Technical Report, Laboratoire de Recherche en Informatique, Université de Paris Sud, to appear. Submitted for publication.

The First-Order Theory of Linear One-Step Rewriting is Undecidable

Ralf Treinen*
Laboratoire de Recherche en Informatique (LRI) & CNRS, Bât. 490
Université de Paris-Sud, F-91405 Orsay cedex, France
email: treinen@lri.fr, web: http://www.lri.fr/~treinen

November 4, 1996

Abstract

The theory of one-step rewriting for a given rewrite system R and signature Σ is the first-order theory of the following structure: its universe consists of all Σ-ground terms, and its only predicate is the relation "x rewrites to y in one step by R". The structure contains no function symbols and no equality. We show that there is no algorithm deciding the $\exists^*\forall^*$-fragment of this theory for an arbitrary linear and non-erasing term-rewriting system.

With the same technique we prove that the theory of encompassment plus one-step rewriting by the rule $f(x) \to g(x)$, the theory of encompassment plus a $WS2S$-definable binary predicate, and the modal theory of one-step rewriting are undecidable.

Sommaire

La théorie de la réécriture pas à pas pour un système de réécriture R et une signature Σ est la théorie du premier ordre de la structure suivante : son univers est formé de tous les Σ-termes clos, et son seul prédicat est "x se réécrit en y par R en un seul pas." Cette structure ne contient pas de symboles de fonctions ni d'égalité. Nous démontrons qu'il n'y a aucun algorithme décidant cette théorie pour n'importe quel système de réécriture linéaire et non-effaçant.

Avec la même technique nous démontrons que la théorie de l'entourage plus la réécriture pas à pas par $f(x) \to g(x)$, la théorie de l'entourage plus un prédicat binaire définissable dans $WS2S$, et la théorie modale de la théorie de la réécriture pas à pas ne sont pas décidables.

*Supported by the *Human Capital and Mobility Program* of the European Union, under the contracts *SOL* (CHRX-CT92-0053) and *CONSOLE* (CHRX-CT94-0495), and by the Esprit working group CCL II (22457).

Contents

1 Introduction

The problem of decidability of the first-order theory of one-step rewriting was posed in [CCD93]. It has been mentioned in the list of open problems in rewriting in 1993 [DJK93] and in 1995 [DJK95].

The theory of one-step rewriting for a given rewrite system R and signature Σ is the first-order theory of the following structure: Its universe consists of all Σ-ground terms, and its only predicate is the relation "x rewrites to y in one step by R". The structure contains no function symbols and no equality. The hope that this theory might be decidable was based on the observation that important properties which are decidable for arbitrary rewrite systems can be expressed in this logic, while undecidable properties of rewrite systems seemed not to be expressible in it. In this paper we prove, however, that this theory is undecidable even for the restricted class of term rewriting systems which are both linear and non-erasing.

1.1 Expressible Properties of Rewriting Systems

The two decidable properties of term rewriting systems which can be expressed in the theory of one-step rewriting are strong confluence and ground reducibility.

- A rewrite system R is *strongly confluent* ([DJ90], this definition differs from the one given in [Hue80]) if

  $$\forall x, y_1, y_2\, (x \to y_1 \wedge x \to y_2 \Rightarrow \exists z\, (y_1 \stackrel{=}{\to} z \wedge y_2 \stackrel{=}{\to} z))$$

  where $x \to y$ means that x rewrites to y in exactly one step of R, and $y \stackrel{=}{\to} z$ means $y = z \vee y \to z$. The system R is *strongly ground confluent* if the above property holds when x is restricted to range over all *ground* terms. It is easy to see that R is strongly confluent iff it is strongly ground confluent in a signature extended by new constants. This property can be expressed in the logic of one-step rewriting, since R is strongly ground confluent if

  $$\forall x, y_1, y_2\, (x \to y_1 \wedge x \to y_2 \Rightarrow \exists z\, (y_1 \to z \wedge y_2 \to z))$$

  holds in the structure of ground terms where $\to$ is the one-step rewriting by the system $R \cup \{x \to x\}$. Note that, as an easy consequence of the Critical Pair Lemma [KB70], strong confluence is decidable.

- A term t is *ground-reducible* by a rewrite system R if each of its ground instances is reducible by R. This can be expressed in a slight extension of our logic, where we have different relations $\to_R$ for different rewrite systems R. Using the additional rewrite system $S = \{t \to t\}$, we can express ground reducibility of t by

  $$\forall x\, (x \to_S x \Rightarrow \exists z\, (x \to_R z))$$

  Decidability of ground reducibility has been shown in [Pla85].

1.2 Decidability Results for Special Cases

Using tree automata techniques, several results have been obtained which have been considered steps towards the decidability of one-step rewriting.

- In case of a signature containing constants and unary function symbols only (so-called *string rewriting systems* or *semi-Thue systems*), the decidability of the theory of one-step rewriting is a consequence of [DT90]. This decidability result can also be obtained by a direct translation into $WS1S$ [Jac95] which is known to be decidable (see [Tho90] for a survey).

- The first-order theory of rewriting by a *ground* term rewriting system has been shown decidable in [DT90]. To be precise, for a given ground term rewriting system R (that is all left and right hand sides of the rules are ground) the structure of all ground terms, where all terms are available as constants, with the predicates "$x \to y$", $x \xrightarrow{*} y$" and "$x \xrightarrow{||} y$" (parallel one-step rewriting) is decidable.

  This result generalizes to the case of left-linear right-ground rewriting systems [Tis90]. The decidability of the theory of one-step rewriting and parallel one-step rewriting only for this class of rewriting systems can also be shown by a direct translation into $WS2S$ using the technique of [Com95].

- For an arbitrary rewrite systems, the decidability of the theory of encompassment was shown in [CCD93]. This means that for given rewrite systems $R_1, \ldots, R_n$ and regular tree languages $L_1, \ldots, L_n$ the structure of ground terms, where all terms are available as constants, with the unary predicates "x is reducible by the rewrite system R_i" and "x is in L_i", is decidable.

Recently, [NPR96] shows the decidability of the positive existential fragment of the theory of one-step rewriting by a reduction to a decidable case of linear second-order unification [Lev96].

Undecidability of the theory of one-step rewriting has been shown for the case of linear and shallow rewriting systems in [STT96]. This proof, however, applies only to the $\exists^*\forall^*\exists^*$ fragment.

1.3 Limits of Expressivity

Note that the theory of the structure consisting of the universe of ground terms together with the predicate "$x \xrightarrow{*} y$" is undecidable in general (even in the restricted case of unary function symbols) by the undecidability of the ground confluence of string rewriting systems [BO93].

One can use the decidability for the case of string rewriting systems to show that the predicate "$x \xrightarrow{*} y$", that is "x rewrites to y in some finite number of steps", can not be expressed in the theory of one-step rewriting (this idea is due to [Jac95]). This is a consequence of the fact that ground confluence of a string rewriting system can be expressed in the logic with the predicate "$x \xrightarrow{*} y$" and hence should be decidable if "$x \xrightarrow{*} y$" can be expressed in the theory of one-step rewriting. Ground confluence of string rewriting systems is however undecidable [BO93][1]. Hence it seems that undecidable properties like confluence or weak termination can not be expressed in the logic of one-step rewriting since they require the "$x \xrightarrow{*} y$" predicate.

1.4 Undecidability Results

The first result of this paper is that there is no algorithm which decides for any rewrite system R the $\exists^*\forall^*$-fragment of the theory of one-step rewriting by R. We construct, for a given instance P of the Post Correspondence Problem, a rewrite system R_P such that solvability of P is equivalent to the validity of a special sentence in the structure of one-step rewriting by R. The formula corresponding to the solvability of P uses both positive and negative literals. This encoding of the Post Correspondence Problem is based on ideas similar to those used in [Tre92].

Our second result concerns the question of decidability of the theory of encompassment plus "simple" classes of rewriting systems. We show that there is no algorithm which decides for any finite set M of terms the $\exists^*\forall^*$-fragment of the theory of the structure of ground-terms, equipped with the predicates "x encompasses t", for any $t \in M$, and the predicate "x rewrites in one step to y by the rule $f(z) \to g(z)$". This rule is syntactically simple: it is linear (there are no multiple variable occurrence on either side), it is non-overlapping (there are only trivial critical pairs) and it is shallow (all variables occur at most at depth 1). Furthermore, this rule has nice semantic properties: it is uniformly confluent (confluent and all derivations with the same source and target have the same length) and it is terminating. As an easy corollary, we show that the theory of encompassment predicates plus *binary* predicates definable by tree automata is in general undecidable. The difference to the class of theories proved decidable in [CCD93] is that only *unary* predicates definable by tree automata are allowed there.

Furthermore, we can show that the *modal theory of one-step rewriting* is undecidable in general. The class of formula of this modal logic is in fact a subclass of the class of first-order formula. The first-order formula obtained by translating modal formulae allow only a very restricted use of variables, e.g. it is not possible to express that an element rewrites to itself, nor to express the confluence property. In fact, the syntax of the modal logic does not employ variables.

An alternative proof of the undecidability of one-step rewriting has been given by Sergei Vorobyov [Vor95]. He defines *one particular* rewrite system R and shows the undecidability of the full first-order theory of one-step rewriting by R. He considers, however, the structure where all the function symbols of the signature are available in the language (this is not the case with our result),

[1] In [BO93], the term "confluence" has been used for what we call "ground confluence".

and he does not attempt to characterize a "simple" undecidable fragment of the theory. The rewrite system of [Vor95] uses in fact a "swapping of variables" to test certain equalities, similar to the system presented here, but it still contains one non-linear rule.

After summarizing some basic notions in Section 2 we give the proof of the main theorem in Section 3. We obtain in 4 the undecidability results for some simple extensions of the theory of encompassment and for the modal theory of one-step rewriting. Section 5 concludes with some remaining open questions.

2 Preliminaries

We summarize the main notions used in this paper, see [DJ90] for an overview of rewriting. We write a signature as a set of function symbols, where we specify (following the PROLOG tradition) the arity of the function symbols after a "/"-sign. The set of terms build over a signature Σ and set X of variables is denoted as $T(\Sigma, X)$, we write $T(\Sigma) = T(\Sigma, \emptyset)$ for the set of Σ-*ground terms*. The set of variables occurring in a term t is denoted as $\mathcal{V}t$.

A Σ-*rewriting system* R is a finite set of so-called *rewriting rules* $l \to r$ where $l \in T(\Sigma, X)$ and $r \in T(\Sigma, \mathcal{V}l)$. A rewriting system is *linear* if no term in its rewriting rules has a multiple occurrence of the same variable. It is *non-erasing* if $\mathcal{V}r = \mathcal{V}l$ holds for any of its rewriting rules $r \to l$.

A term $t \in T(\Sigma, X)$ *rewrites* to s in one step by R if there is a rewriting rule $l \to r$ in R, an occurrence o and a substitution σ such that $t\mid_o = l\sigma$ and $s = t[o \leftarrow r\sigma]$.

We say that a term t *encompasses* a term s if there is a subterm of t which is an instance of s. In other words, t encompasses s if any rewrite rule with left-hand side s reduces t.

We consider first-order predicate logic *without* equality. The $\exists^*\forall^*$-*fragment* of a theory T is the subset of T of all sentences having a prenex normal form of the form

$$\exists x_1, \ldots, x_n \, \forall y_1, \ldots, y_m \, Q$$

where Q contains no quantifier.

We denote concatenation of words by juxtaposition. The length of a word w is written as $|w|$, and w_i, where $1 \leq i \leq |w|$, is the i-th symbol of w, that is $w = w_1 \ldots w_{|w|}$. We write $v < w$ if v is a proper prefix of w.

An instance of the *Post Correspondence Problem (PCP)* is a finite set of pairs of non-empty binary words $\{(p_i, q_i) \mid 1 \leq i \leq n, p_i, q_i \in \{a, b\}^+\}$. A solution of P is a finite non-empty sequence $(i_1, \ldots, i_m) \in \{1, \ldots, n\}^+$ such that

$$p_{i_1} \cdots p_{i_m} = q_{i_1} \cdots q_{i_m}$$

It is undecidable whether an instance of the PCP has a solution [Pos46].

3 One-Step Rewriting

Definition 3.1 *Let Σ be a signature and R be a Σ-rewrite system. The structure $\mathcal{A}_{\Sigma,R}$ is defined as follows: The language of $\mathcal{A}_{\Sigma,R}$ contains no constants or function symbols, and its only predicate symbol is the binary predicate symbol $\rightarrow$. The universe of $\mathcal{A}_{\Sigma,R}$ is the set $T(\Sigma)$, and $t \rightarrow s$ holds in $\mathcal{A}_{\Sigma,R}$ iff t rewrites to s in one rewriting step of R.*

Theorem 3.2 *There is no algorithm which decides for any signature Σ and any non-erasing and linear Σ-rewrite system R the $\exists^*\forall^*$-fragment of the theory of $\mathcal{A}_{\Sigma,R}$.*

We show how to reduce the solvability of an instance of the Post-Correspondence Problem (PCP) to the validity of some $\exists^*\forall^*$-sentence in $\mathcal{A}_{\Sigma,R}$ for some signature Σ and non-erasing and linear rewrite system R. All constructions and proofs are parameterized by the given instance of the PCP. For the sake of convenience, we now fix this instance for the rest of the paper to be

$$P = \{(p_i, q_i) \mid 1 \leq i \leq n\}$$

Definition 3.3 *The signature Σ_P is*

$$\{\epsilon/0, a/1, b/1, g/3, k/5, eq/3\}$$

A Σ_P-ground term t is called a P-term if it does not contain an occurrence of k or eq.

The idea is to encode a solution $(i_1, \ldots, i_m)$ of P by the term depicted in Figure 1. Words from $\{a, b\}^*$ can easily be encoded in $T(\Sigma)$. First we define an application of a word from $\{a, b\}$ to an arbitrary term $t \in T(\Sigma)$ inductively by

$$\begin{array}{rcl} \epsilon(t) & = & t \\ wa(t) & = & a(w(t)) \\ wb(t) & = & b(w(t)) \end{array}$$

Note that, for the case of $a(t)$ and $b(t)$, this coincides with the definition of the operations in $\mathcal{A}_{\Sigma,R}$. A word $w \in \{a, b\}^*$ is now represented by the term $w(\epsilon)$. Note that

- the empty word is represented by the term ϵ,
- the encoding is injective, that is equality of words translates to equality of their respective representations,
- and $w(t)$ represents the word vw iff t represents the word v.

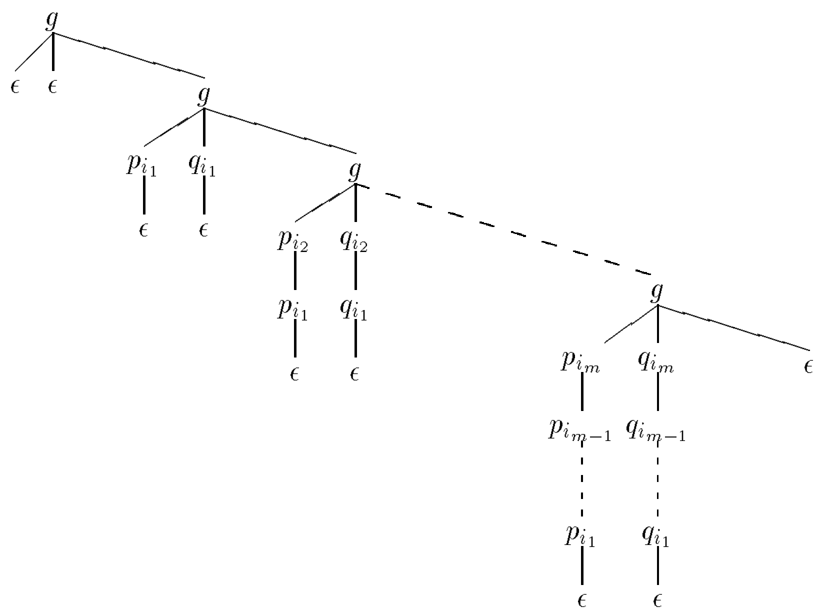

Figure 1: Representation of the solution $(i_1, \ldots, i_m)$ of P.

Definition 3.4 *The rewrite system R_P consists of the following rules:*

$$\begin{aligned}
k(x_1, x_2, x_3, x_4, x_5) &\rightarrow k(x_1, x_2, x_3, x_4, x_5) && (1)\\
eq(x_1, x_2, x_3) &\rightarrow eq(x_1, x_2, x_3) && (2)\\
g(\epsilon, \epsilon, g(x, y, z)) &\rightarrow g(x, y, z) && (3)\\
g(x_1, y_1, g(x_2, y_2, z)) &\rightarrow k(x_1, y_1, x_2, y_2, z) && (4)\\
\{k(x_1, y_1, p_i(x_2), q_i(y_2), z) &\rightarrow g(x_2, y_2, g(p_i(x_1), q_i(y_1), z)) \mid 1 \leq i \leq n\} && (5)\\
\{g(x, y, h(\bar{z})) &\rightarrow eq(x, y, h(\bar{z})) \mid h \in \{\epsilon, a, b\}\} && (6)\\
eq(x, y, h(\bar{z})) &\rightarrow g(y, x, h(\bar{z})) && (7)
\end{aligned}$$

We use the following formulae

$$\begin{aligned}
\Phi(x) &:= \neg x \rightarrow x\\
\Psi(x) &:= \Phi(x) \wedge \exists x' \, (x \rightarrow x' \wedge \Phi(x'))
\end{aligned}$$

Proposition 3.5 *1. $\mathcal{A}_{\Sigma_P, R_P}, \alpha \models \Phi(x)$ iff $\alpha(x)$ is a P-term.*

2. $\mathcal{A}_{\Sigma_P, R_P}, \alpha \models \Psi(x)$ iff $\alpha(x)$ is a P-term containing a subterm of the form $g(\epsilon, \epsilon, g(x, y, z))$.

Proof: The first claim holds since (1) and (2) are the only rules that can rewrite a term to itself. The second claim holds since (3) is the only rule that can rewrite a P-term to a P-term. $\square$

Lemma 3.6 *P is solvable iff*

$$\mathcal{A}_{\Sigma_P,R_P} \models \exists x \left(\underbrace{\Psi(x) \wedge \forall y \left((x \to y \wedge \neg\Phi(y)) \Rightarrow y \to x\right)}_{(8)} \right)$$

Proof: Let $(i_1, \ldots, i_m)$ be a solution of P. Consider the term

$$t = g(\epsilon, \epsilon, g(r_1, s_1, \ldots, g(r_m, s_m, \epsilon) \ldots))$$

where

$$\begin{aligned} r_k &= p_{i_k}(p_{i_{k-1}}(\cdots(p_{i_1}(\epsilon))\cdots)) \\ s_k &= q_{i_k}(q_{i_{k-1}}(\cdots(q_{i_1}(\epsilon))\cdots)) \end{aligned}$$

for $1 \leq k \leq m$. It is easy to see that the formula 8 is satisfied when taking t for the value of x. Note that (4) and (6) are the only rules that can rewrite a P-term to a non-P-term.

On the other hand, let the formula (8) be satisfied by the the term t. We can show by induction on s that every subterm s of t which starts with g is of the form

$$t = g(r_1, s_1, \ldots, g(r_m, s_m, u) \ldots))$$

where u is either ϵ or starts with a or b, and where there is a sequence $(i_1, \ldots, i_m) \in \{1, \ldots, n\}^*$ such that for all $1 < j \leq m$

$$\begin{aligned} r_j &= p_{i_j}(r_{j-1}) \\ s_j &= q_{i_j}(s_{j-1})) \\ r_m &= s_m \end{aligned}$$

Note that, since t is a P-term, any subterm s of t must contain only the symbols g, ϵ, a, b. Hence, if s is a subterm of t starting with g, then either rule (4) or (6) rewrite s at the root to some non-P-term s', which can only be rewritten back to s by the rules (5), resp. (7).

The claim follows since t contains, by Proposition 3.5, a subterm of the form $g(\epsilon, \epsilon, g(\cdot, \cdot, \cdot))$. □

4 Related Theories

In this section we derive with slight modifications of the proof of Section 3 undecidability results for related theories.

4.1 Encompassment plus One-Step Rewriting by $f(x) \to g(x)$

Definition 4.1 *Let Δ be the signature $\{\epsilon/0, a/1, b/1, f/1, g/1, h/3\}$ and let M be a subset of $T(\Delta, X)$. The structure $\mathcal{B}_M$ is defined as follows: The language of $\mathcal{B}_M$ contains no constants or function symbols, and its only predicate symbols are the unary symbols Ψ_u for every $u \in M$ and the binary predicate symbol $\to$. The universe of $\mathcal{B}_M$ is the set $T(\Delta)$, $\Psi_u(t)$ holds in $\mathcal{B}_M$ if t encompasses u, and $t \to s$ holds in $\mathcal{B}_M$ iff t rewrites to s in one rewriting step of $f(x) \to g(x)$.*

Theorem 4.2 *There is no algorithm which decides for any finite set $M \subseteq T(\Delta, X)$ the $\exists^*\forall^*$-fragment of the theory of $\mathcal{B}_M$.*

We will construct, for every instance P of the PCP, a finite set $M \subseteq T(\Delta, X)$ such that P is solvable iff a certain $\exists^*\forall^*$-sentence is valid in $\mathcal{B}_M$. We fix an instance of the PCP:

$$P = \{(p_i, q_i) \mid 1 \leq i \leq n\}$$

We define a first set of terms

$$\begin{aligned}
M_1 \;:=\; & \{g(x)\} && (9)\\
\cup\; & \{a(f(x)), a(h(\bar{x})), b(f(x)), b(h(\bar{x}))\} && (10)\\
\cup\; & \{h(h(\bar{x}), y, z), h(f(x), y, z), h(x, h(\bar{y}), z), h(x, f(y), z)\} && (11)\\
\cup\; & \{h(x, y, h(\bar{z})), h(x, y, a(z)), h(x, y, b(z))\} && (12)\\
\cup\; & \{f(\epsilon), f(a(x)), f(b(x)), f(f(x))\} && (13)
\end{aligned}$$

Definition 4.3 *A term $t \in T(\Delta, X)$ is a* list *if it is of either of the two forms*

$$h(r_1, s_1, f(h(r_2, s_2, \ldots, f(h(r_m, s_m, \epsilon)) \ldots)))$$
$$f(h(r_1, s_1, f(h(r_2, s_2, \ldots, f(h(r_m, s_m, \epsilon)) \ldots))))$$

where $m \geq 0$ and $r_i, s_i \in T(\{\epsilon, a, b\})$ for $1 \leq i \leq m$.

We now introduce the abbreviation

$$list(x) := \bigwedge_{u \in M_1} \neg\Psi_u(x)$$

Proposition 4.4 *$\mathcal{B}_P, \alpha \models list(x)$ iff $\alpha(x)$ is a list.*

Proof: Note that (9) means that t may contain ϵ, a, b, f and h only, (10) that any son of a or b is in $T(\{\epsilon, a, b\})$, (11) that the first two sons of h are in $T(\{\epsilon, a, b\})$, (12) that the last son of h is f or ϵ, and (13) that every son of f is h. $\square$

We now define a second set M_P of terms. In contrast to M_1, this set depends on the instance P of the PCP. Note that this set contains non-linear patterns.

$$M_P := \{g(h(\epsilon, \epsilon, z))\} \cup \{h(x, y, g(h(p_i(x), q_i(y), z))) \mid 1 \leq i \leq m\}$$

The idea is to use the rule $f(x) \to g(x)$ to mark an occurrence of f in g. More precisely, to express that some pattern π matches a term x not containing g we say that $\pi[g/f]$ matches any y where $x \to y$.

Lemma 4.5 *P is solvable iff*

$$\mathcal{B}_M \models \exists x \left(list(x) \wedge \Psi_{h(\epsilon,\epsilon,f(z))}(x) \wedge \Psi_{h(v,v,\epsilon)}(x) \wedge \forall y \; (x \to y \Rightarrow \bigvee_{u \in M_P} \Psi_u(y)) \right)$$

4.2 Encompassment plus $WS2S$-Definable Predicates

Definition 4.6 *Let Δ be the signature $\{\epsilon/0, a/1, b/1, f/1, g/1, h/3\}$, M be a subset of $T(\Delta, X)$, and A a tree-automaton over the signature $\Delta \times \Delta$. The structure $\mathcal{C}_{M,A}$ is defined as follows: The language of $\mathcal{C}_M$ contains no constants or function symbols, and its only predicate symbols are the unary symbols Ψ_u for every $u \in M$ and the binary predicate symbol ρ. The universe of $\mathcal{C}_{M,A}$ is the set $T(\Delta)$, $\Psi_u(t)$ holds in $\mathcal{C}_{M,A}$ if t encompasses u, and $t\rho s$ holds in $\mathcal{B}_{M,A}$ iff A accepts the tree obtained by superimposing t and s.*

Theorem 4.7 *There is no algorithm which decides for any finite set $M \subseteq T(\Delta, X)$ and any automaton A the $\exists^*\forall^*$-fragment of the theory of $\mathcal{C}_{M,A}$.*

Proof: This is in fact an immediate corollary of Theorem 4.2 since the one-step rewriting relation by the rule $f(x) \to g(x)$ is defined by a tree automaton. $\square$

4.3 The Modal Theory of One-Step Rewriting

Definition 4.8 *For a given set L of binary relation symbols the set of L-*modal *formula M_R is given by*

- *$\top$ and $\bot$ are L-modal formulae,*
- *any boolean combination of L-modal formulae is a modal formula,*
- *if w is a L-modal formula and $\rho \in L$ then $\Diamond_\rho w$ and $\Box_\rho w$ are L-modal formula.*

The semantic of a L-modal formula is given by a translation into first-order formula over the language L. The translation function takes two arguments: the formula to translate and a variable. This is the well-known *relational translation*, see e.g. [Ohl91].

Definition 4.9

$$
\begin{aligned}
\phi(x)(\top) &= \top \\
\phi(x)(\bot) &= \bot \\
\phi(x)(w_1 \circ w_2) &= \phi(x)(w_1) \circ \phi(x)(w_2) \\
\phi(x)(\Diamond_\rho w) &= \exists y\ (x\rho y \wedge \phi(y)(w)) \\
\phi(x)(\Box_\rho w) &= \forall y\ (x\rho y \Rightarrow \phi(y)(w))
\end{aligned}
$$

where $\circ$ is any boolean junctor.

R-modal logic is significantly weaker than full first order logic. The lack of variables in the modal logic excludes formulae like $\rho(x, x)$ or even the formula expressing the strong ground confluence of a relation. We can however express ground-reducibility: a term t is ground-reducible by a rule $l \to r$ if

$$(\Diamond_{t \to t} \top) \wedge (\Box_{l \to r} \bot)$$

is not satisfiable.

Definition 4.10 *The* modal theory *of a Σ-rewrite system R is the set of all L-modal formulae satisfiable in $\mathcal{A}_{\Sigma,R}$, where $L = \{\to_r \mid r \in R\}$.*

Theorem 4.11 *There is no algorithm which decides for any Σ and any Σ-rewrite system R the satisfiability of any $\Box\Diamond$-formula of the modal theory of R.*

Proof: We apply the same construction as in the proof of Lemma 4.5. Note that we can express $\Psi_u(x)$ in the modal logic by $\Diamond_{u \to u} \top$. Hence, solvability of P is now equivalent to the satisfiability of

$$\bigwedge_{u \in M_1} \neg \Psi_u \wedge \Psi_{h(\epsilon,\epsilon,f(z))} \wedge \Psi_{h(v,v,\epsilon)} \wedge \Box_{f(x) \to g(x)} \bigvee_{u \in M_P} \Psi_u(y)$$

$\Box$

5 Conclusions

The remaining open cases of the theory of one step rewriting fall in two categories: restrictions of the class of formulae, and restrictions of the class of rewriting systems. In the first category, we have the existential fragment of the theory of one step rewriting (i.e., satisfiability of quantifier-free formulae) and the positive theory (i.e., arbitrary quantification but no negation and no implication). In the latter category, the case of right-ground and the cases of orthogonal rewriting systems remain open.

I'm very grateful to Hubert Comon who pointed me to this problem, and to Max Dauchet, Aart Middeldorp and in particular Florent Jacquemard for many discussions and suggestions.

References

[BO93] Ronald V. Book and Friedrich Otto. *String-Rewriting Systems.* Texts and Monographs in Computer Science. Springer-Verlag, 1993.

[CCD93] Anne-Cécile Caron, Jean-Luc Coquide, and Max Dauchet. Encompassment properties and automata with constraints. In Kirchner [Kir93], pages 328–342.

[Com95] Hubert Comon. Sequentiality, second-order monadic logic and tree automata. In *Tenth Annual IEEE Symposium on Logic in Computer Science*, San Diego, CA, June 1995. IEEE Computer Society Press.

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In van Leeuwen [vL90], chapter 6, pages 243–320.

[DJK93] Nachum Dershowitz, Jean-Pierre Jouannaud, and Jan Willem Klop. More problems in rewriting. In Kirchner [Kir93], pages 468–487.

[DJK95] Nachum Dershowitz, Jean-Pierre Jouannaud, and Jan Willem Klop. Problems in rewriting III. In Jieh Hsiang, editor, *Rewriting Techniques and Applications*, Lecture Notes in Computer Science, vol. 914, pages 457–471. Springer-Verlag, April 1995.

[DT90] Max Dauchet and Sophie Tison. The theory of ground rewrite systems is decidable. In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 242–256. IEEE Computer Society Press, 1990.

[Hue80] Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, October 1980.

[Jac95] Florent Jacquemard. Personal Communication, December 1995.

[KB70] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[Kir93] Claude Kirchner, editor. *Rewriting Techniques and Applications, 5th International Conference, RTA-93*, Lecture Notes in Computer Science, vol. 690, Montreal, Canada, 1993. Springer Verlag.

[Lev96] Jordi Levy. Linear second-order unification. In Harald Ganzinger, editor, *7th International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science, vol. 1103, pages 332–346, New Brunswick, NJ, USA, July 1996. Springer-Verlag.

[NPR96] Joachim Niehren, Manfred Pinkal, and Peter Ruhrberg. Equality up to constraints, context unification, and second order linear unification, 1996. In Preparation.

[Ohl91] H. J. Ohlbach. Semantics-based translation methods for modal logics. *Journal of Logic and Computation*, 1(5):691–746, 1991.

[Pla85] David A. Plaisted. Semantic confluence tests and completion methods. *Information and Computation*, 65(2/3):182–215, 1985.

[Pos46] Emil L. Post. A variant of a recursively unsolvable problem. *Bulletin of the AMS*, 52:264–268, 1946.

[STT96] Franck Seynhaeve, Marc Tommasi, and Ralf Treinen. Grid structure and undecidable properties, 1996. To appear.

[Tho90] Wolfgang Thomas. Automata on infinite objects. In van Leeuwen [vL90], chapter 4, pages 133–191.

[Tis90] Sophie Tison. Automates comme outil de décision dans les arbres. Dossier d'habilitation à diriger des recherches, Université des Sciences et Techniques de Lille, Lille, France, December 1990.

[Tre92] Ralf Treinen. A new method for undecidability proofs of first order theories. *Journal of Symbolic Computation*, 14(5):437–457, November 1992.

[vL90] Jan van Leeuwen, editor. *Handbook of Theoretical Computer Science*, volume B - Formal Models and Semantics. Elsevier Science Publishers and The MIT Press, 1990.

[Vor95] Sergei Vorobyov. Elementary theory of one-step rewriting is undecidable (note). Draft, 1995.