Decidability of the confluence of ground term rewriting systems

Pierre Lescanne, Thierry Heuillard, Max Dauchet, Sophie Tison

INRIA

UNITÉ DE RECHERCHE
INRIA-LORRAINE

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

Rapports de Recherche

N° 675

DECIDABILITY OF THE CONFLUENCE OF GROUND TERM REWRITING SYSTEMS

Pierre LESCANNE
Thierry HEUILLARD
Max DAUCHET
Sophie TISON

JUIN 1987

DECIDABILITY OF THE CONFLUENCE OF GROUND TERM REWRITING SYSTEMS

DECIDABILITE DE LA CONFLUENCE DES SYSTEMES DE REECRITURE DE TERMES CLOS

PAR

PIERRE LESCANNE

THIERRY HEUILLARD

MAX DAUCHET

SOPHIE TISON

Résumé : Le but de cet article est de proposer un algorithme simple pour décider la confluence des systèmes de réécriture de termes clos. Un système de réécriture de termes clos est un système de réécriture où chaque règle est une paire de termes clos, c'est-à-dire de termes sous variables. La confluence n'est pas décidable dans le cas des termes avec variables, nous montrons qu'elle l'est dans le cas des termes clos, suivant en cela une conjecture qu' Huet et Oppen avaient faite dans leur article de synthèse. Nous esquissons aussi un algorithme pour tester cette propriété. Celui-ci est fondé sur des automates d'arbres et des transducteurs d'arbres. Dans cet article, nous considérons les automates d'arbres comme des systèmes de réécritures particuliers. Les spécialistes en théorie des automates traduiront cela facilement dans leur langage.

Abstract : The aim of this paper is to propose a simple algorithm to decide the confluence of ground term rewriting systems. This algorithm is derived from decidability results presented in (1). Here a simple view of the problem and its solution is proposed. Let us recall that a ground term rewriting system is a term rewriting system where each rule is a pair of ground terms, i.e., a pair of terms without variables. Recall the confluence is not decidable for general term rewriting systems, but this paper proves it is for ground term rewriting systems following a conjecture made by Huet and Oppen (3) in their survey. We also sketch a simple algorithm for checking this property. This algorithm is based on tree automata and tree transducers. Here, we regard them as rewriting systems and specialists in automata theory would translate that easily in their language.

In the second part of this paper, a new class of tree transducers is introduced : ground tree transducers, GTT in short. In the third part, stability of GTT relation w.r.t inverse, composition, iteration and precongruence closure of union is proved. In the fourth part, each ground term rewriting system is associated with a GTT and the confluence of ground term rewriting system is reduced to the inclusion of GTT relations. Part 5 solves the decision problem by encoding GTT into rational tree languages.

Decidability of the Confluence of Ground Term Rewriting Systems

Max DAUCHET Sophie TISON
Laboratoire d'Informatique Fondamentale de Lille, UA 369 du CNRS,
Université de Lille-Flandres-Artois,
59655 VILLENEUVE D'ASCQ, FRANCE

Thierry HEUILLARD* Pierre LESCANNE
Centre de Recherche en Informatique de Nancy,
CNRS UA 262 and INRIA-Lorraine,
Campus Scientifique, BP 239,
54506 VANDŒUVRE-LES-NANCY, FRANCE

1 Introduction

The aim of this paper is to propose a simple algorithm to decide the confluence of ground term rewriting systems. This algorithm is derived from decidability results presented in [1]. Here a simple view of the problem and its solution is proposed. Let us recall that a ground term rewriting system is a term rewriting system where each rule is a pair of ground terms, i.e., a pair of terms without variables. The confluence is the property that asserts that $u \stackrel{*}{\leftarrow} s \stackrel{*}{\rightarrow} v$ implies there exists a term t such that $u \stackrel{*}{\rightarrow} t \stackrel{*}{\leftarrow} v$. For example, the reader may try to prove that the system $\{f(f(a)) \rightarrow f(a), g(f(a)) \rightarrow f(g(f(a))), g(f(a)) \rightarrow f(f(a))\}$ is confluent and that the system $\{a \rightarrow f(a,b), f(a,b) \rightarrow f(b,a)\}$ is not. Recall the confluence is not decidable for general term rewriting systems, but this paper proves it is for ground term rewriting systems following a conjecture made by Huet and Oppen [3] in their survey. We also sketch a simple algorithm for checking this property. This algorithm is based on tree automata and tree transducers. Here, we regard them as rewriting systems and specialists in automata theory would translate that easily in their language.

In the second part of this paper, a new class of tree transducers is introduced: *ground tree transducers*, *GTT* in short. In the third part, stability of *GTT* relation w.r.t inverse, composition, iteration and precongruence closure of union is proved. In the fourth part, each ground term rewriting system is associated with a *GTT* and the confluence of ground term rewriting system is reduced to the inclusion of *GTT* relations. Part 5 solves the decision problem by encoding *GTT* into rational tree languages.

*currently at LRI, Université de Paris-Sud, Orsay, France, this work was done as student at the Ecole des mines et de la Métallurgie de Nancy

2 Ground tree transducers

The intuitive idea behind a ground tree transducer is to transform ground terms into ground terms in two steps, with a memory that uses a finite number of values or states. In the first step, one nibbles the term and in the second step one restores a new term. The nibbling consists of transforming some subterms of the term into constants, the states, and the restoration consists of transforming the states into subterms. These operations are rational, this means the relation they define satisfy properties expected for rational relations, namely stability w.r.t. union, composition and iteration.

The steps of nibbling and restoration are described by a tree automaton which is nothing but a ground term rewriting system that works on a set of ground terms enriched by the states as constants. In the following the set of states is named E (or E_G if one wants to refer to a ground term rewriting system G) and its elements are written e. Let F be a finite ranked alphabet. $T(F)$ denotes the set of terms (or trees) on F.

Definition 1 *A relation* $\rightarrow$ *is* F-compatible *or is a* precongruence *if*

$$(\forall f \in F) s \rightarrow t \Rightarrow f(\ldots, s, \ldots) \rightarrow f(\ldots, t, \ldots).$$

Definition 2 *A* tree automaton *G is a rewriting system on $T(F \cup E_G)$ which contains only rules of the form:*

$f(e_1, \ldots, e_n) \rightarrow e_0$ *called* reduction transitions

and

$e \rightarrow e'$ *called* ε-transitions.

for $f \in F$ and $e_1, \ldots, e_n, e_0, e, e'$ in E_G.

The relation $\stackrel{*}{\rightarrow}$ is the least reflexive and transitive precongruence on ground terms that contains $\rightarrow$.

Definition 3 *A* ground tree transducer *on $T(F)$ (a GTT in short) is the relation T or (G, D) associated with two tree automata G and D and defined as follows:*

$t \xrightarrow{T}\!\!\leftarrow t'$ *iff there exists* $u \in T(F \cup E_G \cup E_D)$ *such that* $t \xrightarrow{*}_G u \xleftarrow{*}_D t'$.

In order to produce actual pairs of terms, the set E_G and E_D are supposed non disjoint. $E_G \cap E_D$ is called the *interface*. It could be more intuitive to see u as a term of the form $c(e_1, \ldots, e_n)$ where $c(x_1, \ldots, x_n)$ is a common context of t and t' with $t = c(t_1, \ldots, t_n), t' = c(t'_1, \ldots, t'_n)$ and $t_1 \xrightarrow{*}_G e_1 \xleftarrow{*}_D t'_1, \ldots, t_n \xrightarrow{*}_G e_n \xleftarrow{*}_D t_n$ (see Figure 1.)

A relation associated with a ground term transducer is called a GTT-relation. The relation associated with a ground term transducer T is denoted by $R(T)$. We will first prove a useful lemma.

Lemma 1 *: A GTT-relation can always be associated with a GTT without ε-transition.*

Proof: From the GTT (G, D) we build the GTT (G', D') defined as follows: $E_{G'} = E_G$ and $E_{D'} = E_D$ the rules of G' are of the form:

$$f(e'_n, \ldots, e'_n) \rightarrow_{G'} e'_0$$

if and only if

$$e'_1 \xrightarrow{*}_G e_1,$$
$$\ldots,$$
$$e'_n \xrightarrow{*}_G e_n,$$
$$e_0 \xrightarrow{*}_G e'_0,$$
$$f(e_1, \ldots, e_n) \rightarrow_G e_0.$$

and the same for D. It is easy to see that $t \overset{(G,D)}{\rightarrow\!\!\leftarrow} t'$ if and only if $t \overset{(G',D')}{\rightarrow\!\!\leftarrow} t'$. $\diamond$

3 Stability of GTT relations

From the definition, it is really easy to see that the inverse of a GTT-relation is still a GTT-relation, we have just to commute the rôle of G and D. The union of two GTT-relations R_1 and R_2, is not actually a GTT-relation. However the *precongruence closure* of $R_1 \cup R_2$, which is the least precongruence that contains R_1 and R_2, is associated with the GTT (G_3, D_3) defined by $G_3 \equiv G_2 \cup G_2$ and $D_3 \equiv D_1 \cup D_2$, where (G_1, D_1) and (G_2, D_2) are the GTT associated with R_1 and R_2. This supposes the $E_{G_1} \cup E_{D_1} \neq E_{G_2} \cup E_{D_2}$. So we may state the following proposition.

Proposition 1 *1. The inverse of a GTT-relation is a GTT-relation.*

2. The precongruence closure of the union of two GTT-relations is a GTT-relation.

We are going now to address two other properties of GTT-relations namely the stability by composition and the stability by iteration. We will propose two proofs of these properties, one is based on a diamond lemma and the second one, which is more constructive, is based on properties of minimality of some paths that go from one term to another. Given two relations A and B, we define a relation $\rightarrow_{\{A;B\}}$ by the two inference rules of Figure 2.

We will now prove a diamond lemma.

Lemma 2 *If A and B are described by tree automata without ε-transitions and*

$$u_1 \xleftarrow{*}_{\{A;B\}} t \xrightarrow{*}_{\{B;A\}} u_2$$

with $t \in T(F \cup E_A \cup E_B)$, there exists $s \in T(F \cup E_A \cup E_B)$, such that

$$u_1 \xrightarrow{*}_{\{B;A\}} s \xleftarrow{*}_{\{A;B\}} u_2$$

. In terms of relations we have

$$\xleftarrow{*}_{\{A;B\}} \circ \xrightarrow{*}_{\{B;A\}} \subseteq \xrightarrow{*}_{\{B;A\}} \circ \xleftarrow{*}_{\{A;B\}}$$

Proof: Note that $\leftarrow_{\{A;B\}}$ and $\rightarrow_{\{B;A\}}$ contain ε-transitions. Note also that if $u \xrightarrow{*}_{\{B;A\}} v$ uses only ε-transitions, then $u \xleftarrow{*}_{\{A;B\}} v$ uses only ε-transitions and vice-versa. Let us write $u \overset{\{A;B\}}{\longleftrightarrow} v$ in this case. The proof of the lemma is by induction on the size of the terms. The basic cases are the following.

First case. $u_1 \overset{\{A;B\}}{\longleftrightarrow} t \xrightarrow{*}_{\{B;A\}} u_2$, then if s is taken to be u_2 then $u_1 \xrightarrow{*}_{\{B;A\}} s \equiv u_2$.

Second case. $u_1 \xleftarrow{*}_{\{A;B\}} t \overset{\{A;B\}}{\longleftrightarrow} u_2$, the proof is as previously with $s \equiv u_1$.

General case. The premise of the general case is a diagram of the form given in Figure 3.

Two cases have to be considered, either the occurrences where A and B are used are different or it is the same. If the occurrences are different, we have,

$$t = c(e_1, \ldots, e_n, f(a_1, \ldots, a_p), g(b_1, \ldots, b_q))$$

and

$$t' = c(e'_1, \ldots, e'_n, f(a'_1, \ldots, a'_p), g(b'_1, \ldots, b'_q))$$

and

$$t_1 = c(e_1, \ldots, e_n, a_0, g(b_1, \ldots, b_q))$$

and

$$t_2 = c(e_1, \ldots, e_n, a_0, g(b'_1, \ldots, b'_q))$$

and

$$t'_1 = c(e'_1, \ldots, e'_n, f(a'_1, \ldots, a'_p), b_0)$$

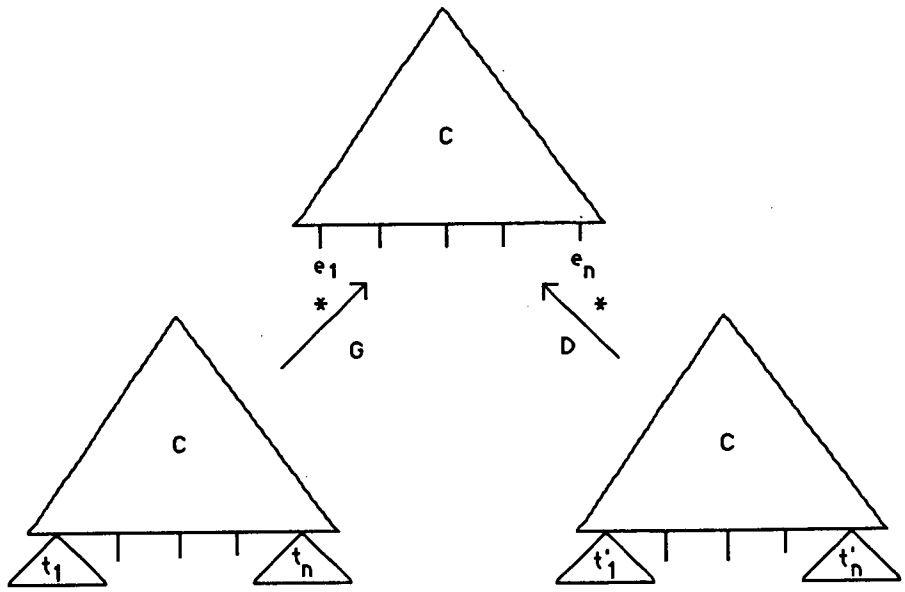


Figure 1: A ground term transduction

$$\frac{f(e_1,\ldots,e_n) \xrightarrow{*}_A e}{f(e_1,\ldots,e_n)\rightarrow_{\{A;B\}} e}$$

$$\frac{e_0 \xleftarrow{*}_B f(e_1,\ldots,e_n)\&e_1 \xrightarrow{*}_{\{A;B\}} e'_1,\ldots,e_n \xrightarrow{*}_{\{A;B\}} e'_n\&f(e'_1,\ldots,e'_n) \xrightarrow{*}_{\{A;B\}} e'_0}{e_0\rightarrow_{\{A;B\}} e'_0}$$

Figure 2: Definition of $\rightarrow_{\{A;B\}}$

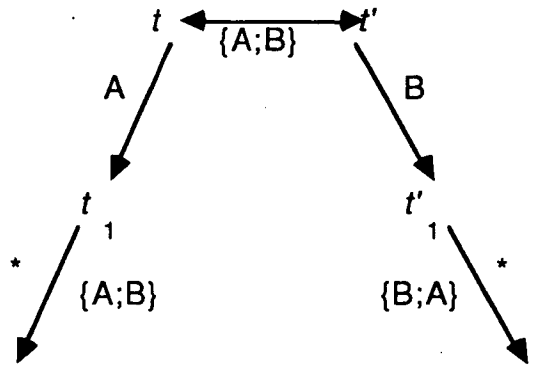


Figure 3: Premise of the general case in Lemma 2.

and

$$t'_2 = c(e'_1, \ldots, e'_n, f(a_1, \ldots, a_p), b_0)$$

where $e_1, \ldots, e_n, e'_1, \ldots, e'_n, a_1, \ldots, a_p, a'_1, \ldots, a'_p, a_0, b_1, \ldots, b_q, b'_1, \ldots, b'_q, b_0$ are in $E_A \cup E_B$.

Let us define

$$s = c(e_1, \ldots, e_n, a_0, b_0)$$

$$s' = c(e'_1, \ldots, e'_n, a_0, b_0)$$

we have $s \overset{\{A;B\}}{\longleftrightarrow} s'$. By induction on the size of the terms, indeed t_1, t'_1, s and s' are of size less than this of t, the diagram can be completed into this of Figure 4.

If the occurrences are the same, we have,

$$t = c(e_1, \ldots, e_n, f(a_1, \ldots, a_p))$$

and

$$t' = c(e'_1, \ldots, e'_n, f(b_1, \ldots, b_p)).$$

We define t_1 and t'_1 as

$$t_1 = c(e_1, \ldots, e_n, a_0)$$

where

$$f(a_1, \ldots, a_p) \overset{*}{\rightarrow}_A a_0,$$

and

$$t'_1 = c(e'_1, \ldots, e'_n, b_0)$$

where

$$f(b_1, \ldots, b_p) \overset{*}{\rightarrow}_B b_0.$$

We have $t_1 \overset{\{A;B\}}{\longleftrightarrow} t'_1$ from the second inference rule, therefore the diagram can be completed into this of Figure 5. $\diamond$

The following propositions are consequences of the previous lemma.

Proposition 2 $\overset{*}{\leftarrow}_A \circ \overset{*}{\rightarrow}_B \subseteq \overset{*}{\rightarrow}_{\{B;A\}} \circ \overset{*}{\leftarrow}_{\{A;B\}}$

Proof: This is obvious from Lemma 2 and from $\leftarrow_A \subseteq \leftarrow_{\{A;B\}}$ and $\rightarrow_B \subseteq \rightarrow_{\{B;A\}}$. $\diamond$

Proposition 3 $\overset{*}{\rightarrow}_B \circ \overset{*}{\leftarrow}_A = \overset{*}{\rightarrow}_{\{B;A\}} \circ \overset{*}{\leftarrow}_{\{A;B\}}$. *The iteration of a GTT-relation is a GTT-relation.*

Proof: The inclusion $\subseteq$ comes by induction from Proposition 2 and Lemma 2. We have $\overset{*}{\rightarrow}_{\{B;A\}} \subseteq (\overset{*}{\rightarrow}_B \circ \overset{*}{\leftarrow}_A)^*$ and $\overset{*}{\leftarrow}_{\{A;B\}} \subseteq (\overset{*}{\rightarrow}_B \circ \overset{*}{\leftarrow}_A)^*$ then obviously $\overset{*}{\rightarrow}_{\{B;A\}} \circ \overset{*}{\leftarrow}_{\{A;B\}} \subseteq (\overset{*}{\rightarrow}_B \circ \overset{*}{\leftarrow}_A)^*$. Since $(\rightarrow_{\{B;A\}}, \leftarrow_{\{A;B\}})$ defines a *GTT*, this provides easily the stability by iteration. $\diamond$

The next proposition is useful for proving the stability by composition.

Proposition 4 *If* $E_A \cap E_B = \emptyset$, *then*

$$\rightarrow_B \circ \leftarrow_A \subseteq \leftarrow_A \circ \rightarrow_B$$

and

$$\overset{*}{\rightarrow}_{\{B;A\}} \circ \overset{*}{\leftarrow}_{\{A;B\}} \subseteq \overset{*}{\leftarrow}_A \circ \overset{*}{\rightarrow}_B$$

Proof: Since the sets of states are disjoint, if $\rightarrow_B$ precedes $\leftarrow_A$, the rewritings take place at two different occurrences and, by the way, they commute. Thus, we obtain $\overset{*}{\rightarrow}_B \circ \overset{*}{\leftarrow}_A \subseteq \overset{*}{\leftarrow}_A \circ \overset{*}{\rightarrow}_B$. We conclude by Proposition 3. $\diamond$

Proposition 5 *The composition of two GTT-relations is a GTT-relation.*

Proof: Suppose that the two relations are associated with the *GTT*'s (G, D) and (Γ, Δ) and that the sets of states satisfy the condition $E_D \cap E_\Gamma = \emptyset$. The relation which is the composition of the relations associated with the two *GTT* is given by $\overset{*}{\rightarrow}_G \circ \overset{*}{\leftarrow}_D \circ \overset{*}{\rightarrow}_\Gamma \circ \overset{*}{\leftarrow}_\Delta$. From Proposition 2 and Proposition 4, this is equal to $\overset{*}{\rightarrow}_G \circ \overset{*}{\rightarrow}_{\{\Gamma,D\}} \circ \overset{*}{\leftarrow}_{\{D,\Gamma\}} \circ \overset{*}{\leftarrow}_\Delta$. By a renaming of the states this can be easily described as a *GTT*. $\diamond$

There is another method for proving the stability based on minimal paths. The idea is to use the relation $\overset{\{A;B\}}{\longleftrightarrow}$. In this relation one can represent transitions of the form $f(e_1, \ldots, e_n) \overset{*}{\rightarrow}_{\{A;B\}} e$ by a descending arrow and relation of the form $e_0 \overset{*}{\rightarrow}_{\{A;B\}} e'_0$ by an horizontal arrow. A concave path is a path starting by descending arrows followed by horizontal arrows then followed by ascending arrows (see Figure 6). The size of a path is the sum of the sizes of the terms occurring along the path.

Proposition 6 *The minimal paths in the relation* $(\overset{*}{\rightarrow}\{B;A\} \circ \overset{*}{\leftarrow}_{\{A;B\}})^*$ *are concave. This means they belong to* $\overset{*}{\rightarrow}_{\{B;A\}} \circ \overset{*}{\leftarrow}_{\{A;B\}}$.

Proof: The principle is by contradiction. The idea is explained in Figure 6, it means that if a path contains a non concave part it is always possible to find a smallest part which is concave. Roughly speaking, if the occurrences where the transitions take place are the same, one can shortcut them by a unique transition, if the occurrences are different, permuting the transitions provides a smaller path because the term below is smaller than the term above. $\diamond$

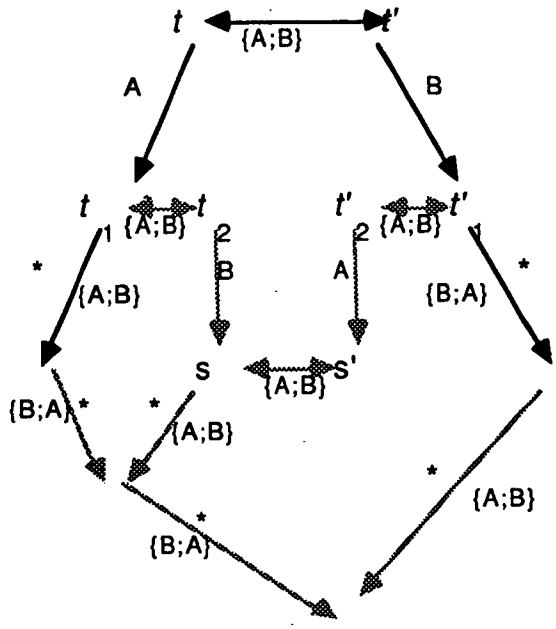


Figure 4: Diagram in the case of different occurrences.

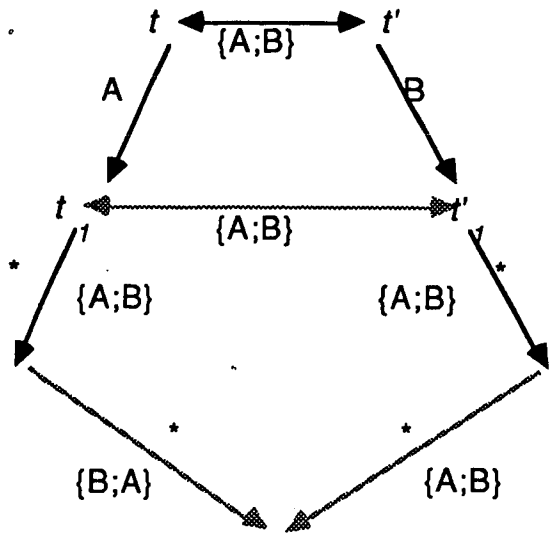


Figure 5: Diagram in the case of the same occurrence.

4 Term Rewriting Systems, Transducers and Confluence

From the previous section, it is now easy to prove that the rewriting relation deduced from a ground term rewriting system can be described by a GTT. First the precongruence that contains a unique rule $s \to t$ is associated with the GTT $(R_s \cup \{e_s \to e_t\}, R_t)$ where R_s and R_t recognize s and t, which means e_s is a state in R_s and $u \xrightarrow{*}_{R_s} e_s$ if and only if u is s and the same for t and R_t. The rewriting relation associated with a ground term rewriting system is obtained from the GTT associated with each rule, by union and iteration, therefore it is a GTT-relation.

The confluence of a rewriting relation $\to$ can be translated into the inclusion

$$\xleftarrow{*} \circ \xrightarrow{*} \subseteq \xrightarrow{*} \circ \xleftarrow{*}$$

. From the previous results these two relations are associated with a GTT. The decidability of this property will be a consequence of the following section which proves that the inclusion of two GTT-relations is decidable.

5 Decidability of the inclusion of two GTT-relations

The idea for deciding the inclusion of two GTT-relations is to map each GTT-relation onto a regular tree language. Since the map is one-to-one, a language will be included in another language if and only if the associated relations are included one in the other. The map is defined as follows. With each pair

$$t = c(t_1, \ldots, t_n) \xrightarrow{*}_G c(e_1, \ldots, e_n)$$

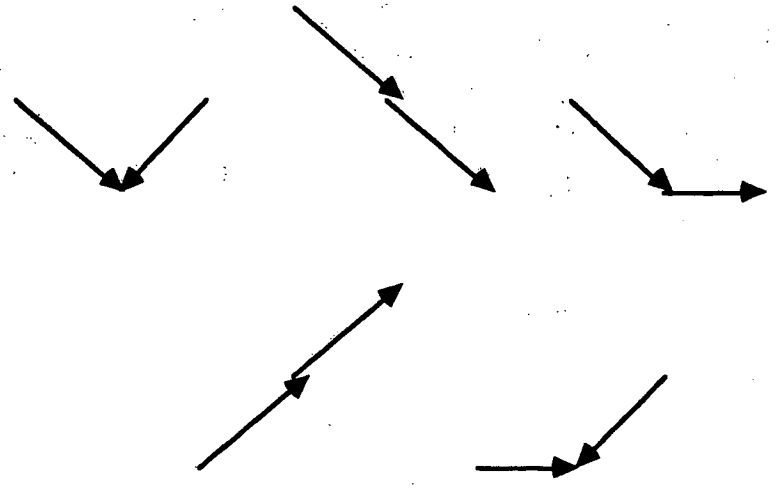
$$c(e_1, \ldots, e_n) \xleftarrow{*}_D t' = c(t'_1, \ldots, t'_n)$$

one associates the term $v\downarrow_\sharp$, where $v\downarrow_\sharp$ is the normal form of the term $v = c(\sharp(t_1, t'_1), \ldots, \sharp(t_n, t'_n))$ for the rewriting system $R_\sharp$ defined by the rules

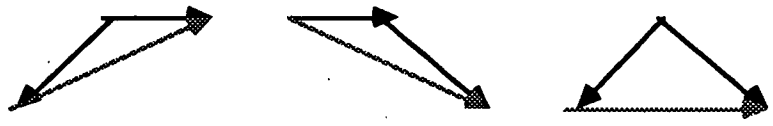
$$\sharp(f(x_1, \ldots, x_p), f(y_1, \ldots, y_p)) \to_{R_\sharp} f(\sharp(x_1, y_1), \ldots, \sharp(x_p, y_p))$$

Let us write $\mathcal{L}(T)$ for the language of the $v\downarrow'_\sharp s$ and let us call it the *tensorial product* of the GTT T. Let us show that $\mathcal{L}(T)$ is recognized by a tree automaton $\mathcal{A}(T)$. The set of states is divided into four parts $E_G, E^c_D, \{ok\}$ and $E_G \times E^c_D$. E^c_D is a copy of E_D in order to avoid confusion. $e^c \in E^c_D$ corresponds to $e \in E_D$. ok is the unique final state, in other words, $v \in \mathcal{L}(T)$ if and only if $v \xrightarrow{*}_{\mathcal{A}(T)} ok$. The transitions are those of G, plus those of the form

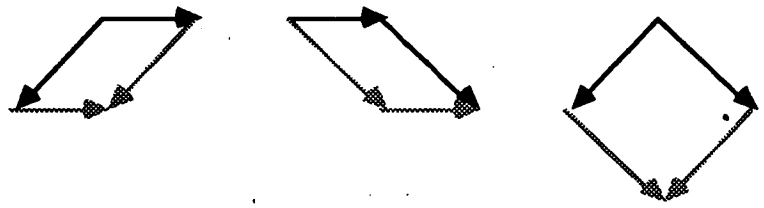
$$f(e^c_1, \ldots, e^c_1) \to e^c \qquad (1)$$



Concave Paths



Same occurrence



Different occurrences

Reducible convex paths

Figure 6: concaveness

$$\sharp(e, e^c) \rightarrow ok \quad (2)$$
$$f(ok, \ldots, ok) \rightarrow ok \quad (3)$$
$$\sharp(e', e^c) \rightarrow \langle e', e^c \rangle \quad (4)$$
$$\langle e, e^c \rangle \rightarrow ok \quad (5)$$
$$f(\langle e_1', e_1^c \rangle, \ldots, \langle e_n', e_n^c \rangle) \rightarrow \langle e', e^c \rangle \quad (6)$$

The first rule corresponds to the rule

$$f(e_1, \ldots, e_n) \rightarrow_D e \quad (7)$$

The last rule corresponds to the rules

$$f(e_1', \ldots, e_n') \rightarrow_G e' \quad (8)$$
$$f(e_1, \ldots, e_n) \rightarrow_D e \quad (9)$$

The first transition translates in E_D^c the transitions of E_D, the second one acknowledges a term $\sharp(s,t)$ where s and t are correctly associated by the GTT T i.e., $s \xrightarrow{T}\!\!\leftarrow t$, the third one is just for transmitting the previous acknowledgement through the context. Since the system $R_\sharp$ pushes down the $\sharp$'s through terms with the same top, it could be the case that a pair is not completely recognized when $\sharp$ appears. The fourth transition is for allowing the recognition to continue. Since the terms are supposed to have the same top, the recognition of the rest of the two terms of the pairs has to be done on both terms together and the recognition has to work on a pair of terms. This is the rôle of the fifth rule. The last rule corresponds to the second one when one works on a pair of states, it means that what has been recognized till this point is the $R_\sharp$-normal form of a term $\sharp(s,t)$ where $s \xrightarrow{T}\!\!\leftarrow t$. Notice that rules (2) and (3) and rules (5) and (6) make $\mathcal{A}(T)$ non deterministic. The following theorem says that $\mathcal{L}(T)$ and $\mathcal{A}(T)$ are correctly associated.

Theorem 1 *The three following statements are equivalent:*

(i) $c(\sharp(t_1, t_1'), \ldots, \sharp(t_n, t_n'))\downarrow_\sharp \in \mathcal{L}(T)$

(ii) $c(t_1, \ldots, t_n) \xrightarrow{T}\!\!\leftarrow c(t_1', \ldots, t_n')$

(iii) $c(\sharp(t_1, t_1'), \ldots, \sharp(t_n, t_n'))\downarrow_\sharp$ *is recognized by the automaton* $\mathcal{A}(T)$, *in other words* $c(\sharp(t_1, t_1'), \ldots, \sharp(t_n, t_n'))\downarrow_\sharp \xrightarrow{*}_{\mathcal{A}(T)} ok$.

Proof: (i) $\Leftrightarrow$ (ii) is a direct consequence of the definition of $\mathcal{L}(T)$. For the proof of (ii) $\Rightarrow$ (iii), let us look at the $\sharp(t_i, t_i')$ part of $c(\sharp(t_1, t_1'), \ldots, \sharp(t_n, t_n'))\downarrow_\sharp$ Its normal form $\sharp(t_i, t_i')\downarrow_\sharp$ is of the form $c_i(\sharp(t_{i,1}, t_{i,1}'), \ldots, \sharp(t_{i,n}, t_{i,n}'))$. If $c(t_1, \ldots, t_n) \xrightarrow{T}\!\!\leftarrow c(t_1', \ldots, t_n')$, we have $t_i \xrightarrow{*}_G e_i \xleftarrow{*}_D t_j'$ for the state e_i of $\mathcal{A}(T)$. To recognize $\sharp(t_i, t_i')\downarrow_\sharp$ i.e., to get $\sharp t_i, t_i')\downarrow_\sharp \xrightarrow{*}_{\mathcal{A}(T)} ok$ one uses the first rule for deriving $t_{i,j} \xrightarrow{*}_{\mathcal{A}(T)} e_{i,j}$ and $t_{i,j}' \xrightarrow{*}_{\mathcal{A}(T)} e_{i,j}'$, then the fourth rule to get $\sharp(e_{i,j}, e_{i,j}') \xrightarrow{*}_{\mathcal{A}(T)} \langle e_{i,j} e_{i,j}' \rangle$, then the fifth rule to get

$$c(\sharp(t_{i,1}, t_{i,1}'), \ldots, \sharp(t_{i,n}, t_{i,n}')) \xrightarrow{*}_{\mathcal{A}(T)} \langle e_i, e_i \rangle$$

, then the sixth rule to get $\langle e_i, e_i \rangle \leftarrow_{\mathcal{A}(T)} ok$. So $c(\sharp(t_1, t_1'), \ldots, \sharp(t_n, t_n')) \leftarrow_{\mathcal{A}(T)} ok$. by using the third rule.

For the proof of (iii) $\Rightarrow$ (ii), the problem is to rebuild the terms $c(t_1, \ldots, t_n)$ and $c(t_1', \ldots, t_n')$ from the recognition of $c(\sharp(t_1, t_1'), \ldots, \sharp(t_n, t_n'))\downarrow_\sharp$, in other words to rebuild $c(\sharp(t_1, t_1'), \ldots, \sharp(t_n, t_n'))$ from its $R_\sharp$-normal form. The idea is just to push up the $\sharp$ in the term till the point where the rules $\langle e_i, e_i \rangle \leftarrow_{\mathcal{A}(T)} ok$ is used. $\diamond$

Since the correspondence between $\mathcal{L}(T)$ and R(T) is one-to-one, the following corollary is an easy consequence of the theorem.

Corollary 1 $R(T) \subseteq R(T')$ *if and only if* $\mathcal{L}(T) \subseteq \mathcal{L}(T')$.

The tree language $\mathcal{L}(T)$ is rational, since it is recognized by the bottom up automaton $\mathcal{A}(T)$. The presentation is a little different of this used in the automata literature, but the reader familiar with this approach can easily convince himself that $\mathcal{A}(T)$ is actually a bottum-up automaton. It is well-known that the inclusion of rational tree languages is decidable. This can be done by extending to bottom-up tree automata the decision procedure for the inclusion of string rational languages. So we have the following corollary.

Corollary 2 *The inclusion of GTT-relations is decidable.*

From the previous section we have the following main corollary.

Corollary 3 *The confluence of a ground tree rewriting system is decidable.*

6 An algorithm for deciding the confluence of ground term rewriting systems

A possible algorithm for deciding the confluence of ground term rewriting system can be divided into three main steps.

- build the GTT associated with the term rewriting system, first one makes a GTT for the precongruence of each individual rule, then a GTT for their union and eventually the iteration of this GTT.

- build the GTT associated with the relations $\overset{*}{\leftarrow} \circ \overset{*}{\rightarrow}$ and $\overset{*}{\rightarrow} \circ \overset{*}{\leftarrow}$ and then build the rational language associated with these relations.
- decide the inclusion of the rational free languages, by classical tree automata techniques.

7 Conclusion

We have shown that the problem of deciding the confluence of ground term rewriting systems can be reduced to the problem of deciding the inclusion of two rational tree languages, problem which has a well-known decision procedure. In addition, this paper proposes a full algorithm. It should be noticed that this algorithm is essentially interesting for no terminating term rewriting systems, because otherwise a method based on superposition (detection of subterms in this case) similar to the Knuth-Bendix algorithm would work. Notice also that it was known for a long time [2] that the termination of such ground term rewriting systems was decidable.

8 References

1. Dauchet M., S. Tison, *Tree automata and decidability in ground term rewriting systems*, Internal Report, Universit de Lille 1, Laboratoire Associé au CNRS 236 and FCT'85, *Lecture Notes in Computer Science* vol.199, pp 80-84 (1985).
2. Huet G., Lankford, *On the Uniform Halting Problem for Term Rewriting Systems*, Rapport Laboria 283 (1978).
3. Huet G., D. Oppen, *Equations and Rewrite Rules*: A Survey, in Formal Languages: Perspectives and Open Problems, Ed. Book R., Academic Press (1980).

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique