Proofs and reachability problem for ground rewrite systems [1]

Coquidé J.L.(*) Gilleron R.(**)
(*)L.I.F.L(CNRS URA 369), University of Lille-Flandres-Artois
UFR IEEA, 59655 Villeneuve d'Ascq cedex, France
(**)L.I.F.L(CNRS URA 369), IUT A, departement informatique, BP 179
59653 Villeneuve d'Ascq cedex, France

abstract: The different reachability problems for ground rewrite systems are decidable[OY86], [DEGI89]. We prove these results using ground tree transducers of [DATI85] and wellknown algorithms on recognizable tree languages in order to obtain efficient algorithms. We introduce and study derivation proofs to describe the sequences of rules used to reduce a term t in a term t' for a given ground rewrite system S and sketch how compute a derivation proof in linear time. Moreover, we study the same problem for recognizable tree languages.

1 INTRODUCTION

The reachability problem for rewrite system is to decide whether, given a rewrite system S and two ground terms t and t', t can be reduced in t' with S. It is well-known that this problem is undecidable for general rewrite systems. We study this problem for ground rewrite systems, that is to say finite rewrite systems for which the left-hand sides and right-hand sides of rules in S are ground terms(without variables). The reachability problem for ground rewrite systems was proved decidable by Oyamaguchi in [OY86] and by Dauchet and Tison in [DATI85] as a consequence of decidability of confluence for ground rewrite systems. In [DEGI89], we also study this problem and give a real time decision algorithm, this after the compilation of the ground rewrite system in a ground tree transducer(a software Valeriann is an implementation of these algorithms[DADE89]). More recently, Dauchet and Tison have proved that the theory of ground rewrite systems is decidable, see [DATI90]. In this paper, we consider some other reachability problems. Given two recognizable tree languages F and F', different reachability problems are: Is the set of reductions of terms in F with S included in F'?, is there some term in F which reduces in a term in F'?... We prove these problems decidable using algebraical tools of tree automata(these algorithms are now implemented in Valeriann). As the reachability problems are decidable, a natural question is: If t can be reduced to t' with S, how can we reduce t to t' with S? We introduce the notion of derivation proofs as terms over a new ranked alphabet in order to obtain the "history" of a reduction of t in t' with S. The frontier of a derivation proof gives the sequence of rules used to reduce t in t',

---

[1]This work was in part supported by the "PRC Mathématiques et Informatique" et ESPRIT2 Working Group ASMICS.

moreover, we can find at which occurence a rule is applied. We prove that the set of derivation proofs is a recognizable tree language and give a linear time algorithm for a derivation proof for the reachability problem.

2 PRELIMINARIES

Let Σ be a finite ranked alphabet and T_Σ be the set of terms over Σ.
Let X be a denumerable set of variables, $X_m=\{x_1,...,x_m\}$, $T_\Sigma(X)$ and $T_\Sigma(X_m)$ the set of terms over $\Sigma \cup X$ and $\Sigma \cup X_m$. A context c is a term in $T_\Sigma(X_m)$ such that each variable occurs exactly once in c and we denote $c(t_1,...,t_m)$ the result of the substitution of each x_i by a term t_i.

2.1 Rewrite systems

A rewrite system $S=\{l_i \rightarrow r_i \ / \ l_i,r_i \in T_\Sigma(X),\ i\in I\}$ on T_Σ is a set of pairs of terms in $T_\Sigma(X)$. We only consider the case I finite. $\rightarrow_S$ is the rewrite relation induced by S and $\xrightarrow{*}_S$ the reflexive and transitive closure of $\rightarrow_S$.
A ground rewrite system is such that l_i and r_i are ground terms (without variables). For more developments see [HUOP80] and [DEJO89].

2.2 Tree automata and recognizable tree languages

A bottom-up automaton (or frontier to root) is a quadruple $A=(\Sigma,Q,Q_f,R)$ where Σ is a finite ranked alphabet, Q a finite set of states of arity 0, Q_f a subset of Q, R a finite set of rules of the next configuration:
(i) $f(q_1,...,q_n)\rightarrow q$ with $n\geq 0$, $q_1,...,q_n$ in Q or (ii) $q\rightarrow q'$ with q,q' in Q (ε-rules).
Note we can consider R as a ground rewrite system on $T_{\Sigma\cup Q}$, $\rightarrow_A$ is the rewrite relation $\rightarrow_R$.
The tree language recognized by A is $L(A)=\{t\in T_\Sigma \ / \ t \xrightarrow{*}_A q,\ q\in Q_f\}$.
Let A be a bottom-up automaton, there exists a bottom-up automaton without ε-rules such that L(B)=L(A) and there exists a deterministic(i.e no ε-rules and no two rules with the same left-hand side) bottom-up automaton C such that L(C)=L(A).
A tree language F is recognizable if there exists a bottom-up automaton A such that L(A)=F.
The class of recognizable tree languages is closed under union, intersection and complementation. We can decide if a recognizable tree language is empty, inclusion, equality of recognizable tree languages.
A regular grammar is a quadruple $G=(A,V,\Sigma,R)$ where V is a set of variables of arity 0, V and Σ are disjoint, A belongs to V and R is a finite set of rules, $R=\{l\rightarrow r \ / \ l\in V,\ r\in T_{\Sigma\cup V}\}$. G generates the tree language $L(G)=\{t\in T_\Sigma \ / \ A \xrightarrow{*}_R t\}$. L(G) is a recognizable tree language. For more developments see [GEST84]

2.3 Reachability problems

Let S be a rewrite system, the first order reachability problem is: Given two terms t and t', can we reduce t to t' by S. Different second order reachability problems are: Given two recognizable tree languages F and F', is the set of reductions of terms in F by S included in F' or is there some term in F which reduces to a term in F' by S,...

2.4 Ground tree transducers

A ground tree transducer (gtt) is a pair V=(A,B) of bottom-up automata. Let $A=(\Sigma,Q_A,Q_{Af},R_A)$ and $B=(\Sigma,Q_B,Q_{Bf},R_B)$. The tree transformation induced by V is the set $r(V)=\{(t,t') \ / \ t\in T_\Sigma, t'\in T_\Sigma, \exists s\in T_{\Sigma\cup(Q_A\cap Q_B)}$ such that $t \xrightarrow{*}_A s \ _B\xleftarrow{*} t'\}$ or $r(V)=\{(t,t') \ / \ t=c(t_1,\ldots,t_n) \xrightarrow{*}_A s=c(i_1,\ldots,i_n) \ _B\xleftarrow{*} t'=c(t'_1,\ldots,t'_n)$, $t_j,t'_j\in T_\Sigma$, $i_j\in Q_A\cap Q_B\}$.

Note that we can choose $Q_{Af} = Q_{Bf} = \varnothing$. The states in $Q_A\cap Q_B$ are called interface states.

The class of the tree transformations induced by ground tree transducers is closed by inverse, composition and iteration. For union, the next property holds: Let $V_1=(A_1,B_1)$ and $V_2=(A_2,B_2)$ be two ground tree transducers with disjoint sets of states, V be the ground tree transducer obtained by union of states and union of rules, we have $r(V)^*=(r(V_1)\cup r(V_2))^*$. The proofs can be found in [DATI85] and [DHLT87], see also[FUVA90].

3 REACHABILITY PROBLEM FOR GROUND REWRITE SYSTEMS

3.1 Ground rewrite systems and ground tree transducers

Proposition. For each ground rewrite system S, there exists a ground tree transducer V such that $\xrightarrow{*}_S = r(V)$.

A first proof of this proposition is in [DATI85], a polynomial time algorithm to compute V from S is in [DEGI89].We first compute a ground tree transducer U such that $\rightarrow_S = r(U)$ and then compute a ground tree transducer V such that $r(V) = (r(U))^* = \xrightarrow{*}_S$. An example is in figure 1.

3.2 Ground rewrite systems and recognizable tree languages

Proposition. Let S be a ground rewrite system and F a recognizable tree language, the set $[F]_S$ of reductions of terms in F by S is recognizable.

Proof. Let V=(A,B) be the ground tree transducer such that $r(V)= \xrightarrow{*}_S$. So we have $[F]_S =\{(t,t') \ / \ t \xrightarrow{*}_S t'$ and $t\in F\}=\{(t,t') \ / \ (t,t')\in r(V)$ and $t\in F\}$ and so $[F]_S=\{(t,t')/F\ni t=c(t_1,\ldots,t_n) \xrightarrow{*}_A s=c(i_1,\ldots,i_n) \ _B\xleftarrow{*} t'=c(t'_1,\ldots,t'_n)$, $t_j,t'_j\in T_\Sigma$, $i_j\in Q_A\cap Q_B\}$

Let $A=(\Sigma,Q_A,\varnothing,R_A)$, $B=(\Sigma,Q_B,\varnothing,R_B)$ and $M=(\Sigma,Q_M,Q_{Mf},R_M)$ a deterministic bottom-up automaton such that $L(M)=F$. We suppose Q_B and Q_M are disjoint set of states.
We define a bottom-up automaton $C=(\Sigma,Q_C,Q_{Cf},R_C)$ with $Q_C=Q_B\cup Q_M$, $Q_{Cf}=Q_{Mf}$, $R_C=R_B\cup R_M\cup R'$ where $R'=\{i\to q/\ i\in Q_A\cap Q_B,\ q\in Q_M,\ \exists u\in T_\Sigma$ s.t $u \xrightarrow{*}_A i$ and $u\xrightarrow{*}_M q\}$. As the sets $\{s\ /\ s\xrightarrow{*}_A i\}$ and $\{s\ /\ s\xrightarrow{*}_M q\}$ are recognizable, we can construct R'. We have $L(C)=[F]_S$. The complete proof is with a double inclusion.We just point out how we can recognize a term t' in $[F]_S$ with C. We first use rules of B so $t'=c(t'_1,..,t'_n)\xrightarrow{*}_B c(i_1,..,i_n)$, then we use rules in R' then we have $c(i_1,..,i_n)\xrightarrow{*}_{R'} c(q_1,...,q_n)$ and using rules of M, $c(q_1,...,q_n)\xrightarrow{*}_M q\in Q_{Mf}=Q_{Cf}$.

3.3 Reachability problem for ground rewrite systems

Proposition. The first-order reachability problem for ground rewrite systems is decidable.
Proof. Let t be a term of T_Σ, the set $[t]_S$ of reductions of t by S is recognizable, so we can decide if t' is in $[t]_S$.
Proposition. The different second-order reachability problems for ground rewrite systems are decidable
Proof. Let F and F' two recognizable tree languages, the set $[F]_S$ of reductions of terms in F by S is recognizable so the different second order reachability problems for ground rewrite systems are decidable, for example, we can decide if $[F]_S$ is included in F', if the intersection of $[F]_S$ and F' is not empty,..., this, using classical algorithms for recognizable tree languages.

figure 1

Exemple : Let $\Sigma=\{a,b,c,f,g\}$ avec a,b,c,f,g with arity 0,0,0,1,1
Let $R=\{1: a\to f(b);\ 2: b\to g(c);\ 3: f(g(c))\to g(a)\}$ ground rewrite system

$G_1: a\to i_1$	$D_1: b\to q'_1\ ,\ f(q'_1)\to i_1$
$G_2: b\to i_2$	$D_2: c\to q'_2\ ,\ g(q'_2)\to i_2$
$G_3: c\to q_1\ ,\ g(q_1)\to q_2\ ,\ f(q_2)\to i_3$	$D_3: a\to q'_3\ ,\ g(q'_3)\to i_3$

We first construct U=(G,D) with $R_G=R_{G1}\cup R_{G2}\cup R_{G3}$ and $R_D=R_{D1}\cup R_{D2}\cup R_{D3}$
In order to have the ground tree transducer V such that $r(V)=(r(U))^*= \xrightarrow{*}_R$, we add new rules to the bottom-up automaton G and D.

We obtain E={
$q'_3\to i_1$ $(a\to_G i_1\ ,\ a\to_D q'_3)$
$q'_1\to i_2$ $(b\to_G i_2\ ,\ b\to_D q'_1)$
$q'_2\to q_1$ $(c\to_G q_1\ ,\ c\to_D q'_2)$
$i_2\to q_2$ $(g(q_1)\to_G q_2\ ,\ g(q'_2)\to_D i_2\ ,\ q'_2\to_E q_1)$
$i_1\to i_3$ $(f(q_2)\to_G i_3\ ,\ f(q'_1)\to_D i_1\ ,\ q'_1\to_E i_2\to_E q_2)$ }

Let V=(A,B) with $R_A=R_G\cup\{i_2\to q_2\ ,\ i_1\to i_3\}$ and $R_B=R_D\cup\{i_1\to q'_3\ ,\ i_2\to q'_1,\ i_3\to i_1\}$
With R , $a\xrightarrow{*}_R g(a)$
With U, $a\xrightarrow{*}_G i_1\ {}_D\xleftarrow{*} f(b)\xrightarrow{*}_G f(i_2)\ {}_D\xleftarrow{*} f(g(c))\xrightarrow{*}_G i_3\ {}_D\xleftarrow{*} g(a)$
With V , $a\xrightarrow{*}_A i_3\ {}_B\xleftarrow{*} g(a)$.
Note that the new rules suppress the generation-erasing steps for U

Note that the proposition in the section 3.2 is a consequence of results in [DATI85], but, here we have a direct proof and construct a bottom-up automaton to recognize $[F]_S$ in order to obtain more efficient algorithms. Note also that in [DEGI89], we study the reachability problem for ground rewrite systems modulo some sets of equations (commutativity, associativity, commutativity and associativity). We have got some new results for ground rewrite systems modulo ACI(associativity, commutativity and idempotency), that is to say: with restriction on the configuration of terms, the first order reachability problem is decidable and if we can apply the rules ACI on a finite set of terms, then, recognizability is preserved. See [GI90].

4 DERIVATION PROOFS FOR THE REACHABILITY PROBLEM

4.1 Introduction and notations

Let S be a ground rewrite system, t and t' two ground terms, we can decide if t reduces to t' with S, our aim is now to find a or all the possible derivations of t in t' with S. We define derivation proofs as terms in T_Δ where Δ is a new ranked alphabet such that the frontier of a term gives the sequence of rules used to reduce t in t' with S. See the figure 2 and figure 3 for examples. We only consider standard ground rewrite systems as defined further. In Section 4.2, we define the recognizable tree language of derivation proofs for the first order reachability problem for (t,t',S), in Section 4.3, we sketch the construction for the second order reachability problem for (F,F',S) and in Section 4.4, the construction of a derivation proof for (t,t',S) in linear time. In Section 4.5, we consider the case of general ground rewrite systems. We denote r(b) the arity of a symbol b in Σ, hg(t) the height of a term t and $\Phi(t)$ the frontier of a term t.

$S=\{l_i \rightarrow r_i \,/ i\in I\}$ is a ground rewrite system of standard rules, that is to say we have three kind of rules:

(i) Erasing rules: $f(a_1,\ldots,a_n)\rightarrow a$, $f\in \Sigma_n, a_1,\ldots,a_n, a\in \Sigma_0$

(ii) Generating rules: $a\rightarrow f(a_1,\ldots,a_n)$, $f\in \Sigma_n, a_1,\ldots,a_n, a\in \Sigma_0$

(iii) ε-rules: $a\rightarrow b$, $a,b\in \Sigma_0$

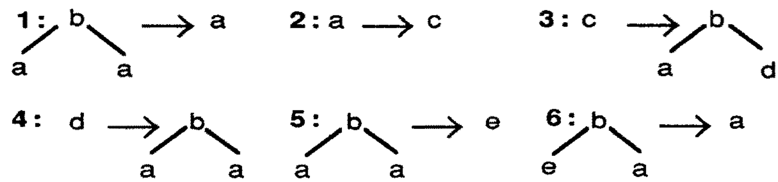
Δ is a finite ranked alphabet, $\Delta=\Sigma\cup I\cup\{\beta_c/c\in \Sigma,\ c\notin \Sigma_0,\ r(\beta_c)=r(c)+2\}\cup\{\alpha/ r(\alpha)=2\}$.

4.2 Derivation proofs for the first-order reachability problem

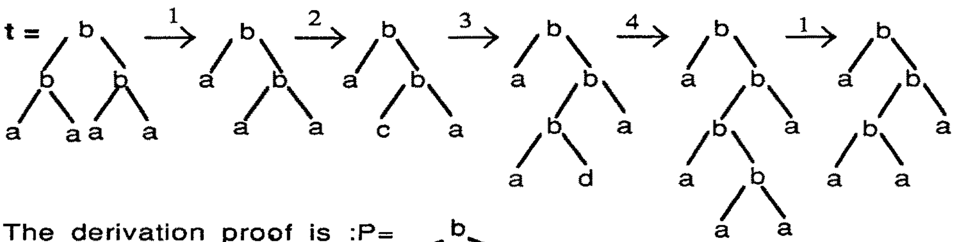
Theorem. Let t, t' be terms in T_Σ and S a standard ground rewrite system, the set of all the derivation proofs for (t,t',S) is a recognizable tree language.

Proof. We define in the sections 4.2.1 to 4.2.5 a regular grammar in order to generate this tree language.

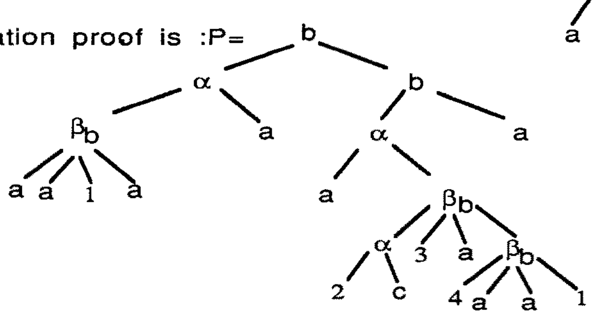
Figure2



Let S={1,2,3,4,5,6} and let us consider the derivation



The derivation proof is :P=



The projection of the frontier of P in I* is 12341

Note: α is for height preserving reductions

Note: β is for height decreasing or increasing reductions

4.2.1: First we study the derivation proofs for (a,b,S) with a, b symbols of Σ_0 with a sequence of reductions such that all reductions with generating rules preceed the reductions with erasing rules. Let G_1 be the regular grammar $\mathfrak{G}_{a,b} = (G_{a,b}, V_1, \Delta, R_1)$ with $V_1 = \{G_{a',b'} \ / \ (a',b') \in \Sigma_0 * \Sigma_0\}$ and R_1 is defined as follow:

$$G_{a,a} \to a + \sum_{i:a\to a'} \alpha(\, i \,,\, G_{a',a}\,) + \sum_{i:b'\to a} \alpha(\, G_{a,b'} \,,\, i\,) + \sum_{\substack{j:a\to c(a_1,\dots,a_n)\\ k:c(a'_1,\dots,a'_n)\to a}} \beta_c(\, j \,,\, G_{a_1,a'_1} ,\dots, G_{a_n,a'_n} \,,\, k\,)$$

$$a\neq b: \quad G_{a,b} \to \sum_{i:a\to a'} \alpha(\, i \,,\, G_{a',b}\,) + \sum_{i:b'\to b} \alpha(\, G_{a,b'} \,,\, i\,) + \sum_{\substack{j:a\to c(a_1,\dots,a_n)\\ k:c(a'_1,\dots,a'_n)\to b}} \beta_c(\, j \,,\, G_{a_1,a'_1} ,\dots, G_{a_n,a'_n} \,,\, k\,)$$

Lemma. $\mathcal{F}(\mathfrak{G}_{a,b})$= {derivation proofs for (a,b,S) with a, b symbols of Σ_0 using all the generating rules before the erasing one} sketch of proof. We first apply ε-rules, then, if we apply a generation rule,the root of the rule will not be erased during the generation part and when this node will be erased,we could only use ε-rules or erasing rules. The reduction terminates with ε-rules. We prove the lemma by induction on the length of the derivation and the height of the tree.

4.2.2: We study the derivation proofs for (a,b,S). We define a regular grammar $\mathcal{H}_{a,b} = (H_{a,b}, V_2, \Delta, R_2)$: $V_2 = V_1 \cup \{H_{a',b'} \ / \ (a',b') \in \Sigma_0 * \Sigma_0\}$

$R_2 = R_1 \cup \{$ rules defined as follow:

$$H_{a',b'} \rightarrow G_{a',b'} + \sum_{c \in \Sigma_0} \alpha(\, H_{a',c} \, , H_{c,b'} \,) + \sum_{\substack{j:a' \rightarrow c(a_1,\ldots,a_n) \\ k:c(a'_1,\ldots,a'_n) \rightarrow b'}} \beta_c(\, j \, , G_{a_1,a'_1} \, ,\ldots, G_{a_n,a'_n} \, , k \,)$$

Lemma. $\mathcal{F}(\mathcal{H}_{a,b})=$ {derivation proofs for (a,b,S) with a, b symbols of Σ_0}

sketch of proof. If in the derivation there is several sequences generation-erasing , we separate them, then we use the rules of R_1. We prove the lemma by induction on the length of the derivation and the height of the tree.

4.2.3: We study the derivation proofs for (t,a,S) where t is a term and a a symbol in Σ_0. We define a regular grammar $\mathcal{H}_{t,a} = (H_{t,a}, V_3, \Delta, R_3)$:

$V_3 = V_2 \cup \{H_{u,a'} \ / \ a' \in \Sigma_0$, u is a ground subtree of t$\}$

$R_3 = R_2 \cup \{$ rules defined as follows:

$$\underset{t \notin \Sigma_0}{H_{t,a}} \rightarrow \sum_{k:c(a'_1,\ldots,a'_n) \rightarrow a'} \beta_c(\, H_{t_1,a'_1} \, ,\ldots, H_{t_n,a'_n} \, , k \, , H_{a',a} \,)$$

Lemma. $\mathcal{F}(\mathcal{H}_{t,a})=$ {derivation proofs for (t,a,S) / $t \in T_\Sigma$, $a \in \Sigma_0$}.

sketch of proof. If the root of t is c , we must erase it during the derivation; so we apply an erasing rule the root of which is c; we iterate that process. When we have only an element of Σ_0, we use the rules of the previous part. We prove the lemma by induction on the length of the derivation and the height of the tree.

4.2.4: We study the derivation proofs for (a,t,S) where t is a term of T_Σ , a is a symbol of Σ_0. We define a regular grammar $\mathcal{H}_{a,t} = (H_{a,t}, V_4, \Delta, R_4)$:

$V_4 = V_2 \cup \{H_{a',u} \ / \ a' \in \Sigma_0$, u is a ground subtree of t$\}$

$R_4 = R_2 \cup \{$ rules defined as follows:

$$\underset{t \notin \Sigma_0}{H_{a,t}} \rightarrow \sum_{k:a' \rightarrow c(a'_1,\ldots,a'_n)} \beta_c(\, H_{a,a'} \, , k \, , H_{a'_1,t_1} ,\ldots, H_{a'_n,t_n})$$

Lemma. $\mathcal{F}(\mathcal{H}_{a,t})=$ {derivation proofs for (a,t,S) / $a \in \Sigma_0$, $t \in T_\Sigma$}.

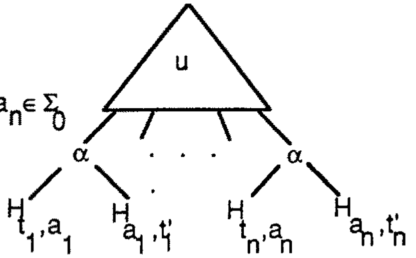
sketch of proof. Similar to the previous one.

4.2.5: We study the derivation proofs for (t,t',S), t and t' terms of T_Σ. We define a regular grammar $\mathcal{H}_{t,t'} = (H_{t,t'}, V_5, \Delta, R_5)$:

$V_5 = V_3 \cup V_4$

$R_5 = R_4 \cup R_3 \cup \{$ rules defined as follows:

C is the set of common contexts of t and t'

$H_{t,t'} \longrightarrow \sum_{\substack{u\in C\\ a_1,\dots,a_n\in\Sigma_0}}$ 

Lemma. $\mathcal{F}(\mathcal{H}_{t,t'})= \{$derivation proofs for (t,t',S) / $t, t'\in\Sigma\}$

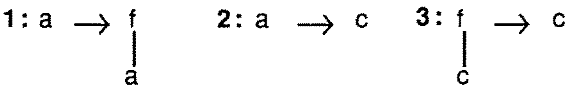
sketch of proof. We first prove

$(t \xrightarrow{*}_S t') \Leftrightarrow (\exists\, u\in T_\Sigma(X_n), \exists\, t_1,\dots,t_n,t'_1,\dots,t'_n \in T_\Sigma , \exists\, a_1,\dots,a_n \in \Sigma_0$

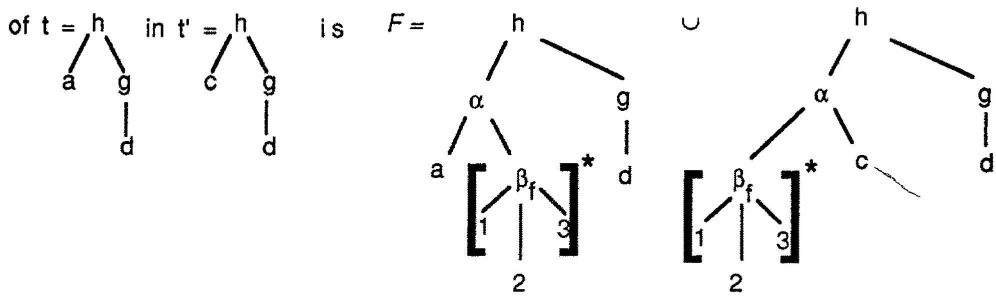
$t=u(t_1,\dots,t_n) \xrightarrow{*}_S u(a_1,\dots,a_n) \xrightarrow{*}_S u(t'_1,\dots,t'_n)=t' \quad)$

using this decomposition, the previous lemmas and the definition of $\mathcal{H}_{t,t'}$, we can prove the result. Moreover, the proof of the theorem is achieved as $\mathcal{F}(\mathcal{H}_{t,t'})$ is a recognizable tree language.

figure 3



Let S={1,2,3}.The recognizable tree language of derivation proofs



4.3 Derivation proofs for the second-order reachability problem

Let F,F' recognizable tree languages and S a standard ground rewrite system. We consider the following second order reachability problem: Is $[F]_S \cap F' \neq \emptyset$, that is to say is there some term t in F which reduces in a term t' in F'.

Theorem. Let F, F' be recognizable tree languages and S a standard ground rewrite system,the set of derivation proofs for (F,F',S) is a recognizable tree language.

sketch of proof. As for the previous theorem, we define a regular grammar to generate this recognizable tree language. The construction is similar, we use the bottom-up automata M and M' such that L(M)=F and L(M')=F'. The derivation proofs for (F,F',S) is the set of all derivation proofs for (t,t',S) such that $t \in F$, $t' \in F'$ and $t \xrightarrow{*}_S t'$.

4.4 A derivation proof for the first order reachability problem

The method used in Sections 4.3 and 4.4 is only interesting from a theoretical point of view, our aim is now to find an algorithm which computes a derivation proof for (t,t',S) in linear time in order to realize an implementation of this algorithm in our software Valeriann.

Proposition. Let S be a standard ground rewrite system, For every data t and t', we can, in linear time, decide if t reduces in t' with S and build a derivation proof for (t,t',S).

Sketch of proof. (a complete proof is in [C090])

Let us remember that it is well-known that for a non deterministic bottom-up-automaton, it is possible to recognize a term t in linear time. The idea is to associate to each node the set of states which can reach this node with M(we reduce the non-determinism along the branches of the term t).

first step: During the compilation of the ground rewrite system S in a ground tree transducer V=(A,B), we associate to each ε-rule a derivation proof. The compilation is in polynomial time (see [DEGI89]).

second step: Let c be the greatest common context of t and t'. Let us consider $T=c(\#(t_1,t'_1),\ldots,\#(t_n,t'_n))$ with # new symbol of arity 2, $t=c(t_1,\ldots,t_n)$ and $t'=c(t'_1,\ldots,t'_n)$. We define a deterministic bottom-up tree transducer U in order to associate to each node of T the sets of states which can reach this node with A and B. We obtain U(T).

Third step: $t \xrightarrow{*}_S t'$ if and only if, for every branch starting from the top of U(T) to a symbol #, there exists a node such that the corresponding sets of states for A and B contain a same interface state. So, we have $t=u(u_1,\ldots,u_n) \xrightarrow{*}_A s=u(i_1,\ldots,i_n) \ {}_B\!\xleftarrow{*} \ t'=u(u'_1,\ldots,u'_n)$ and then $t \xrightarrow{*}_S t'$ and then we can decide in linear time if $t \xrightarrow{*}_S t'$. Further more, we build in linear time a derivation proof. We use the previous decomposition and the table which associate to each ε-rule a derivation proof. This last construction can be implemented with a top-down tree transducer starting from U(T). A complete algorithm is in [CO90].

4.5 General case for ground rewrite systems

If we do not suppose S standard, that is to say S is a ground rewrite system, using new symbols, we can decompose each rule of S in standard rules, then we can associate to S a standard ground rewrite system S'. We can define a regular grammar for S' to obtain the derivation proofs for S' and then with a tree transducer, we can obtain the derivation proofs for S.

5 CONCLUSION

We are now working to extend the algorithm for a derivation proof in linear time to general ground rewrite systems and have to implement this algorithm in Valeriann.

Bibliography

[BR69] Brainerd, W.S, Tree generating regular systems, *Inf. and control*, 14 (1969), pp217-231.

[CO90] Coquidé, J.L. , *Ph.D*, Lille, to appear. (1990).

[COU89] Courcelle, B.,On recognizable sets and tree automata, *Resolution of Equations in Algebraic Structures*, Academic Press, M.Nivat & H. Ait-Kaci edts, (1989).

[DADE89] Dauchet, M., Deruyver, A., "VALERIANN":Compilation of Ground Term Rewriting Systems and Applications, Rewriting Technics and Applications, (1989), *Lec. Notes Comp. Sci.*, 355(Dershowitz ed.).

[DHTL87] Dauchet, M., Heuillard, P., Lescanne, P., Tison, S., Decidability of the Confluence of Ground Term Rewriting Systems, *2nd Symposium on Logic in Computer Science*, New-York, IEEE Computer Society Press (1987)

[DATI85] Dauchet, M., Tison, S., Decidability of Confluence in Ground Term Rewriting Systems, Fondations of Computation Theory, Cottbus, *Lec. Notes Comp. Sci.*, 199, (1985)

[DATI90] Dauchet, M., Tison, S., The theory of Ground Rewrite System is Decidable, IEEE Symposium on Logic in Computer Science, to appear, (1990)

[DEGI89] Deruyver, A., Gilleron, R., Compilation of Term Rewriting Systems, CAAP 89, *Lec. Notes. Comp. Sci.*, (Diaz ed), 354, (1989)

[DEJO89] Dershowitz, N., Jouannaud, J.P., Rewrite systems, *Handbook of Theoretical Computer Science*, J.V.Leeuwen editor, North-Holland, to appear.(1989).

[EN75] Engelfriet, J., Bottom-up and Top-down Tree Transformations, a Comparison, *Math. Systems Theory*, 9, (1975)

[FUVA89] Fülöp, Z., Vàgvölgyi, S., Ground Term Rewriting rules for the Word Problem of Ground Term Equations, submitted paper, (1989).

[GEST84] Gecseg, F., Steinby, M., Tree automata, *Akademiai Kiado*, (1984).

[GI90] Gilleron, R., *Ph.D*, Lille, to appear (1990).

[HUOP80] Huet, G., Oppen, D.C., Equations and Rewrite Rules: A survey, in R.V.Book, ed., New York, *Academic Press*, Formal Language Theory: Perspectives and Open Problems, (1980).

[NEOP80] Nelson, G., Oppen, D.C., Fast Decision Procedures Based on Congruence Closure, *JACM*, 27, (1980).

[OY86] Oyamaguchi, M., The reachability Problem for Quasi-ground Term Rewriting Systems, *Journal of Information Processing*, 9-4, (1986).

[TI89] Tison, S., The Fair Termination is decidable for Ground Systems, Rewriting Technics and Applications, Chapel Hill, *Lec. Notes Comp. Sci.*, 355 (Dershowitz ed.), (1989).