INRIA

UNITÉ DE RECHERCHE
INRIA-LORRAINE

Rapports de Recherche

N° 1273

Programme 1
Programmation, Calcul Symbolique
et Intelligence Artificielle

ON RELATIONSHIP BETWEEN TERM REWRITING SYSTEMS AND REGULAR TREE LANGUAGES

G. A. KUCHEROV

Août 1990

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

RR_1273

Une relation entre les systèmes de réécriture de termes et les langages réguliers d'arbres

On relationship between term rewriting systems and regular tree languages

G.A. Kucherov

Résumé

Si P est un ensemble de termes ouverts, $Red(P)$ est l'ensemble des termes qui contiennent au moins un sous-terme qui est une instance d'un terme dans P. Le théorème principal de cet article affirme que si L est un langage régulier de termes clos et P est un ensemble fini de termes ouverts tels que $L \subseteq Red(P)$, alors il existe un ensemble fini P^* tel que tous les termes de P^* sont des instances *linéaires* de termes de P et $L \subseteq Red(P^*)$. L'article montre comment ces résultats peuvent s'appliquer à la réductibilité sur les termes clos des systèmes de réécriture. Ce travail a été effectué lors d'une visite de l'auteur à l'INRIA-Lorraine.

Abstract

If P i a set of open terms, $Red(P)$ is the set of terms that contain at least a subterm which is an instance of a term in P. The main theorem of this paper says that if L is a regular language of ground terms and P is a finite set of open terms such that $L \subseteq Red(P)$, then there exists a finite set P^* such that all the terms of P^* are *linear* instances of terms in P and $L \subseteq Red(P^*)$. Applications of this result to ground reducibility of term rewrite systems are also discussed. This work was done while the author was visiting INRIA-Lorraine.

On relationship between term rewriting systems and regular tree languages

G.A.Kucherov
Institute of Informatics Systems
Siberian Division, USSR Academy of Sciences
Novosibirsk, 630090, USSR

The paper presents a new result on relationship between term rewriting systems (TRSs) and regular tree languages. Important consequences (concerning, in particular, a problem of ground-reducibility) are discussed.

1. Introduction.

It is well known that in many applications of term rewriting technique non-left-linear TRSs (i.e. those with possibly multiple occurence of a variable in the left-hand side of a rule) are essentially more difficult to deal with than the left-linear ones (see, e.g., [Der81,GKM83,Huet80]). According to general mathematical intuition this is not very surprising, since in most fields of mathematics non-linear objects (e.g., differential equations, algebraic equations, finite difference schemes, etc.) are principally more complex. But a formal comparative study of a power of linear and non-linear objects as well as a problem of "linear approximation" is always of great interest.

Obviously, a comparison between left-linear and non-left-linear TRSs may have different purposes and, respectively, may be stated in different ways. For example, in [DC87] an expressive power of "generating" different sets of ground terms is compared ("generating" means reduction of a set of ground terms by a given TRS). This paper deals with another approach where a power of left-linear and non-left-linear TRSs is compared with respect to corresponding sets of reducible ground terms. Within this approach a following question may be propounded: if $\mathcal{R}$ is a non-left-linear TRS and $\mathcal{S}$ is a set of $\mathcal{R}$-reducible ground terms then whether a "weaker" left-linear TRS $\mathcal{L}$ exists s.t. every term in $\mathcal{S}$ is $\mathcal{L}$-reducible.

Let us give the following toy example which illustrates also a "more pragmatic" motivation of the problem. Suppose we want to specify an equality predicate *eq* over natural numbers defined by free

constructors $0,s$. An axiom

$$eq(x,x)\rightarrow true \qquad (1)$$

naturally comes into the mind but it does not form a complet axiomatization of eq since it rests undefined when its arguments ar different. To complete the specification we have to add, e.g., th following three axioms.

$$eq(0,s(x))\rightarrow false \qquad (2)$$

$$eq(s(x),0)\rightarrow false \qquad (3)$$

$$eq(s(x),s(y))\rightarrow eq(x,y) \qquad (4)$$

Apparently, (1)-(4) form a complete specification of eq, i.e. al terms of the set $S=\{eq(M_1,M_2)\,|\,M_1,M_2\in T_F\}$ are reducible. Now we not that axiom (1) and no other axiom is applied, only when both argument of eq are 0. Thus, preserving completeness of the specification, axio (1) can be transformed into $eq(0,0)\rightarrow true$ so that the resultin specification is left-linear. (The resulting specification may b "less effective" but in this paper we are not concerned abou effectiveness but only about principal issues).

In fact, the above example illustrates also a main result of th paper which states that this "equivalent" transformation is possibl to perform whenever S is a *regular tree language*. This principa result has a number of important corollaries, at least two of them ar discussed in the paper. The first one characterizes a class of regula tree languages defined by TRSs. It is known ([GB85]) that left-linea TRSs define regular languages of reducible ground terms, but a question has still to be answered whether this "TRS-definable" subclass of regular tree languages is extended if non-left-linear TRSs are admitted. In this work we give a negative answer to this question. We show that if some (non-left-linear) TRS define a regular tree language, than there exists a left-linear TRS defining the same language.

Another corollary concerns a problem of ground-reducibility of a term for a given TRS [KZ85,JK86]. A term t is $\mathcal{R}$-ground-reducible if all ground instances of t are $\mathcal{R}$-reducible. Ground-reducibility is very important in reasoning in abstract data types [JK86, Com88, Kuc88, Fr86, KNZ87]. As it is stated in [Pla85,KNZ87], ground-reducibility is decidable for any term t and any TRS $\mathcal{R}$. However, while tractable and transparent ground-reducibility testing algorithms are known for left-linear systems (see [Kuc88]), the problem appears to be much more

complicated in the case of non-left-linear TRSs. Original algorithms [Pla85,KNZ87] serve rather for demonstration of decidability than for the practical use. An algorithm of [Com88] is based on a special notion of "conditional grammar" and, basically, on "cleaning algorithm" for such grammars. In [Kou89] a direct algorithm is presented, but from our point of view its description is not clear enough to consider the algorithm practicable. In our opinion, the design of a practicable algorithm for the general case is still on the agenda. Besides, the complexity of the problem also have to be studied. Some complexity measures for related problems are presented in [KNRZ87].

In our work we show that if a linear term is $\mathcal{R}$-ground-reducible then it is $\mathcal{R}^*$-ground-reducible, where $\mathcal{R}^*$ is a left-linear system derived from $\mathcal{R}$ (similarly to the above example). For example, if we have to specify completely a function f, i.e. we should have a term $f(x_1,\ldots,x_n)$ ground-reducible, then we can always get rid of non-left-linear rules. Thus, non-left-linearity is able only to make the computation of f "more effective", and is never inevitable in defining its values. In other words, non-left-linear TRSs are not more expressive than linear ones in defining linear terms completely. Since a set of ground instances of a linear term is regular, this gives another example of the leit-motif of the work: *in specifying a regular set of ground terms non-left-linearity of rules is always possible to avoid.*

2. Preliminaries

2.1. *Term rewriting systems*

We assume the reader to have a minimal knowledge about term rewriting systems (for more details see [HO80]). Let T_F denote a set of ground terms over a signature F. $T_F(\mathcal{X})$ stands for a free algebra over a set of generators $\mathcal{X}$ (=a set of terms over $\mathcal{X}$). In particular, $T_F(X)$ denotes a set of terms with variables, and $var(t)\subseteq X$ is a set of variables in $t\in T_F(X)$. We define a set of occurences $O(t)$ in the usual way as sequences of natural numbers (including the empty sequence, corresponding to the root occurence). If $t\in T_F(\mathcal{X})$, then $\hat{O}(t)\subseteq O(t)$ denotes a subset of occurences of generators in t (for instance, if $t\in T_F(X)$, then $\hat{O}(t)$ is a set of variable occurences in t). For two occurences $u,v\in O(t)$ define $u\leqslant v$ iff u is a prefix of v and $u<v$ iff $u\leqslant v$

and $v \neq u$. *SUB* (respectively *GSUB*) denotes the set of substitutions (respectively ground substitutions).

Given a TRS $\mathcal{R}$ a term $t \in T_F(X)$ is ($\mathcal{R}$-)reducible iff a subterm of t is matched by the left-hand side of a rule in $\mathcal{R}$. A term $t \in T_F(X)$ is ($\mathcal{R}$-)ground-reducible iff for every $\delta \in GSUB$ $\delta(t)$ is ($\mathcal{R}$-)reducible. Throughout the paper we consider only reducibility (ground-reducibility) of terms, but not the results of reductions. Therefore we identify a TRS $\mathcal{R}$ with the set of its left-hand sides and we treat $\mathcal{R}$ as an ordinary set of terms. $Gr(\mathcal{R})$ stands for a set of ground instances of terms of $\mathcal{R}$ and $Red(\mathcal{R})$ stands for a set of $\mathcal{R}$-reducible ground terms.

For a term $t \in T_F(X)$ a term $\sigma(t)$ is the result of superposition (a superposant) of a term s into t at $u \in O(t) \setminus \hat{O}(t)$, if σ is a most general unifier of s and t/u (σ is called a superposition substitution).

2.2 Regular tree languages

Regular tree languages [GS84] are a natural generalization of "classical" regular languages (of words over finite alphabet). All basic results about regular languages have their counterparts in the tree case. In particular, there are three main presentations of regular tree languages:

- tree automata;
- regular tree grammars;
- regular expressions;

In what follows we essentially use the first two of them. We always consider ground terms with the natural tree structure as language objects. In the rest of the section we give the definitions of tree automata and regular grammars for ground terms languages.

A *bottom-up tree automaton* is a quadruple $\mathcal{A}=(Q,F,Q_{fin},P)$, where Q is a finite set of states, F a signature, $Q_{fin} \subseteq Q$, and P is a finite set of transition rules

$$f(q_1,\ldots,q_n) \to q_0, \text{ where } f \in F_n,\ n \geq 1,\ \{q_0,\ldots,q_n\} \subseteq Q,$$

or $$a \to q_0, \text{ where } a \in F_0,\ q_0 \in Q.$$

Informally, given a ground term M an automaton attempts to visit all the nodes of M moving from leaves towards the root. A node can be visited if all its immediate descendants have been visited.

An automaton $\mathcal{A}$ defines a (multi-valued, partial) function $\psi_{\mathcal{A}}: T_F \to Q$ defined recursively as follows.

- if $M=a\in F_0$ and P contains a rule $a\to q_0$, then $\psi_{\mathcal{A}}(M)=q_0$,
- if $M=f(M_1,\ldots,M_n)$, $f\in F_n$, for every $i\in(1,n)$ $\psi_{\mathcal{A}}(M_i)=q_i$, and P contains a rule $f(q_1,\ldots,q_n)\to q_0$, then $\psi_{\mathcal{A}}(M)=q_0$.

An automaton $\mathcal{A}$ is said to accept a term M, if $\psi(M)\in Q_{fin}$. An automaton $\mathcal{A}$ defines a regular language of terms accepted by $\mathcal{A}$. An automaton $\mathcal{A}$ is called

- *deterministic*, if there are no two transition rules in P with the same left-hand sides,
- *completely defined*, if for every $f\in F_n$ and every $q_1,\ldots,q_n\in Q$ there exists a transition rule with the left-hand side $f(q_1,\ldots,q_n)$,
- *equivalent to an automaton* $\mathcal{B}$, if $\mathcal{A}$ and $\mathcal{B}$ define the same language,
- *minimal*, if there is no equivalent automaton with the less number of states.

It is known [Dau83] that every bottom-up tree automaton $\mathcal{A}$ has the unique minimal completely defined deterministic automaton $\mathcal{B}$, equivalent to $\mathcal{A}$. Obviously, for such an automaton $\mathcal{B}$ a function $\psi_{\mathcal{B}}$ is single-valued and total.

A *regular tree grammar* is a quadruple $\Phi=(N,F,r,R)$, where N is a finite set of nonterminal symbols, F a signature, $r\in N$ an axiom, and R a finite set of deduction rules

$$p\to t, \text{ where } p\in N,\ t\in T_F(N).$$

Terms of $T_F(N)$ (note that $T_F\subseteq T_F(N)$) are called *sentential forms*. A sentential form s_2 is derived (at one step) from a sentential form s_1, if there exists a deduction rule $p\to t$ and $u\in\hat{O}(s_1)$ such that $s_1/u=p\in N$ and $s_2=s_1[u\leftarrow t]$. A language generated by a grammar $\Phi=(N,F,r,R)$ is a set of ground terms derivable from r at an arbitrary number of steps.

Regular grammars and tree automata are two different ways for defining the class of all regular tree languages.

3. Basic results

This section gives a summary of the results of [LM87] playing an important role in our paper. Besides that these results are explicitly used below in the proof of theorem 5.1, there is a deep analogy between them and the results of this paper. Also, we introduce two partial order relations on TRSs and we state some of their properties. These relations seem to be useful and allow our results to

have a compact presentation.

Definition 3.1. For $\mathcal{R}_1,\mathcal{R}_2 \subseteq T_F(X)$ the relation $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2$ holds if $Gr(\mathcal{R}_1) \subseteq Gr(\mathcal{R}_2)$.

Obviously, this relation is a partial order and we define $\mathcal{R}_1 \sim \mathcal{R}$ iff $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2$ and $\mathcal{R}_2 \trianglelefteq \mathcal{R}_1$.

For convenience' sake we write "a linear set" as a synonym for "a set of linear terms".

Definition 3.2. A linear set $\mathcal{R}^*$ is called a *linear instantiation* of a set $\mathcal{R}$ iff every term in $\mathcal{R}^*$ is an instance of a term in $\mathcal{R}$.

In other words, for every $t \in \mathcal{R}^*$ there exists $s \in \mathcal{R}$ s.t. $\sigma(s)=t$ for some $\sigma \in SUB$, and σ assigns a ground term to every non-linear variable of s. Obviously, definition 3.2 implies $\mathcal{R}^* \trianglelefteq \mathcal{R}$.

By means of these definitions we can summarize the results of [LM87] as follows.

Theorem 3.1. 1. The relation $\trianglelefteq$ induces a structure of boolean algebra on the class of finite linear sets. It means that for any finite linear set $\mathcal{R}$ there exists a finite linear set $\bar{\mathcal{R}}$ s.t.

$$\overline{Gr(\mathcal{R})} = Gr(\bar{\mathcal{R}}), \qquad (*)$$

where $\overline{Gr(\mathcal{R})} = T_F \setminus Gr(\mathcal{R})$.

2. For any finite set $\mathcal{R}$ there exists a finite set $\bar{\mathcal{R}}$ satisfying (*), iff there exists a finite set $\mathcal{R}'$ s.t. $\mathcal{R} \sim \mathcal{R}'$. Without loss of generality we can always assume $\mathcal{R}'$ to be a linear instantiation of $\mathcal{R}$.

3. There is an algorithm which tests for any finite set $\mathcal{R}$ the existence of a finite $\bar{\mathcal{R}}$ and compute $\bar{\mathcal{R}}$ when it exists.

Theorem 3.1 distinguishes between two kinds of finite sets depending on whether they are equivalent modulo $\sim$ to a linear one or not. Linear sets (together with those equivalent to them) form a boolean algebra w.r.t boolean operations induced by $\trianglelefteq$, while the others form only a lattice.

The following corollary is useful for our presentation.

Corollary. Suppose $\mathcal{R}_1,\mathcal{R}_2$ are finite sets , $\mathcal{R}_1$ is linear and $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2$. Then there exists a linear instantiation $\mathcal{R}_2^*$ of $\mathcal{R}_2$ s.t. $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2^*$.

Proof. Suppose $\mathcal{R}_3$ is a set of all possible unificants between terms of $\mathcal{R}_1$ and terms of $\mathcal{R}_2$. Since $\mathcal{R}_1 \trianglelefteq \mathcal{R}_2$ then $\mathcal{R}_3 \sim \mathcal{R}_1$, and according to part 2 of theorem 3.1 there exists a linear instantiation $\mathcal{R}_3^*$ s.t. $\mathcal{R}_3 \sim \mathcal{R}_3^*$. Every term of $\mathcal{R}_3$ is an instance of a term of $\mathcal{R}_2$ and, hence, $\mathcal{R}_3^*$ is a desired linear instantiation.■

To be completely precise, theorem 3.1 is a variation of [LM87].

Instead of "complement" $\bar{\mathcal{R}}$, the paper considers existence of a finit "difference" $\mathcal{R}_1\backslash\mathcal{R}_2$ for finite $\mathcal{R}_1,\mathcal{R}_2$ ($\mathcal{R}_3=\mathcal{R}_1\backslash\mathcal{R}_2$, if $Gr(\mathcal{R}_3)=Gr(\mathcal{R}_1)\backslash Gr(\mathcal{R}_2)$). Clearly, computing $\mathcal{R}_1\backslash\mathcal{R}_2$ reduces easily t computing $\{t\}\backslash\{\sigma_1(t),\ldots,\sigma_k(t)\}$ for different terms t. A substitutio σ_i is called non-restricting (as in [Fr86]), if for every $x\in var(t$ $\sigma_i(x)$ is a linear term and for every $y\in var(t)$ $x\neq y$ implie $\sigma_i(x)\cap\sigma_i(y)=\emptyset$. Theorem 3.1 can be easily extended to the case o differences, in this case linear sets are generalized to the set $\{\delta_1(t),\ldots,\delta_m(t)\}$, where substitutions $\delta_1,\ldots,\delta_m$ are non-restricting E.g., part 2 of the theorem can be reformulated as follows.

Theorem 3.2.[LM87] For any term t there exists a finit difference $\{t\}\backslash\{\sigma_1(t),\ldots,\sigma_k(t)\}$ iff there exists a se $\{\delta_1(t),\ldots,\delta_m(t)\}$ s.t. all substitutions $\delta_1,\ldots,\delta_m$ are non-restrictin and $\{\sigma_1(t),\ldots,\sigma_k(t)\}\sim\{\delta_1(t),\ldots,\delta_m(t)\}$. Without loss of generality w can always assume $\{\delta_1(t),\ldots,\delta_m(t)\}$ to be a linear instantiation o $\{\sigma_1(t),\ldots,\sigma_k(t)\}$.

Now we introduce another partial order relation on TRSs.

Definition 3.3. For TRSs $\mathcal{R}_1,\mathcal{R}_2$ the relation $\mathcal{R}_1\sqsubset\mathcal{R}_2$ holds if $Red(\mathcal{R}_1)\subseteq Red(\mathcal{R}_2)$.

Obviously, $\mathcal{R}_1\trianglelefteq\mathcal{R}_2$ implies $\mathcal{R}_1\sqsubset\mathcal{R}_2$. We again define a correspondin equivalence: $\mathcal{R}_1\cong\mathcal{R}_2$ iff $\mathcal{R}_1\sqsubset\mathcal{R}_2$ and $\mathcal{R}_2\sqsubset\mathcal{R}_1$.

Consider now algebraic properties induced by the relation $\sqsubset$. While a l.u.b. $\mathcal{R}_1\vee\mathcal{R}_2$ exists for any $\mathcal{R}_1,\mathcal{R}_2$ (we can simply set u $\mathcal{R}_1\vee\mathcal{R}_2=\mathcal{R}_1\cup\mathcal{R}_2$ since $Red(\mathcal{R}_1\cup\mathcal{R}_2)=Red(\mathcal{R}_1)\cup Red(\mathcal{R}_2)$), a g.l.b. $\mathcal{R}_1\wedge\mathcal{R}_2$ does no generally exist even for linear $\mathcal{R}_1$ and $\mathcal{R}_2$ (i.e. there does not exist a finite $\mathcal{R}_3$ s.t. $Red(\mathcal{R}_3)=Red(\mathcal{R}_1)\cap Red(\mathcal{R}_2)$). To show this, we give the following counterexample. Let F consist of two unary functions g,h and two constants a,b. Suppose $\mathcal{R}_1=\{g(x)\}$, $\mathcal{R}_2=\{a\}$. A set T_F can be represented as $\{g,h\}^*\{a,b\}$. By definition $\mathcal{R}_1\wedge\mathcal{R}_2$ defines a set of terms reducible by both $\mathcal{R}_1$ and $\mathcal{R}_2$. Clearly, $Red(\mathcal{R}_1)\cap Red(\mathcal{R}_2)=\{g,h\}^*g\{g,h\}^*a$. Now we show that there does not exist $\mathcal{R}_3$ s.t. $Red(\mathcal{R}_3)=Red(\mathcal{R}_1)\cap Red(\mathcal{R}_2)$. All terms of $\mathcal{R}_3$ must be ground and end with a, otherwise there exists a $\mathcal{R}_3$-reducible term which ends with b and therefore does not belong to $Red(\mathcal{R}_1)\cap Red(\mathcal{R}_2)$. Hence, $\mathcal{R}_3$ cannot be finite, because it must contain, e.g., an infinite set gh^na. Thus, this is only an upper semilattice, which is induced by $\sqsubset$ on sets of terms.

Finally we note that some results in the following sections can

be interpreted as lifting of theorems 3.1,3.2 from the level o: relation $\sqsupset$ to that of $\sqsubset$.

4. TRSs and regular tree languages

A fundamental result about relationship between linear TRSs an regular tree languages is due to [GB85].

Theorem 4.1.[GB85] If $\mathcal{R}$ is a linear TRS then *Red($\mathcal{R}$)* is a regula tree language.

Obviously, the theorem can be extended to those non-linear set which are equivalent modulo ~ to a linear one (since $\theta_1 \cong \theta_2$ whenever $\theta_1 \sim \theta_2$).

From the other hand, it is known that there exist regula languages of ground terms, which cannot be represented as *Red($\mathcal{R}$)* for any TRS $\mathcal{R}$ [FV88]. (A trivial example [Com89a] is the language of even numbers $\{s^{2n}(0) \mid n \geq 0\}$. A term 0 belongs to the language, but if 0 is reducible then any term $s^k(0)$ is reducible.)

However, the following question arises. Do left-linear TRSs generate all regular tree languages representable as *Red($\mathcal{R}$)*? In other words, does a non-linear set $\mathcal{R}$ exist s.t. *Red($\mathcal{R}$)* is a regular tree language and for any linear set $\mathcal{L}$ *Red($\mathcal{L}$)*$\neq$*Red($\mathcal{R}$)*? In this section we answer this question positively.

We start with technical lemmas.

Lemma 4.1. Let $\alpha=\langle\alpha^1,\alpha^2\rangle \in (T_F(X))^2$ be a pair of terms and let $\{M_l\}_{l=1}^{\infty}$, where $M_l=\langle M_l^1, M_l^2\rangle \in (T_F)^2$, be an infinite sequence of pairs of ground terms s.t. for any l M_l is not an instance of α. Then there exists an infinite subsequence of indexes $j_1<j_2<j_3<\ldots$ s.t. for any j_k, j_l a pair $\langle M^1_{j_k}, M^2_{j_l}\rangle$ is not an instance of α.

Proof. We proceed by case analysis.

Case 1. If there exists an infinite sequence $M_{j_1}, M_{j_2}, \ldots$ s.t. either for every j_k $M^1_{j_k}$ is not matched by α^1 or for every j_k $M^2_{j_k}$ is not matched by α^2 then this is a desired sequence.

Case 2. Otherwise without loss of generality assume that for every j $M^1_j=\delta_j(\alpha_1)$ and $M^2_j=\gamma_j(\alpha_2)$ for some $\delta_j, \gamma_j \in GSUB$. M_j is not an instance of α, and therefore this may be the case only when α^1 and α^2 have a common variable and for every j subterms of M^1_j and M^2_j, corresponding to this variable, are different. Since there is only a

finite number of variables in α, we assume again that the latter is true for all M_j and for a single variable $z \in var(\alpha^1) \cap var(\alpha^2)$.

Now consider an infinite sequence $\{N_l\}_{l=1}^{\infty}$, where $N_l=\langle N_l^1, N_l^2\rangle$, and $N_l^1=\delta_l(z)$, $N_l^2=\gamma_l(z)$. Suppose $N_l^1 \neq N_l^2$ for any l. To complete the proof we show that an infinite subsequence $j_1<j_2<j_3<\ldots$ exists s.t. for any j_k, j_l $N_{j_k} \neq N_{j_l}$. Clearly, for such a subsequence the sequence $M_{j_1}, M_{j_2}, M_{j_3}, \ldots$ satisfies the theorem.

Case 2.1. If an infinite subsequence $N_{j_1}, N_{j_2}, N_{j_3}, \ldots$ exists s.t. either the first or the second component is constant for all N_{j_l}, then this is a desired subsequence.

Case 2.2. Otherwise for any N_k there exists only a finite number of N_l s.t. either $N_k^1=N_l^1$ or $N_k^2=N_l^2$. We produce the desired subsequence by the following recursive procedure.

Consider N_1 and remove from the sequence $N_2, N_3, \ldots$ those N_l for which $N_l^1=N_1^2$ or $N_l^2=N_1^1$. Place N_1 into the resulting sequence. Since the finite number of N_l has been removed, then the remaining subsequence is infinite, and the procedure is applied recursively to it. The subsequence which results from an infinite application of the procedure satisfies the theorem, since for any j_k, j_l $N_{j_k}^1 \neq N_{j_l}^2$, and hence $\langle M_{j_k}^1, M_{j_l}^2\rangle$ is not an instance of α.■

Lemma 4.2. Let $\{\alpha_1, \ldots, \alpha_K\} \subseteq (T_F(X))^N$ be a finite set of N-tuples of terms and $\{M_l\}_{l=1}^{\infty}$, where $M_l=\langle M_l^1, \ldots, M_l^N\rangle \in (T_F)^N$, be an infinite number of N-tuples of ground terms, and for any $l, 1 \leqslant j \leqslant K$ M_l is not an instance of α_j. Suppose i_1, i_2 s.t. $1 \leqslant i_1 < i_2 \leqslant N$ and let M_{kl} denote a tuple derived from M_k either by replacing $M_k^{i_1}$ by $M_l^{i_1}$ or by replacing $M_k^{i_2}$ by $M_l^{i_2}$. Then there exist infinitely many pairs k, l s.t. $k \neq l$ and M_{kl} is not an example of α_j for any j.

Proof. We show that there exists an infinite subsequence $j_1<j_2<\ldots$ s.t. for any j_k, j_l $M_{j_k j_l}$ satisfies the desired property. Using induction over K, we assume that $K=1$, i.e. there is a single tuple $\alpha=\langle \alpha_1, \ldots, \alpha_N\rangle$ (this corresponds to the base case as well as the induction step).

If M is a N-tuple, then denote $(M)^{-i}$ a $(N-1)$-tuple derived from M

by deleting the i-th component.

If there exists an infinite subsequence $j_1<j_2<\ldots$ s.t. $(M_{j_l})^{-i_1}$ (or $(M_{j_l})^{-i_2}$) is not an instance of $(\alpha)^{-i_1}$ (respectively $(\alpha)^{-i_2}$), then this is a desired subsequence.

If such a sequence does not exist, then assume without loss of generality that for any j a pair $\langle M_j^{i_1}, M_j^{i_2}\rangle$ is not an instance of $\langle \alpha^{i_1}, \alpha^{i_2}\rangle$. Then by lemma 4.1 a subsequence $j_1<j_2<\ldots$ exists s.t. for any j_k, j_l $\langle M_{j_k}^{i_1}, M_{j_l}^{i_2}\rangle$ is not an instance of $\langle \alpha^{i_1}, \alpha^{i_2}\rangle$. Thus, this is a desired sequence.■

Suppose Lan is a regular tree language, $\mathcal{R}$ a TRS, and $Lan\subseteq Red(\mathcal{R})$, i.e. every term $M\in Lan$ is $\mathcal{R}$-reducible. Let us split Lan into two sublanguages as follows. $Lan=Lan^0\uplus Lan^1$, where Lan^0 consists of $M\in Lan$, reducible only at root (and irreducible at any $u\neq\varepsilon$), and Lan^1 consists of $M\in Lan$, reducible at some $u\neq\varepsilon$. Obviously, $Lan^0\subseteq Gr(\mathcal{R})$. The following lemma says that Lan^0 can be "approximated" with a linear instantiation of $\mathcal{R}$.

Lemma 4.3. Let $\mathcal{R}$, Lan, Lan^0, Lan^1 be defined as above (in particular, $Lan^0\subseteq Gr(\mathcal{R})$). Then there exists a linear instantiation $\mathcal{R}^*$ s.t. $Lan^0\subseteq Gr(\mathcal{R}^*)$.

Proof. Our goal is to build a linear instantiation $\mathcal{R}^*$ which satisfies the lemma.

If $\mathcal{R}$ is linear, then $\mathcal{R}^*=\mathcal{R}$. Otherwise consider a non-linear term $p\in\mathcal{R}$ and suppose $y\in var(p)$ occurs twice or more in p. Consider all instances $\delta_1(p),\delta_2(p),\ldots$ s.t.

(a) $\forall i\ \delta_i(p)\in Lan^0$,

(b) $\forall i\ \delta_i(p)\notin Gr(\mathcal{R}\setminus\{p\})$ (i.e. $\delta_i(p)$ is not an instance of any other $t\in\mathcal{R}$).

If there are no such examples, then we delete p from $\mathcal{R}$ keeping up the condition $Lan^0\subseteq Gr(\mathcal{R})$. Then we reapply recursively the procedure. If such examples exist, consider a set $\Delta=\{\delta_1(y),\delta_2(y),\ldots\}$ of ground terms, corresponding to y. If Δ is finite, then we instantiate p substituting for y by terms of Δ. Thus, we are getting rid of a non-linearity (still keeping up $Lan^0\subseteq Gr(\mathcal{R})$), and we reapply again the procedure. Finally, assume that Δ is infinite. In the rest of the

proof we show that this entails a contradiction with the lemma conditions.

Suppose Lan is defined by a completely specified deterministic bottom-up tree automaton $\mathcal{A}$ with a set of states Q. The idea is to construct a term $\hat{M}$ out of the terms $\delta_1(p),\delta_2(p),\ldots$ s.t. $\hat{M}$ is accepted by $\mathcal{A}$, but $\hat{M}$ is irreducible, thus coming to contradiction.

Note that if $M_1,M_2\in T_F$, $M_1\in Lan$, $u\in O(M_1)$ and $\Psi_{\mathcal{A}}(M_1/u)=\Psi_{\mathcal{A}}(M_2)$, then $M_1[u\leftarrow M_2]\in Lan$ (remind that $\Psi_{\mathcal{A}}$ is single-valued and total). Since Δ is infinite, it contains infinitely many terms corresponding to a state $q_0\in Q$. Therefore without loss of generality we assume $\Psi_{\mathcal{A}}(\delta_i(y))=q_0$ for all $\delta_i(y)\in\Delta$. We also assume that all terms of Δ are different.

Suppose $u_1,u_2\in\hat{O}(p)$ s.t. $p/u_1=p/u_2=y$. We assume $\hat{M}$ to have one of the forms $\hat{M}=\delta_i(p)[u_1\leftarrow\delta_j(y)]$ or $\hat{M}=\delta_i(p)[u_1\leftarrow\delta_j(y)]$ for some $i\neq j$. On the one hand, $\hat{M}\in Lan$ (according to the above remark). On the other hand, $\hat{M}$ is not an instance of p (i.e. is not reducible by p at root), since different terms are assigned to y. Finally, $\hat{M}$ is $\mathcal{R}$-irreducible at any $u\in\hat{O}(p)$, since a subterm $\hat{M}/u$ coincide with that of $\delta_i(p)$ (or $\delta_j(p)$). Thus, it rests to show that there exists a term $\hat{M}$ which cannot be reduced at any $u\in O(p)\setminus\hat{O}(p)$ (in this way we need not consider the reduction by p at root which is not possible by construction of $\hat{M}$).

Let us introduce two variables $y',y''\notin var(p)$ and denote $\tilde{p}=p[u_1\leftarrow y'][u_2\leftarrow y'']$. Find all superpositions of terms of $\mathcal{R}$ into $\tilde{p}$ except of the superposition of p into $\tilde{p}$ at root, and suppose $\sigma_1(\tilde{p}),\ldots,\sigma_K(\tilde{p})$ is a set of superposants. Obviously, if $\hat{M}$ is reducible at $u\in O(p)\setminus\hat{O}(p)$, then it is matched by some $\sigma_j(\tilde{p})$. Thus, we have to build (or to prove the existence of) a term $\hat{M}$ which is not matched by any of $\sigma_1(\tilde{p}),\ldots,\sigma_K(\tilde{p})$.

Note that every $\delta_i(p)$ is not matched by any of $\sigma_1(\tilde{p}),\ldots,\sigma_K(\tilde{p})$, for otherwise $\delta_i(p)$ would be reducible in a manner other than by p at root, but this contradicts to (a),(b).

Suppose $var(\tilde{p})=\{x_1,\ldots,x_{i_1},\ldots,x_{i_2},\ldots,x_N\}$, where $x_{i_1}=y'$, $x_{i_2}=y''$. Every superposant $\sigma_j(\tilde{p})$ can be represented as a tuple

$$\alpha_j=\langle\sigma_j(x_1),\ldots,\sigma_j(x_{i_1}),\ldots,\sigma_j(x_{i_2}),\ldots,\sigma_j(x_N)\rangle=$$
$$=\langle\alpha_j^1,\ldots,\alpha_j^{i_1},\ldots,\alpha_j^{i_2},\ldots,\alpha_j^N\rangle.$$

Every $\delta_i(p)$ is also an instance of $\tilde{p}$ and can be represented as a tuple

$$M_i=\langle\delta_i(x_1),\ldots,\delta_i(y),\ldots,\delta_i(y),\ldots,\delta_i(x_N)\rangle=$$

$$=\langle M_i^1,\ldots,M_i^{t_1},\ldots,M_i^{t_2},\ldots,M_i^N\rangle.$$

Since every tuple M_i is not an instance of any α_j, then by lemma 4.2 for some $k\neq l$ there exists a tuple

$$M_{kl}=\langle M_k^1,\ldots,M_k^{t_1},\ldots,M_l^{t_2},\ldots,M_k^N\rangle \text{ or}$$

$$M_{kl}=\langle M_k^1,\ldots,M_l^{t_1},\ldots,M_k^{t_2},\ldots,M_k^N\rangle$$

s.t. M_{kl} is not an instance of any α_j. The tuple M_{kl} defines a desired term $\hat{M}$, which is an instance of $\tilde{p}$ but no $\sigma_j(\tilde{p})$. $\hat{M}$ belongs to *Lan* but is not reducible. This completes the proof.■

Lemma 4.3 allows us to state as its consequence the main result of this section.

Theorem 4.2. Let $\mathcal{R}$ be an arbitrary TRS. If *Red*($\mathcal{R}$) is a regular tree language, then there exists a linear instantiation $\mathcal{R}^*$ s.t. *Red*($\mathcal{R}^*$)=*Red*($\mathcal{R}$).

Proof. Suppose *Lan*=*Red*($\mathcal{R}$). Consider a subset $Lan^0\subseteq Lan$ consisting of the terms, reducible only at root. By lemma 4.3 there exists a linear instantiation $\mathcal{R}^*$ s.t. $Lan^0\subseteq Gr(\mathcal{R}^*)$. Since every term of *Lan* has a subterm belonging to Lan^0 (every reducible term has a reducible subterm which does not contain a proper reducible subterm), then every term of *Lan* is $\mathcal{R}^*$-reducible and, hence, $Lan\subseteq Red(\mathcal{R}^*)$. On the other hand, $\mathcal{R}^*$ is an instantiation of $\mathcal{R}$ and therefore $Red(\mathcal{R}^*)\subseteq Lan$. Thus, $Red(\mathcal{R}^*)=Lan$.■

To sum up, theorem 4.2 shows that a class of regular ground term languages defined by arbitrary TRSs coincides with the one defined by linear TRSs.

5 Main result

Using lemma 4.3, we prove in this section the main result of the work. Similar to the previous section we start with a technical lemma.

Lemma 5.1. Suppose *Lan* is a regular ground term language defined by a regular grammar $\Phi=(N,F,r,P)$, and suppose $\mathcal{L}$ is a linear set. Then a regular language $Lan\cap Gr(\mathcal{L})$ can be defined by a grammar $\Phi'=(N,F,r,P')$ (i.e. a grammar with the same set of nonterminals).

Proof. Given a grammar $\Phi=(N,F,r,P)$ consider a set S of sentential forms s.t.

(a) $\forall s \in S$ s is derivable from r,

(b) $\forall s \in S\ \forall u \in \hat{O}(s)\ |u| \geq depth(\mathcal{L})$,

(c) $\forall s_1, s_2 \in S$ s_2 is not derivable from s_1,

where $|u|$ denotes the length of u, $depth(\mathcal{L})=max\{depth(t) \mid t \in \mathcal{L}\}$, and $depth(t)=max\{|u| \mid u \in O(t)\}$.

Sentential forms, satisfying (a)-(c), can be computed as a result of all possible derivations starting from r, where every derivation step is applied to a nonterminal only if its depth is less than $depth(\mathcal{L})$. Obviously, S is finite.

According to condition (b) and the linearity of $\mathcal{L}$, for every $s \in S$ and every $t \in \mathcal{L}$ exactly one of the following two propositions holds.

1. s is an "instance" of t, i.e. s can be produced out of t by substituting for variables by terms of $T_F(N)$. In this case all ground terms derivable from s are instances of t.

2. s is not "unifiable" with t. In this case no ground term derivable from s is an instance of t.

Separate a subset $S' \subseteq S$ of sentential forms satisfying condition 1. It is clear that a set of ground terms, derivable from S', is exactly $Lan \cap Gr(\mathcal{L})$. Then we replace the rules of P, having r in their left-hand sides, by the rules $r \to s$ for every $s \in S'$. The resulting grammar Φ' defines the language $Lan \cap Gr(\mathcal{L})$ and has the same set of nonterminals.■

Now we are in position to establish the main result.

Theorem 5.1. Suppose Lan is a regular ground term language, and $Lan \subseteq Red(\mathcal{R})$. Then there exists a linear instantiation $\mathcal{R}^*$ s.t. $Lan \subseteq Red(\mathcal{R}^*)$.

Proof. Let $\Phi=(N,F,r,P)$ be a regular tree grammar defining Lan. Instead of Lan we consider a subset $Lan_0 \subseteq Lan$, consisting of the terms which contain no proper subterm belonging to Lan (since $Lan_0 \subseteq Red(\mathcal{R})$ iff $Lan \subseteq Red(\mathcal{R})$, this restriction is correct). Lan_0 is also a regular language, defined by a grammar $\Phi_0=(N,F,r,P_0)$ that can be built from Φ by deleting every deduction rule for which r occurs in its right-hand side.

We proceed by induction on the number of nonterminals in N.

Base case. If $N=\{r\}$, then Lan is finite, for it consists of all right-hand sides of P that have no occurences of r (=are ground terms). For a finite language the theorem trivially holds.

Induction step. Suppose the theorem holds for any language for which the number of nonterminals is less than k, and suppose a cardinality of N equals k. Without loss of generality we assume P_0 to have a single rule having r in the left-hand side. Indeed, if the theorem is true for a single rule $r \to t$, then it can be automatically extended to the case of finitely many such rules by taking a finite union of corresponding linear instantiations.

For every $p \in N$ denote by $Lan_0(p)$ a set of ground terms derivable from p by the grammar Φ_0 (thus, $Lan_0 = Lan_0(r)$). Since r does not occur in the right-hand sides of rules of P_0 and occures in the left-hand side of the only rule $r \to t$, then this rule can be applied only at the first step of the derivations of terms of Lan_0 and cannot be applied otherwise. In particular, the rule $r \to t$ is not applicable in deriving the terms of $Lan_0(p)$ for any $p \in N \setminus \{r\}$. Thus, a language $Lan_0(p)$ is defined by a grammar with the number of nonterminals less than k.

Now consider a rule $r \to t$. Let $\hat{O}(t) = \{u_1, \ldots, u_m\}$ and $p_1, \ldots, p_n \in N$ be nonterminals occuring in t. Substitute at $u_1, \ldots, u_m$ different variables $x_1, \ldots, x_m$ (so that every x_i corresponds to a nonterminal $p_{j_i} \in \{p_1, \ldots, p_n\}$) and let $\tilde{t}$ be a resulting linear term. Every $M \in Lan_0$ is an instance of $\tilde{t}$ and is associated with the tuple $\langle M/u_1, \ldots, M/u_m \rangle$, where M/u_i is an element of the corresponding language $Lan_0(p_{j_i})$. By the theorem condition M is $\mathcal{R}$-reducible, and there are two alternatives:

(a) M is reducible at some $u \in O(\tilde{t}) \setminus \hat{O}(\tilde{t})$, but is not at any $v \geq u_i$ for any $u_i \in \hat{O}(\tilde{t})$, that is, every component of the tuple $\langle M/u_1, \ldots, M/u_m \rangle$ is irreducible,

(b) M is reducible at some $v \geq u_i$ for some $u_i \in \hat{O}(\tilde{t})$, that is, there is a reducible component in $\langle M/u_1, \ldots, M/u_m \rangle$.

Compute all superpositions of $\mathcal{R}$ into t and let $\sigma_1(\tilde{t}), \ldots, \sigma_L(\tilde{t})$ be corresponding superposants. Every term satisfying (a) is an instance of $\sigma_1(\tilde{t}), \ldots, \sigma_L(\tilde{t})$. Note also that while $\sigma_1(\tilde{t}), \ldots, \sigma_L(\tilde{t})$ are $\mathcal{R}$-reducible, $\mathcal{R} \cong \mathcal{R} \cup \{\sigma_1(\tilde{t}), \ldots, \sigma_L(\tilde{t})\}$. Applying lemma 4.3 to Lan and $\mathcal{R} \cup \{\sigma_1(\tilde{t}), \ldots, \sigma_L(\tilde{t})\}$, we conclude that there exists a set $\{\delta_1(\tilde{t}), \ldots, \delta_K(\tilde{t})\}$ s.t. every $\delta_k(\tilde{t})$ is linear, and every term M satisfying (a) is an instance of $\{\delta_1(\tilde{t}), \ldots, \delta_K(\tilde{t})\}$.

Obviously, the language $Lan_0 \cap Gr(\{\delta_1(\tilde{t}), \ldots, \delta_K(\tilde{t})\})$ is reducible by a linear instantiation $\mathcal{R}^*$. Now we show that the language

$Lan_O \setminus Gr(\{\delta_1(\tilde{t}),\ldots,\delta_K(\tilde{t})\})$ is also reducible by a linear instantiation of $\mathcal{R}$.

Let $\{\eta_1(\tilde{t}),\ldots,\eta_J(\tilde{t})\}=\{\tilde{t}\}\setminus\{\delta_1(\tilde{t}),\ldots,\delta_K(\tilde{t})\}$ (see theorem 3.2). Since $Lan_O \cap Gr(\{\delta_1(\tilde{t}),\ldots,\delta_K(\tilde{t})\})$ contains all the terms satisfying (a), then $Lan_O \cap Gr(\{\eta_1(\tilde{t}),\ldots,\eta_J(\tilde{t})\})$ contains only terms, satisfying (b). Thus, for every $M \in Lan_O \cap Gr(\{\eta_1(\tilde{t}),\ldots,\eta_J(\tilde{t})\})$ the tuple $\langle M/u_1,\ldots,M/u_m\rangle$ is reducible. Denote

$H_1=\langle\eta_1(x_1),\ldots,\eta_1(x_m)\rangle,\ldots,H_J=\langle\eta_J(x_1),\ldots,\eta_J(x_m)\rangle$.

If for $M \in Lan_O$ the tuple $\langle M/u_1,\ldots,M/u_m\rangle$ is an instance of one of $H_1,\ldots,H_J$, then there exists a reducible component M/u_i. Without loss of generality we assume that the tuples $H_1,\ldots,H_J$ are pairwise non-unifiable. Then for every $H_k=\langle\eta_k(x_1),\ldots,\eta_k(x_m)\rangle$ there exists a component $\eta_k(x_i)$ s.t. every term $N \in Lan_O(p_{j_i}) \cap Gr(\eta_k(x_i))$ is $\mathcal{R}$-reducible. (Otherwise there exists $M \in Lan_O \cap Gr(\{\eta_1(\tilde{t}),\ldots,\eta_J(\tilde{t})\})$ for which the tuple $\langle M/u_1,\ldots,M/u_m\rangle$ is not reducible). Consider a language $Lan_O(p_{j_i}) \cap Gr(\eta_k(x_i))$. Since $Lan_O(p_{j_i})$ is defined by a grammar with the number of nonterminals less than k, then by lemma 5.1 the language $Lan_O(p_{j_i}) \cap Gr(\eta_k(x_i))$ is defined by a grammar with the same number of nonterminals. By induction hypothesis every term of the language $Lan_O(p_{j_i}) \cap Gr(\eta_k(x_i))$ is reducible by a linear instantiation $\mathcal{R}_k^*$. Thus, all tuples $\langle M/u_1,\ldots,M/u_m\rangle$ that are instances of $H_1,\ldots,H_J$ are reducible by a linear instantiation $\mathcal{R}_1^* \cup \ldots \cup \mathcal{R}_L^*$, i.e., a language $M \in Lan_O \cap Gr(\{\eta_1(\tilde{t}),\ldots,\eta_J(\tilde{t})\})$ is reducible by a linear instantiation of $\mathcal{R}$.■

Thus, we have proved the main result which says that when specifying a regular language of ground terms (so that every term of the language is reducible), non-left-linearity of rules is always possible to avoid.

If $Lan=Red(\mathcal{R})$, we obtain immediately theorem 4.2 of the previous section. If $Lan=Red(\mathcal{L})$ for a linear set $\mathcal{L}$, another important corollary is obtained.

Theorem 5.2. Let $\mathcal{L},\mathcal{R}$ be two sets of terms s.t. $\mathcal{L} \sqsubseteq \mathcal{R}$, and suppose $\mathcal{L}$ is linear. Then there exists a linear instantiation $\mathcal{R}^*$ s.t. $\mathcal{L} \sqsubseteq \mathcal{R}^*$.

Note that the only difference between theorem 5.2 and the corollary of theorem 3.1 is that the relation $\sqsupset$ is replaced by $\sqsubseteq$.

In the following section we show how theorem 5.2 gives rise to important applications to the problem of ground-reducibility.

6. Applications to ground-reducibility

In the introductory example we have built a linear instantiation of a given TRS, preserving ground-reducibility of a term $eq(x,y)$. In this section we answer the question whether such a transformation is always possible. In other words, if $\mathcal{R}$ is a TRS and a term t is $\mathcal{R}$-reducible, then does a linear instantiation $\mathcal{R}^*$ always exist s.t. t is $\mathcal{R}^*$-ground-reducible?

Lemma 6.1. Suppose $\mathcal{L},\mathcal{R}$ are TRSs, Then the following statements are equivalent.

1. $\mathcal{L}\sqsubseteq\mathcal{R}$.
2. for every $t\in\mathcal{L}$ t is $\mathcal{R}$-ground-reducible.

Proof. Obvious.■

Lemma 6.1 together with theorem 5.2 give a positive answer to the question above provided that t is a linear term.

Lemma 6.2. If $\mathcal{R}$ is a TRS, and a term t is linear and $\mathcal{R}$-ground-reducible, then there exists a linear instantiation $\mathcal{R}^*$ s.t. t is $\mathcal{R}^*$-ground-reducible.

Proof. Follows directly from lemma 6.1 and theorem 5.2.■

Now consider the case when t is non-linear. Suppose $t\in\mathcal{T}_F(X)$ and $var(t)=\{x_1,\ldots,x_n\}$. Compute all superpositions of $\mathcal{R}$ into t and suppose $\{\sigma_1(t),\ldots,\sigma_m(t)\}$ is a set of corresponding superposants. Now introduce a new n-ary function symbol h and let $h(x_1,\ldots,x_n)$ denote t. Denote also $\sigma_i(h(x_1,\ldots,x_n))=h(\sigma_i(x_1),\ldots,\sigma_i(x_n))=h(s_{i1},\ldots,s_{in})$. Clearly, t is $\mathcal{R}$-ground-reducible, iff $h(x_1,\ldots,x_n)$ is ground-reducible by $\mathcal{R}\cup\{h(s_{11},\ldots,s_{1n}),\ldots,h(s_{m1},\ldots,s_{mn})\}$. According to lemma 6.2 we conclude that $h(x_1,\ldots,x_n)$ is ground-reducible by a linear instantiation $\mathcal{R}^*\cup\{h(s_{11},\ldots,s_{1n}),\ldots,h(s_{k1},\ldots,s_{kn})\}^*=\mathcal{R}^*\cup\{h(s^*_{11},\ldots,s^*_{1n}),\ldots,h(s^*_{k1},\ldots,s^*_{kn})\}$. Coming back to t, we can see that if $h(s_{j1},\ldots,s_{jn})$ corresponds to a superposition of some $l\in\mathcal{R}$ into t, then $h(s^*_{j1},\ldots,s^*_{jn})$ corresponds to a superposition of some instance $\delta(l)$ into t, where the superposition substitution is non-restricting

(see section 3). Thus, we have proved the following result.

Theorem 6.1. Given a TRS $\mathcal{R}$ suppose t is $\mathcal{R}$-ground-reducible. Then there exists an instantiation $\mathcal{R}^*$ s.t. t is $\mathcal{R}^*$-ground-reducible and every $l \in \mathcal{R}^*$ is linear or superposes into t yielding a non-restricting superposition substitution.

Theorem 6.1 is a generalization of lemma 6.2 to non-linear terms. If we have to specify completely a non-linear term, then we can avoid those non-linearities in left-hand sides generating restricting superposition substitutions. For example, specifying a term $f(x,x,y)$, the left-hand-sides $f(s(x),s(x),s(y))$, $f(0,x,x)$ are non-linear, but do not produce restricting superposition substitutions and cannot generally be instantiated. On the contrary, a left-hand-side $f(s(x),s(y),s(y))$ presents a non-linearity which can always be avoided.

7. Concluding remarks

New results on relationship between term rewriting systems and regular tree languages have been presented. A central result is theorem 5.1 which is fundamental and has several important consequences. One of them (theorem 4.2) is a characterization of a class of term rewriting systems defining a regular set of reducible ground terms. It is shown that this class coincides with that of left-linear rewriting systems modulo equivalence ~. Another corollary (theorem 5.2) has important applications to a problem of ground-reducibility. We have proved (lemma 6.2), for example, that ground-reducibility of a linear term can be always preserved after an instantiation of the rewriting system into a left-linear one.

A possible further development of the work may primarily concern algorithmic aspects. The presented proofs are completely unconstructive and do not allow us to extract any algorithms from them. Meanwhile, a problem of designing such algorithms is very important. E.g., having an algorithm of constructing a linear instantiation for lemma 5.2, we could always reduce a ground-reducibility testing problem to the left-linear case for which practicable decision algorithms are known (see introduction). An algorithm for testing regularity of $Red(\mathcal{R})$ for a given TRS $\mathcal{R}$ is also of great importance. A problem of designing such an algorithm was raised in [Com89b] where an assumption of regularity of $Red(\mathcal{R})$ played

a principal role. However, this algorithm is deeply related to a general ground-reducibility testing algorithm and therefore seems to be of similar complexity. Design and complexity analysis of the abovementioned algorithms may be a subject for the future work.

Most of presented results were obtained during the author's visit to Computer Science Research Center of Nancy. Many thanks must be given to Hubert Comon and Pierre Lescanne for stimulating discussions and practical help.

References

[Com88] Comon H., An effective method for handling initial algebras.-Lect. Notes Comput. Sci., vol.343, 1989, p.108-118.

[Com89a] Comon H., Private communication, October 1989, Nancy.

[Com89b] Comon H., Inductive proofs by specifications transformation.- Lect. Notes Comput. Sci., 1989, vol.355, p.76-91.

[DC87] Dauchet M., De Comite, A gap between linear and non linear term-rewriting systems.- Lect. Notes Comput. Sci., 1987, vol.256, p.95-104.

[Der81] Dershowitz N., Termination of linear term rewriting systems.-Lect. Notes Comput. Sci., 1981, vol.115, p.448-458.

[Fr86] Fribourg L., A strong restriction of the inductive completion procedure, Lect. Notes Comput. Sci., 1986, vol. 226, p.105-115.

[FV88] Fulop Z., Vagyvolgyi S., A characterization of irreducible sets modulo left-linear term-rewriting systems by tree automata.-Lect. Notes Comput. Sci., vol.343, 1989, p.157.

[GB85] Gallier J.H., Book R.V., Reductions in tree replacement systems.- Theoretical Computer Science, 1985, vol.37, p.123-150.

[GKM83] Guttag J.V., Kapur D., Masser D.R., On proving uniform termination and restricted termination of term rewriting systems.-SIAM Journ. Comput., 1983, vol.12, n.1, p.189-214.

[GS84] Gecseg F., Steinby M., Tree automata.-Akademiai Kiado, Budapest, 1984.

[Huet80] Hüet G., Confluent reductions: abstract properties and applications to term rewriting systems.- Journ. ACM, 1980,vol.27, n.4, p.797-821.

[HO80] Huet G., Oppen D.C., Equations and rewrite rules: a survey.- Formal Languages Theory: Perspectives and Open Problems, Academic Press, New York, 1980, p.349-405.

[JK86] Jouannaud J.-P., Kounalis E., Automatic proofs by induction in equational theories without constructors.- Proc. Symp. Logic in Comput. Sci., Cambridge, Mass., June 16-18, 1986, p.358-366.

[KNZ87] Kapur D., Narendran P., Zhang H., On sufficient-completeness and related properties of term rewriting systems.- Acta Informatica, 1987, 24, p.395-415.

[KNRZ87] Kapur D., Narendran P.,Rosenkrantz D.J., Zhang H., Sufficient-Completeness, Quasi-Reducibility and Their Complexity. -TR 87-26, Comp. Sci. Dep., State Univ. of New York at Albany, 1987.

[KZ85] Kounalis E., Zhang H., A general completeness test for equational specifications.- Proc. 4th Hungarian Comput. Sci. Conf., 1985, p.185-200.

[Kou89] Kounalis E., Testing for inductive (oo)-reducibility.- Lect. Notes Comput. Sci., vol.431, 1990, p.221-238.
[Kuc88] Kucherov G.A. A new quasi-reducibility testing algorithm and its application to proofs by induction.-Lect. Notes Comput. Sci., vol.343, 1989, p.204-213.
[LM87] Lassez J.-L., Marriott K., Explicit representation of terms defined by counter examples, Journ. Automated Reasoning, 1987, 3, p.301-318.
[Pla85] Plaisted D.A., Semantic confluence tests and completion methods.- Inf. and Contr., 1985, 65, p.182-215.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399