t_i))$, *with* $N \in \mathsf{arity}(a)$

The decomposition of a term collects all caps of the terms between occurrences of a's and adds a symbol $\diamond$ for each occurrence of a encountered except for the topmost. For reasons that will become clear later, we treat differently the cap of the whole term (which is not collected in the decomposition of the term).

Example 2
The following term t



has as cap the term $f(\diamond, h(\diamond))$ and its decomposition is given by

$$\mathsf{dec}(t) = [g(\diamond, z), x, y, \diamond, x, p(x), x, h(x)]$$

We can now define the transformation on the TRS. As can be expected we will decompose the right-hand-side of the rules in R and create new rules using this decomposition.

Definition 4.3 *Given a TRS R over $T(\mathcal{F}_a, \mathcal{X})$ such that the function symbol a occurs at most on the right-hand-side of the rules in R, $E(R)$ is the TRS over $T(\mathcal{F}_\diamond, \mathcal{X})$ given by*

$$E(R) = \{l \to u \mid (l \to r) \in R \text{ and } u = \mathsf{cap}(r) \text{ or } u \in \mathsf{dec}(r)\}$$

Example 3
Let R be given by the rules

$$f(f(x)) \to g(a(f(x), x)) \qquad\qquad g(g(x)) \to f(g(x))$$

Then the transformed TRS, $E(R)$ is given by:

$$\begin{array}{ll} f(f(x)) \to g(\diamond) & \qquad g(g(x)) \to f(g(x)) \\ f(f(x)) \to f(x) & \\ f(f(x)) \to x & \end{array}$$

Note that if the function symbol a does not occur on a term t then $\mathsf{cap}(t) = t$ and $\mathsf{dec}(t) = \emptyset$, hence rules in which a does not occur are not affected by the transformation.

From the definition of E, we can see that in general the TRS $E(R)$ has more rules but is syntactically simpler than the original one, so the transformation can be quite useful if we are able to infer termination of R from termination of $E(R)$. Termination however is not preserved, i. e., if R is terminating, $E(R)$ is not necessarily terminating, as the following example shows. Consider the terminating TRS R given by:

$$f(x,x) \rightarrow f(a(x),x)$$

The transformed TRS $E(R)$ is given by:

$$\begin{array}{lcl} f(x,x) & \rightarrow & f(\diamond,x) \\ f(x,x) & \rightarrow & x \end{array}$$

and is obviously not terminating.

The main purpose of this paper is to show that termination of $E(R)$ implies termination of R. Before going into the technical details we give a general idea of the proof. If $E(R)$ is terminating, the relation $\rightarrow^+_{E(R)}$ is well-founded. If we consider the poset $(Tr(T(\mathcal{F}_\diamond,\mathcal{X})),\succ)$ (where $\succ$ is the tree extension of $\rightarrow^+_{E(R)}$ as defined in 3.1) then $\succ$ is also well-founded. We now use the trees over $T(\mathcal{F}_\diamond,\mathcal{X})$ to interpret the terms of $T(\mathcal{F}_a,\mathcal{X})$ in such a way that for terms $s,t \in T(\mathcal{F}_a,\mathcal{X})$ if $s \rightarrow_R t$ then $\mathsf{tree}(s) \gg \mathsf{tree}(t)$, where $\mathsf{tree}(u)$ is a tree over $T(\mathcal{F}_\diamond,\mathcal{X})$ associated with the term u, and $\gg$ is a well-founded extension of $\succ$. Termination of R follows from well-foundedness of $\gg$.

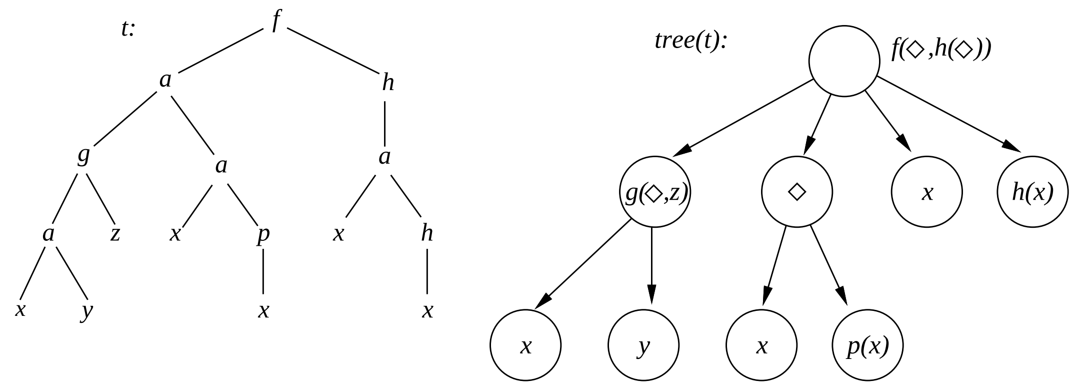
We introduce some definitions and auxiliary results. Recall the definition of tree lifting of an order from section 3.

Definition 4.4 *Given a term $t \in T(\mathcal{F}_a,\mathcal{X})$ we associate to it a tree over $T(\mathcal{F}_\diamond,\mathcal{X})$, denoted by* $\mathsf{tree}(t)$*, where* $\mathsf{tree} : T(\mathcal{F} \cup \{a,\diamond\},\mathcal{X}) \rightarrow Tr(T(\mathcal{F}_\diamond,\mathcal{X}))$ *is defined as follows:*

- $\mathsf{tree}(x) = (x,[])$, *for any* $x \in \mathcal{X}$
- $\mathsf{tree}(f(s_1,\ldots,s_m)) = (\mathsf{cap}(f(s_1,\ldots,s_m)), \bigcup_{i=1}^{m} M_i)$, *where* $\mathsf{tree}(s_i) = (\mathsf{cap}(s_i), M_i)$
- $\mathsf{tree}(a(s_1,\ldots,s_N)) = (\mathsf{cap}(a(s_1,\ldots,s_N)), \bigcup_{i=1}^{N} [\mathsf{tree}(s_i)])$

An observation similar to the one made after definition 4.1 is in order here. We will apply the function tree to terms that may contain the constant $\diamond$, therefore we include it already in the domain of the function.

Example 4
The following picture shows the same term as in example 2 together with its corresponding tree.



To maintain the domains of related functions consistent, the same has to be done in the following definition.

Definition 4.5 *Given a tree T over $T(\mathcal{F}_\diamond, \mathcal{X})$, the multiset of its labels is given by the function* $\mathsf{labels} : Tr(T(\mathcal{F} \cup \{a, \diamond\}, \mathcal{X})) \to M(T(\mathcal{F}_\diamond, \mathcal{X}))$*, defined inductively as:*

- $\mathsf{labels}((s, [])) = [s]$
- $\mathsf{labels}((s, [T_1, \dots, T_k])) = [s] \cup \left(\bigcup_{i=1}^{k} \mathsf{labels}(T_i) \right)$

The following lemma is easily proven by induction on the structure of the terms.

Lemma 4.6 *Let $t \in T(\mathcal{F}_a, \mathcal{X})$ be any term. If* $\mathsf{tree}(t) = (\mathsf{cap}(t), [T_1, \dots, T_k])$ *then* $\mathsf{dec}(t) = \bigcup_{i=1}^{k} \mathsf{labels}(T_i)$*, where the union taken over zero elements is the empty multiset.*

Lemma 4.6 allows us to rewrite definition 4.3 equivalently as:

Definition 4.7 *Given a TRS R over $T(\mathcal{F}_a, \mathcal{X})$ such that the function symbol a occurs at most on the right-hand-side of the rules in R, $E(R)$ is a TRS over $T(\mathcal{F}_\diamond, \mathcal{X})$ given by*

$$E(R) = \{l \to u \mid (l \to r) \in R \text{ and } u \in \mathsf{labels}(\mathsf{tree}(r))\}$$

Lemma 4.8 *Let $t \in T(\mathcal{F}_a, \mathcal{X})$ and $\sigma : \mathcal{X} \to T(\mathcal{F}_a, \mathcal{X})$ be an arbitrary substitution. We define $\tau : \mathcal{X} \to T(\mathcal{F}_\diamond, \mathcal{X})$ as the substitution satisfying $\tau(x) = \mathsf{cap}(\sigma(x))$, for all $x \in \mathcal{X}$. Then $\mathsf{cap}(t\sigma) = \mathsf{cap}(t)\tau$.*

Proof We proceed by induction on t. If $t = x \in \mathcal{X}$ then $\mathsf{cap}(t\sigma) = \mathsf{cap}(\sigma(x)) = \tau(x) = t\tau = \mathsf{cap}(t)\tau$, by definition of τ and 4.1. If $t = f(t_1, \dots, t_m)$ then

$$\begin{array}{lcl} \mathsf{cap}(f(t_1, \dots, t_m)\sigma) & = & \mathsf{cap}(f(t_1\sigma, \dots, t_m\sigma)) \\ \text{(by definition 4.1)} & = & f(\mathsf{cap}(t_1\sigma), \dots, \mathsf{cap}(t_m\sigma)) \\ \text{(by induction hypothesis)} & = & f(\mathsf{cap}(t_1)\tau, \dots, \mathsf{cap}(t_m)\tau) \\ & = & \mathsf{cap}(f(t_1, \dots, t_m))\tau \end{array}$$

If $t = a(t_1, \dots, t_N)$ then by definition 4.1, $\mathsf{cap}(t\sigma) = \diamond = \diamond\tau = \mathsf{cap}(t)\tau$. □

Lemma 4.9 *Let t be a non-ground term in $\mathcal{T}(\mathcal{F},\mathcal{X})$[2]. Let $\sigma : \mathcal{X} \to T(\mathcal{F}_a,\mathcal{X})$ be any substitution and let $x \in Var(t)$. Let* $\mathsf{tree}(t\sigma) = (\mathsf{cap}(t\sigma), M_t)$ *and* $\mathsf{tree}(\sigma(x)) = (\mathsf{cap}(\sigma(x)), M_x)$. *Then $M_x \subseteq M_t$ (being $\subseteq$ multiset inclusion).*

Proof Since $x \in Var(t)$ we can write t as $C[x]$, for some context $C[\ ]$. We prove the lemma by induction on the context. Suppose $C = \Box$, then $t = x$ and the result holds. Suppose now that $t = f(t_1, \ldots, D[x], \ldots, t_k)$, with $D[x]$ occurring at some position j, $1 \leq j \leq k$, such that $\mathsf{tree}(D[x]\sigma) = (\mathsf{cap}(D[x]\sigma), M)$, $\mathsf{tree}(\sigma(x)) = (\mathsf{cap}(\sigma(x)), M_x)$ and $M_x \subseteq M$. Since

$$\mathsf{tree}(t\sigma) = \mathsf{tree}(f(t_1\sigma, \ldots, D[x]\sigma, \ldots, t_k\sigma)) = (\mathsf{cap}(f(t_1\sigma, \ldots, D[x]\sigma, \ldots, t_k\sigma)), \bigcup_{i=1}^{k} M_i)$$

where $\mathsf{tree}(t_i\sigma) = (\mathsf{cap}(t_i\sigma), M_i)$, for all $i \neq j$ and for $i = j$, $M_j = M$, we have that $M_x \subseteq M \subseteq \bigcup_{i=1}^{k} M_i$, and the result follows. $\Box$

Remark. From now on we assume that $E(R)$ is terminating and define $>$ to be $\to^{+}_{E(R)}$. Therefore $>$ is well-founded and closed under contexts and substitutions.[3] In $Tr(T(\mathcal{F}_\diamond,\mathcal{X}))$ we consider $\succ$, the tree lifting of $>$. Since $>$ is a well-founded partial order we also have that $\succ$ is a well-founded partial order on $Tr(T(\mathcal{F}_\diamond,\mathcal{X}))$.

Lemma 4.10 *Let $t \in \mathcal{T}(\mathcal{F},\mathcal{X}) \setminus \mathcal{X}$. Let $x \in Var(t)$ such that $t > x$. Let $\sigma : \mathcal{X} \to T(\mathcal{F}_a,\mathcal{X})$ be any substitution. Then* $\mathsf{tree}(t\sigma) \succ \mathsf{tree}(\sigma(x))$, *for any substitution $\sigma : \mathcal{X} \to T(\mathcal{F}_a,\mathcal{X})$.*

Proof By definitions 4.4 and 4.1, and lemma 4.8, $\mathsf{tree}(t\sigma) = (\mathsf{cap}(t\sigma), M_t) = (\mathsf{cap}(t)\tau, M_t) = (t\tau, M_t)$, where $\tau : \mathcal{X} \to T(\mathcal{F}_\diamond,\mathcal{X})$ is defined by $\tau(x) = \mathsf{cap}(\sigma(x))$, for all $x \in \mathcal{X}$. Similarly, $\mathsf{tree}(\sigma(x)) = (\mathsf{cap}(\sigma(x)), M_x) = (\tau(x), M_x)$. Since the conditions of lemma 4.9 are satisfied, we conclude that $M_x \subseteq M_t$ and since $t > x$ and $>$ is closed under substitutions (in $T(\mathcal{F}_\diamond,\mathcal{X})$), we can conclude that $t\tau > \tau(x)$. Now it is clear from definition 3.1 that $\mathsf{tree}(t\sigma) \succ \mathsf{tree}(\sigma(x))$. $\Box$

Lemma 4.11 *Let $s \in \mathcal{T}(\mathcal{F},\mathcal{X}) \setminus \mathcal{X}$ and $t \in T(\mathcal{F}_a,\mathcal{X})$ such that $Var(t) \subseteq Var(s)$ and $s > v$ for all $v \in$* $\mathsf{dec}(t)$. *Let $\sigma : \mathcal{X} \to T(\mathcal{F}_a,\mathcal{X})$ be any substitution and suppose that* $\mathsf{tree}(s\sigma) = (\mathsf{cap}(s\sigma), M_s)$, $\mathsf{tree}(t\sigma) = (\mathsf{cap}(t\sigma), M_t)$. *Then for all $U \in M_t$ either $U \in M_s$ or* $\mathsf{tree}(s\sigma) \succ U$.

Proof We proceed by induction on the structure of t. If $t = x \in \mathcal{X}$ then the result follows from lemma 4.9.

For $t = f(t_1, \ldots, t_m)$, $\mathsf{tree}(t\sigma) = \mathsf{tree}(f(t_1\sigma, \ldots, t_m\sigma)) = (\mathsf{cap}(f(t_1\sigma, \ldots, t_m\sigma)), \bigcup_{i=1}^{m} M_i)$, where $\mathsf{tree}(t_i\sigma) = (\mathsf{cap}(t_i\sigma), M_i)$, for all $1 \leq i \leq m$. Fix some i, $1 \leq i \leq m$. Since $\mathsf{dec}(t) = \bigcup_{j=1}^{m} \mathsf{dec}(t_j)$ and by hypothesis $s > v$ for all $v \in \mathsf{dec}(t)$, we also have that $s > u$ for any $u \in \mathsf{dec}(t_i)$. Also $Var(t_i) \subseteq Var(t) \subseteq Var(s)$, so we can apply the induction

[2] Note that the symbol a does not occur on t.

[3] In fact any well-founded partial order over $T(\mathcal{F}_\diamond,\mathcal{X})$ compatible with $E(R)$, closed under substitutions and contexts would do.

hypothesis to t_i and conclude that given any $U \in M_i$ either $U \in M_s$ or $\mathsf{tree}(s\sigma) \succ U$. Since $U \in \bigcup_{j=1}^{m} M_j \Rightarrow U \in M_i$, for some $1 \leq i \leq m$, the result holds.

If $t = a(t_1, \ldots, t_N)$ then $\mathsf{tree}(t\sigma) = (\diamond, \bigcup_{i=1}^{N} [\mathsf{tree}(t_i\sigma)])$. We need to see that for any i, $1 \leq i \leq N$, either $\mathsf{tree}(t_i\sigma) \in M_s$ or $\mathsf{tree}(s\sigma) \succ \mathsf{tree}(t_i\sigma)$.

Fix then any i, $1 \leq i \leq N$. By lemma 4.8 we know that $\mathsf{tree}(t_i\sigma) = (\mathsf{cap}(t_i\sigma), M_i) = (\mathsf{cap}(t_i)\tau, M_i)$, where $\tau : \mathcal{X} \to T(\mathcal{F}_\diamond, \mathcal{X})$ is given by $\tau(x) = \mathsf{cap}(\sigma(x))$, for all $x \in \mathcal{X}$. Also $\mathsf{tree}(s\sigma) = (\mathsf{cap}(s\sigma), M_s) = (\mathsf{cap}(s)\tau, M_s) = (s\tau, M_s)$. By hypothesis $s > u$ for all $u \in \mathsf{dec}(t)$ and since $\mathsf{dec}(t) = \bigcup_{j=1}^{N} ([\mathsf{cap}(t_j)] \cup \mathsf{dec}(t_j))$, we can say that $s > u$ for all $u \in \mathsf{dec}(t_i)$. Further $Var(t_i) \subseteq Var(t) \subseteq Var(s)$, so we can apply the induction hypothesis to t_i and conclude that if $U \in M_i$ then either $U \in M_s$ or $\mathsf{tree}(s\sigma) \succ U$. Since by hypothesis $s > \mathsf{cap}(t_i)$, and $>$ is closed under substitutions we conclude that $s\tau > \mathsf{cap}(t_i)\tau$ and by definition 3.1 we have $\mathsf{tree}(s\sigma) \succ \mathsf{tree}(t_i\sigma)$, as we wanted. □

Lemma 4.12 *Let $s \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}$ and $t \in T(\mathcal{F}_a, \mathcal{X})$ such that $Var(t) \subseteq Var(s)$ and $s > v$ for all $v \in \mathsf{dec}(t) \cup [\mathsf{cap}(t)]$. Finally let $\sigma : \mathcal{X} \to T(\mathcal{F}_a, \mathcal{X})$ be any substitution. Then $\mathsf{tree}(s\sigma) \succ \mathsf{tree}(t\sigma)$.*

Proof By definition 4.4 and lemma 4.8, $\mathsf{tree}(s\sigma) = (\mathsf{cap}(s\sigma), M_s) = (\mathsf{cap}(s)\tau, M_s) = (s\tau, M_s)$ and $\mathsf{tree}(t\sigma) = (\mathsf{cap}(t\sigma), M_t) = (\mathsf{cap}(t)\tau, M_t)$, where $\tau : \mathcal{X} \to T(\mathcal{F}_\diamond, \mathcal{X})$ is given by $\tau(x) = \mathsf{cap}(\sigma(x))$, for all $x \in \mathcal{X}$. By lemma 4.11 we conclude that for any $U \in M_t$ either $U \in M_s$ or $\mathsf{tree}(s\sigma) \succ U$. Since $s > \mathsf{cap}(t)$ and $>$ is closed under substitutions, we have $s\tau > \mathsf{cap}(t)\tau$, and by definition 3.1 we conclude that $\mathsf{tree}(s\sigma) \succ \mathsf{tree}(t\sigma)$. □

Lemma 4.13 *Let $l \to r$ be a rule in R and $\sigma : \mathcal{X} \to T(\mathcal{F}_a, \mathcal{X})$ an arbitrary substitution. Then $\mathsf{tree}(l\sigma) \succ \mathsf{tree}(r\sigma)$.*

Proof From the definition of $E(R)$ (see definition 4.3), we know that $l \to u$, with $u \in [\mathsf{cap}(r)] \cup \mathsf{dec}(r)$, is a rule in $E(R)$ and therefore $l > u$ for any $u \in [\mathsf{cap}(r)] \cup \mathsf{dec}(r)$. Also $Var(r) \subseteq Var(l)$ and a does not occur in l, therefore all the hypothesis of lemma 4.12 are satisfied, so we can apply it to conclude that $\mathsf{tree}(l\sigma) \succ \mathsf{tree}(r\sigma)$. □

We need to see that if $s \to_R t$ then $\mathsf{tree}(s) \succ \mathsf{tree}(t)$. For the case that $s = l\sigma$ and $t = r\sigma$, for some rule $l \to r \in R$ and substitution $\sigma : \mathcal{X} \to T(\mathcal{F}_a, \mathcal{X})$, the result holds as was seen in lemma 4.13, but unfortunately the tree construction is not closed under context, i. e., if $s \to_R t$, $\mathsf{tree}(s) \succ \mathsf{tree}(t)$ and C is any non-trivial context then $\mathsf{tree}(C[s]) \succ \mathsf{tree}(C[t])$ is not necessarily true as can be seen in the next example. Let R be:

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

The system $E(R)$ is given by:

$$\begin{aligned} f(x) &\to g(\diamond) \\ f(x) &\to x \end{aligned}$$

In R we have the rewrite step $f(0) \to g(a(0))$. In $E(R)$ we have both $f(0) \to g(\diamond)$ and $f(0) \to 0$. Consequently

$$\mathsf{tree}(f(0)) = (f(0), []) \succ (g(\diamond), [(0, [])]) = \mathsf{tree}(g(a(0)))$$

Take now $C[] = g(\square)$ and consider the reduction $g(f(0)) \to_R g(g(a(0)))$. In $E(R)$, $g(f(0))$ can rewrite to $g(g(\diamond))$ or $g(0)$. However we cannot compare the trees $\mathsf{tree}(g(f(0)))$ and $\mathsf{tree}(g(g(a(0))))$ since $\mathsf{tree}(g(f(0))) = (g(f(0)), [])$ and $\mathsf{tree}(g(g(a(0)))) = (g(g(\diamond)), [(0, [])])$, and to conclude that the former is bigger than the later (with respect to $\succ$), it is necessary to have $g(f(0)) \to^+_{E(R)} 0$, which is not necessarily true.

To cope with this problem we will define another relation on $T(\mathcal{F}_\diamond, \mathcal{X})$. This relation, that is denoted by $\gg$, is still closed under substitutions but no longer closed under contexts. Interestingly enough $\gg$ will enable us to have a tree construction closed under contexts.

Definition 4.14 *Let $>$ be a partial relation on $T(\mathcal{F}_\diamond, \mathcal{X})$ closed under contexts and substitutions. We define a relation $\gg$ on $T(\mathcal{F}_\diamond, \mathcal{X})$ as follows: $s \gg t$ iff $s \neq t$ and $s \geq C[t]$, for some context C.*

This relation appeared already in [8]. We note that $C[t] \gg t$, for any non-trivial context C. We have the following result.

Lemma 4.15 *In the conditions of definition 4.14, if $>$ is well-founded then $\gg$ is a partial well-founded order on $T(\mathcal{F}_\diamond, \mathcal{X})$ extending $>$.*

Proof We check transitivity and well-foundedness since irreflexivity follows from well-foundedness. Suppose that $s \gg t$ and $t \gg u$ for some $s, t, u \in T(\mathcal{F}_\diamond, \mathcal{X})$. By definition we have contexts C, D such that $s \geq C[t]$ and $t \geq D[u]$. Since $>$ is closed under contexts and is transitive, we conclude that $s \geq C[D[u]]$. To conclude that $s \gg u$ we still need to see that $s \neq u$. Suppose that is not so, i. e., $s = u$. If one of the contexts C or D is not the trivial context, we must have $s > C[D[s]]$, contradicting well-foundedness of $>$. So if $s = u$, we must have $C = D = \square$, but then $s > t > s$, again contradicting well-foundedness of $>$. Therefore $s \neq u$ and since $s \geq C[D[u]]$, we conclude that $s \gg u$.

Suppose now that $\gg$ is not well-founded, then we have an infinite descending chain

$$s_0 \gg s_1 \gg s_2 \gg \ldots$$

and by definition of $\gg$, for each $i \geq 0$ there is a context C_i such that $s_i \geq C_i[s_{i+1}]$. Again since $>$ is closed under contexts and transitive we conclude that there is an infinite descending chain in $(T(\mathcal{F}_\diamond, \mathcal{X}), >)$ given by

$$s_0 \geq C_0[s_1] \geq C_0[C_1[s_2]] \geq \ldots$$

First note that the sequence constituted by the terms in the sequence above does not contain any infinite subsequence $(t_i)_{i \geq 0}$ such that $t_i = t_{i+1}$, for if it would be so, due to the form of each t_i and to the fact that all terms are finite we would have an index k, such that $C_l = \square$, for any $l \geq k$, and since all terms s_j are different we would have the infinite descending sequence

$$s_k > s_{k+1} > s_{k+2} > \ldots$$

contradicting well-foundedness of $>$.

Finally if $s > t$ then $s \neq t$, since $>$ is well-founded (and therefore irreflexive). By taking C as the empty context in definition 4.14, we conclude that $s \gg t$. □

We now consider the trees over $T(\mathcal{F}_\diamond, \mathcal{X})$ but with the tree lifting associated with $\gg$, i. e., in definition 3.1 we take $>$ to be $\gg$. In order to keep the syntax simple we denote the order in $Tr(T(\mathcal{F}_\diamond, \mathcal{X}))$ also by $\gg$. It should be clear from context whether we are referring to the order on $T(\mathcal{F}_\diamond, \mathcal{X})$ or on $Tr(T(\mathcal{F}_\diamond, \mathcal{X}))$.

The following result is a consequence of lemmas 3.3, 4.13 and 4.15.

Lemma 4.16 *Let $l \to r$ be a rule in R and $\sigma : \mathcal{X} \to T(\mathcal{F}_a, \mathcal{X})$ an arbitrary substitution. Then* $\mathsf{tree}(l\sigma) \gg \mathsf{tree}(r\sigma)$.

Lemma 4.17 *Let $s, t \in T(\mathcal{F}_a, \mathcal{X})$. If $s \to_R t$ then* $\mathsf{cap}(s) \to^{0,1}_{E(R)} \mathsf{cap}(t)$.

Proof We proceed by induction on the definition of reduction. If $s = l\sigma$ and $t = r\sigma$ for some rule $l \to r$ of R and some substitution $\sigma : \mathcal{X} \to T(\mathcal{F}_a, \mathcal{X})$ then (by definition 4.1 and lemma 4.8) $\mathsf{cap}(l\sigma) = \mathsf{cap}(l)\tau = l\tau$, where $\tau : \mathcal{X} \to T(\mathcal{F}_\diamond, \mathcal{X})$ is the substitution given by $\tau(x) = \mathsf{cap}(\sigma(x))$, for all $x \in \mathcal{X}$. Similarly $\mathsf{cap}(r\sigma) = \mathsf{cap}(r)\tau$. Since $l \to \mathsf{cap}(r)$ is a rule in $E(R)$, we have $l\tau \to^1_{E(R)} \mathsf{cap}(r)\tau$, as we had to show.

Suppose $s \to_R t$ and $\mathsf{cap}(s) \to^{0,1}_{E(R)} \mathsf{cap}(t)$.
Let $f(s_1, \ldots, s, \ldots, s_k) \to_R f(s_1, \ldots, t, \ldots, s_k)$, by definition 4.1,

$$\mathsf{cap}(f(s_1, \ldots, s, \ldots, s_k)) = f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(s), \ldots, \mathsf{cap}(s_k))$$

and

$$\mathsf{cap}(f(s_1, \ldots, t, \ldots, s_k)) = f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(t), \ldots, \mathsf{cap}(s_k))$$

Consequently

$$\begin{array}{c} \mathsf{cap}(s) \to^{0,1}_{E(R)} \mathsf{cap}(t) \\ \Downarrow \\ f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(s), \ldots, \mathsf{cap}(s_k)) \to^{0,1}_{E(R)} f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(t), \ldots, \mathsf{cap}(s_k)) \\ \Downarrow \\ \mathsf{cap}(f(s_1, \ldots, s, \ldots, s_k)) \to^{0,1}_{E(R)} \mathsf{cap}(f(s_1, \ldots, t, \ldots, s_k)) \end{array}$$

Now let $a(s_1, \ldots, s, \ldots, s_N) \to_R a(s_1, \ldots, t, \ldots, s_N)$.
By definition 4.1 we have $\mathsf{cap}(a(s_1, \ldots, s, \ldots, s_N)) = \diamond = \mathsf{cap}(a(s_1, \ldots, t, \ldots, s_N))$, so $\mathsf{cap}(a(s_1, \ldots, s, \ldots, s_N)) \to^0_{E(R)} \mathsf{cap}(a(s_1, \ldots, t, \ldots, s_N))$. □

Lemma 4.18 *Let $s, t \in T(\mathcal{F}_a, \mathcal{X})$ such that $s \to_R t$ and* $\mathsf{tree}(s) \gg \mathsf{tree}(t)$. *Then* $\mathsf{tree}(C[s]) \gg \mathsf{tree}(C[t])$, *for any context C.*

Proof We proceed by induction on the context. If C is the trivial context, then the result holds by hypothesis. Let then $C[\ ] = f(s_1, \ldots, \Box, \ldots, s_k)$, with $\Box$ occurring at position j, for some fixed $1 \leq j \leq k$. Then

$$\begin{aligned} \mathsf{tree}(C[s]) = \mathsf{tree}(f(s_1, \ldots, s, \ldots s_k)) &= (\mathsf{cap}(f(s_1, \ldots, s, \ldots s_k)), \bigcup_{i=1}^{k} M_i) \\ &= (f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(s), \ldots \mathsf{cap}(s_k)), \bigcup_{i=1}^{k} M_i) \end{aligned}$$

where $\mathsf{tree}(s_i) = (\mathsf{cap}(s_i), M_i)$ for $1 \leq i \leq k$, $i \neq j$, and $\mathsf{tree}(s) = (\mathsf{cap}(s), M_j)$. Similarly

$$\begin{aligned} \mathsf{tree}(C[t]) = \mathsf{tree}(f(s_1, \ldots, t, \ldots s_k)) &= (\mathsf{cap}(f(s_1, \ldots, t, \ldots s_k)), \bigcup_{i=1}^{k} M'_i) \\ &= (f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(t), \ldots \mathsf{cap}(s_k)), \bigcup_{i=1}^{k} M'_i) \end{aligned}$$

where $M'_i = M_i$, for $1 \leq i \leq k$, $i \neq j$, and $\mathsf{tree}(t) = (\mathsf{cap}(t), M'_j)$.
By hypothesis $\mathsf{tree}(s) \gg \mathsf{tree}(t)$ and therefore either

- $\mathsf{cap}(s) \gg \mathsf{cap}(t)$ and for all $U \in M'_j$ either $\mathsf{tree}(s) \gg U$ or there is an element $V \in M_j$ such that $V \gg U$.
  Since $s \to_R t$ then by lemma 4.17 we have $\mathsf{cap}(s) \to^{0,1}_{E(R)} \mathsf{cap}(t)$, and due to irreflexivity of $\gg$ we indeed have $\mathsf{cap}(s) \to_{E(R)} \mathsf{cap}(t)$. Hence

  $$f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(s), \ldots \mathsf{cap}(s_k)) \to_{E(R)} f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(t), \ldots \mathsf{cap}(s_k))$$

  Since $\to_{E(R)} \subseteq \gg$, we therefore have

  $$(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(s), \ldots \mathsf{cap}(s_k)) \gg f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(t), \ldots \mathsf{cap}(s_k))$$

  To conclude that $\mathsf{tree}(C[s]) \gg \mathsf{tree}(C[t])$ we only need to see that for any $U \in M'_j$ either $\mathsf{tree}(C[s]) \gg U$ or there is an element $V \in \bigcup_{i=1}^{k} M_i$ such that $V \gg U$. Take then $U \in M'_j$ and suppose that there is no such element V. Since $M_j \subseteq \bigcup_{i=1}^{k} M_i$ and $f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(s), \ldots \mathsf{cap}(s_k)) \gg \mathsf{cap}(s)$, we conclude that $\mathsf{tree}(C[s]) \gg \mathsf{tree}(s)$. We must also have $\mathsf{tree}(s) \gg U$, since $\mathsf{tree}(s) \gg \mathsf{tree}(t)$, $\mathsf{cap}(s) \gg \mathsf{cap}(t)$ and $M_j \subseteq \bigcup_{i=1}^{k} M_i$. By transitivity of $\gg$ we conclude that $\mathsf{tree}(C[s]) \gg U$.

- $\mathsf{cap}(s) = \mathsf{cap}(t)$ and $M_j \gg_{mul} M'_j$. In this case we have $\bigcup_{i=1}^{k} M_i \gg_{mul} \bigcup_{i=1}^{k} M'_i$. Since $f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(s), \ldots \mathsf{cap}(s_k)) = f(\mathsf{cap}(s_1), \ldots, \mathsf{cap}(t), \ldots \mathsf{cap}(s_k))$, we conclude that $\mathsf{tree}(C[s]) \gg \mathsf{tree}(C[t])$.

Suppose now that $C[\,] = a(s_1, \ldots, \Box, \ldots, s_N)$, with $\Box$ occurring at position j, for some fixed $1 \leq j \leq N$. Then

$$\mathsf{tree}(C[s]) = \mathsf{tree}(a(s_1, \ldots, s, \ldots s_N)) = (\diamond, \bigcup_{i=1, i\neq j}^{N} [\mathsf{tree}(s_i)] \cup [\mathsf{tree}(s)])$$

and

$$\mathsf{tree}(C[t]) = \mathsf{tree}(a(s_1, \ldots, t, \ldots s_N)) = (\diamond, \bigcup_{i=1, i\neq j}^{N} [\mathsf{tree}(s_i)] \cup [\mathsf{tree}(t)])$$

Since $\mathsf{tree}(s) \gg \mathsf{tree}(t)$ also $\bigcup_{i=1, i\neq j}^{N} [\mathsf{tree}(s_i)] \cup [\mathsf{tree}(s)] \gg_{mul} \bigcup_{i=1, i\neq j}^{N} [\mathsf{tree}(s_i)] \cup [\mathsf{tree}(s)]$ and by definition 3.1 we conclude that $\mathsf{tree}(C[s]) \gg \mathsf{tree}(C[t])$. $\Box$

We have seen that given a TRS R, whenever $E(R)$ terminates, we can lift the well-founded order $\rightarrow^+_{E(R)}$ to a well-founded order $\gg$ on $Tr(T(\mathcal{F}_\diamond, \mathcal{X}))$. Furthermore we can associate to each term $t \in T(\mathcal{F}_a, \mathcal{X})$ a tree in $Tr(T(\mathcal{F}_\diamond, \mathcal{X}))$ in such a way that if $s \rightarrow_R t$ then $\mathsf{tree}(s) \gg \mathsf{tree}(t)$. Consequently the relation $\rightarrow^+_R$ is well-founded. In other words, we have proved our main result.

Theorem 4.19 *If $E(R)$ terminates then R terminates.*

5 Final remarks

As mentioned before, dummy elimination bears similarities with *distribution elimination*, but since the premisses for applying the transformations are different, the two techniques can't really be compared (actually an interesting still open problem is the necessity of the right-linearity requirement on *distribution elimination* in the absence of distribution rules). Still when both techniques are applicable, i. e., the function symbol to be eliminated occurs only in the rhs (no distribution rules are present) and the resulting system is, with respect to *distribution elimination*, right-linear, the techniques seem to be incomparable as the following TRS's show. Consider the TRS R below and suppose we want to eliminate the symbol a.

$$f(f(x)) \rightarrow f(a(f(x)))$$

Then *distribution elimination* results in the system $E_D(R)$:

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

and dummy elimination results in the system $E(R)$:

$$\begin{aligned} f(f(x)) &\rightarrow f(\diamond) \\ f(f(x)) &\rightarrow f(x) \end{aligned}$$

In this case we have that both R and $E(R)$ terminate but $E_D(R)$ does not, suggesting that the transformation E is stronger than E_D.

Now consider the next system R, again with a being the symbol to be eliminated.

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

Distribution elimination results in the system $E_D(R)$:

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

and dummy elimination results in the system $E(R)$:

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

In this case we have that both R and $E_D(R)$ terminate but $E(R)$ does not, suggesting the reverse conclusion. However the transformation associated with dummy elimination seems to be more drastic, since the subterms having as root the symbol to be eliminated are simply chopped off, and rewriting systems are produced that are syntactically simpler that the ones resulting from *distribution elimination*. This idea is enforced by the fact that *distribution elimination* is sound and complete (i. e., termination of R implies termination of $E_D(R)$) with respect to particular kinds of termination like *total termination* [6] and (termination proofs using) *recursive path order (rpo)*, while dummy elimination is not. In the first example above, the original system cannot be proven terminating using *rpo*. Furthermore the system is not simply terminating and thus also not totally terminating. The transformed system $E(R)$ is trivially proven terminating by *rpo* taken over a precedence $\triangleright$, satisfying $f \triangleright \diamond$, and therefore is both simply and totally terminating.

The fact that dummy elimination is not sound with respect to these restricted kinds of termination is not a negative point, on the contrary since it suggests that the transformation is quite strong with respect to syntactical simplification, which is in fact the goal we aim to.

Another interesting point is that we don't need to restrict ourselves to elimination of a single function symbol. If we have more than one function symbol appearing only on the rhs's of rewrite rules, we can easily modify the definitions and statements made in order to get rid of all those symbols simultaneously (we just need a little more of case discrimination).

Finally instead of using the tree lifting order $\succ$ we could have used a version of *rpo* on trees. However *rpo* is too strong for our purposes and the order $\succ$ while being a subset of *rpo*, is still closed under transitivity, contains the multiset extension and still has the same kind of properties as *rpo*, namely monotonicity and well-foundedness preservation.

Dummy elimination could be a useful technique for helping on termination proofs, especially if used in conjunction with automatic tools, since it is very easy to incorporate it as a pre-processing unit to check if the TRS to be proven terminating can be transformed.

References

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

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

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

[4] 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.

[5] Dershowitz, N., and Manna, Z. Proving termination with multiset orderings. *Communications ACM 22*, 8 (1979), 465–476.

[6] Ferreira, M. C. F., and Zantema, H. Total termination of term rewriting. In *Proceedings of the 5th Conference on Rewriting Techniques and Applications* (1993), C. Kirchner, Ed., vol. 690 of *Lecture Notes in Computer Science*, Springer, pp. 213–227. Full version to appear in Applicable Algebra in Engineering, Communication and Computing.

[7] Ferreira, M. C. F., and Zantema, H. Well-foundedness of term orderings. To appear at CTRS 94 (Workshop on Conditional and Typed Term Rewriting Systems).

[8] Kamin, S., and Lévy, J. J. Two generalizations of the recursive path ordering. University of Illinois, 1980.

[9] Zantema, H. Termination of term rewriting by semantic labelling. Tech. Rep. RUU-CS-92-38, Utrecht University, 1992. To appear in Fund. Informaticae.

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

[11] Zantema, H., and Geser, A. A complete characterization of termination of $0^p1^q \rightarrow 1^r0^s$. Tech. Rep. UU-CS-1994-44, University of Utrecht, 1994. Submitted for publication.