# Decidable Approximations of Term Rewriting Systems

Florent Jacquemard

Laboratoire de Recherche en Informatique. CNRS URA 410
Univ. Paris-Sud, Bât. 490
91405 Orsay cedex. France
phone: +33 (1) 69 41 66 35 e-mail: jacquema@lri.fr

**Abstract.** A linear term rewriting system $\mathcal{R}$ is *growing* when, for every rule $l \to r \in \mathcal{R}$, each variable which is shared by $l$ and $r$ occurs at depth one in $l$. We show that the set of ground terms having a normal form w.r.t. a growing rewrite system is recognized by a finite tree automaton. This implies in particular that reachability and sequentiality of growing rewrite systems are decidable. Moreover, the word problem is decidable for related equational theories. We prove that our conditions are actually necessary: relaxing them yields undecidability of reachability.
Concerning sequentiality, the result may be restated in terms of approximations of term rewriting systems. An approximation of a system $\mathcal{R}$ is a renaming of the variables in the right hand sides which yields a growing rewrite system. This gives the decidability of a new sufficient condition for sequentiality of left-linear rewrite systems, which encompasses known decidable properties such as strong, NV and NVNF sequentiality.

## 1 Introduction

Most properties of term rewriting systems are undecidable. For instance, this is the case of termination (Huet and Lankford [12]), (Dauchet [3] for the case of one rule rewrite system), confluence (Huet [10]), word problem (Post [20]), reachability and sequentiality (Huet and Lévy [14]). Hence much efforts have been devoted to give sufficient conditions under which these properties are decidable. For example, for ground rewrite systems all these properties are decidable (Dauchet et al [5]). When the systems are just right ground and left-linear, the reachability (Oyamaguchi [17]) and the termination (Dershowitz [6]) are still decidable.

Our result can be understood in two ways:

1. to give a class of rewrite systems for which sets of normalizable ground terms are recognizable and reachability and other properties are decidable,
2. to give a general decidable approximation of rewrite systems.

Indeed, instead of restricting the class of term rewriting systems under consideration, one may see the problem as finding decidable approximations of the reduction relation. This approach may be interesting e.g. when studying sequential reduction strategies. The sequentiality is a property which has been introduced by Huet and Lévy [14]. They characterize left-linear rewrite systems for which there exists a normalizing reduction strategy. Sequentiality is undecidable for (general) left-linear rewrite systems. Thus, if we want to apply Huet and Lévy's reduction strategy, we are left to check sufficient conditions, namely properties stronger than sequentiality which are decidable. For instance, Huet and Lévy [14] show the decidability of strong sequentiality in the case of orthogonal term rewriting systems, and Sadfi [21] and Comon [1] generalized this result to left linear systems. This corresponds to the sequentiality of an approximation of the rewrite system in which all right hand sides are replaced by fresh variables. The interesting point is that roughly, a rewrite system is sequential as soon as its approximation is sequential, which allows to use the normalizing strategy of [14] for rewrite systems whose approximation is sequential – yet decidable.

In this paper, we give a class of rewrite systems for which both reachability and sequentiality are decidable. This corresponds also to a finer approximation than forgetting about the right hand sides. There were a lot of works in this direction. For example, Nagaya, Sakai and Toyama [16] show that the renaming of variables in the right hand sides with fresh variables yields a finer – yet decidable – approximation. They call NVNF-sequential an orthogonal and left-linear rewrite system for which this approximation is sequential. Actually, NVNF-sequentiality is an extension of Oyamaguchi [19] NV-sequentiality. Dauchet and Tison [4] showed that the first order theory of left-linear and right-ground and rewrite systems is decidable. Decidability of NV-sequentiality (resp. NVNF-sequentiality), even without the orthogonality assumption, is a direct consequence of this result, as noticed by Comon [1]. More generally, Comon shows that the sequentiality is decidable for a class of linear rewrite system whenever for each system $\mathcal{R}$ of the class, the set of ground terms which have a normal form w.r.t. $\mathcal{R}$ is recognizable by a finite tree automaton. He proves this result by showing that if the above hypothesis holds, then sequentiality may be expressed in Rabin's weak second order monadic theory of the tree. As a consequence, sequentiality is decidable for shallow linear rewrite systems [1] – in which all the variables occur at depth at most 1. Following this idea, we give a more general result here: if in any rule $l \to r$ of a linear rewrite system $\mathcal{R}$, each variable occurring in both $l$ and $r$ occurs at depth 1 in $l$, then the sequentiality of $\mathcal{R}$ is decidable, as well as the reachability. A rewrite system satisfying these properties is called *growing*. We achieve this result, proving first that the set of normalizable ground terms for a growing rewrite system is recognized by a finite tree automaton. Growing rewrite systems provide a better approximation of rewrite systems, hence of sequentiality: there are indeed rewrite systems whose such approximations are sequential and not the rougher ones.

The paper is organized as follows: the basics concerning term rewriting systems and tree automata are recalled respectively in Sect. 2 and 3. We give our recognizability result in Sect. 4. In Sect. 5 we derive that reachability is decidable for growing rewrite systems and give similar results for the word problem in section 6. In Sect. 7, we show that these results also apply to sequentiality, defining a new decidable criterium for sequentiality. Moreover, we present a comparison between different notions of sequentiality in the literature, redefined in terms of approximations. At last, we remark in Sect. 8 that these results cannot be generalized: we give undecidability results for systems in which the growing condition is relaxed.

## 2 Term Rewriting

We assume that the reader is familiar with notions of terms and term rewriting system. We recall some basic definitions. Missing ones can be found in the survey of Dershowitz and Jouannaud [7].

*Terms.* $\mathcal{F}$ is a finite ranked alphabet of function symbols. The set of (ground) terms with function symbols in $\mathcal{F}$ is written $\mathcal{T}(\mathcal{F})$. The *arity* of a function symbol $f \in \mathcal{F}$ is noted $ar(f)$. $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is the set of terms built over $\mathcal{F}$ and a set of variable symbols $\mathcal{X}$. $\mathcal{V}ar(t)$ is the set of variables occurring in $t$. A term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is called *linear* if every variable $x \in \mathcal{V}ar(t)$ occurs at most once in $t$.

A term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ may be viewed as a labeled tree $t = f(t_1 \dots t_n)$ where $n = ar(f)$ and the subtrees $t_1 \dots t_n$ are terms of $\mathcal{T}(\mathcal{F}, \mathcal{X})$. We may also see $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ as a mapping from the set of strings $\{1 \dots k\}^*$ into $\mathcal{F} \cup \mathcal{X}$, where $k = Max\{ar(f) \mid f \in \mathcal{F}\}$. If $\varepsilon$ is the empty string and $p \cdot p'$ is the concatenation of strings $p$ and $p'$, we define this mapping recursively by $t(\varepsilon) = f$ and for each $1 \leq i \leq k$, and each $p \in \{1 \dots k\}^*$, $t(i \cdot p) = t_i(p)$. $\mathcal{P}os(t)$ is the domain of the mapping $t$ and is called the set of *positions* of $t$.

The *subterm* of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ at position $p \in \mathcal{P}os(t)$ is $t|_p$ and $t[t']_u$ is the term in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ obtained by *replacement* of the subterm $t|_p$ at position $p \in \mathcal{P}os(t)$ by $t' \in \mathcal{T}(\mathcal{F}, X)$. We may also use the notation $t[t']_p$ to indicate that $t|_p = t'$.

A substitution is a mapping from $\mathcal{X}$ to $\mathcal{T}(\mathcal{F}, X)$. As usual, we do not distinguish between a substitution and its homomorphism extension in the free algebra $\mathcal{T}(\mathcal{F}, X)$. A bijective substitution from $\mathcal{X}$ to $\mathcal{X}$ is called a *variable renaming.* The postfix notation $t\sigma$ is used for the application of a substitution $\sigma$ to a term $t$. A term $t\sigma$ is called an *instance* of $t$.

*Rewrite Systems.* A *Rewrite system* over $\mathcal{F}$ is a finite set of *rules* which are ordered pairs of terms $(l, r) \in \mathcal{T}(\mathcal{F}, \mathcal{X})^2$, commonly written $l \to r$. The term $l$ (resp. $r$) is called the left (resp. right) member of the rule $l \to r$. When all the terms $l$ and $r$ in $\mathcal{R}$ are ground, resp. linear, $\mathcal{R}$ is a *ground rewrite system*, resp. *linear rewrite system.* Moreover, if all the left hand sides $l$ (resp. right hand sides $r$) of $\mathcal{R}$ are ground or linear, then $\mathcal{R}$ is called a *left-ground*[1], *left-linear* (resp. *right-ground, right-linear*) rewrite system.

A term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ rewrites to $t'$ by the system $\mathcal{R}$ iff there exists a rule $l \to r$ in $\mathcal{R}$, a position $p \in \mathcal{P}os(t)$ and a substitution $\sigma$ such that $l\sigma = t|_p$ and $t' = t[r\sigma]_p$. The subterm $t|_p$ is called a *redex.* In that case, we write $t \xrightarrow[\mathcal{R}]{} t'$. We

[1] Never occurs

also say that $t$ is *reducible* by $\mathcal{R}$. A term is said to be a *normal form* of $\mathcal{R}$ if it is not reducible by $\mathcal{R}$. A term $t$ is *normalizable* w.r.t. a rewrite system $\mathcal{R}$ iff there exists a normal form $n$ of $\mathcal{R}$ such that $t \xrightarrow[\mathcal{R}]{*} n$, where the binary relation $\xrightarrow[\mathcal{R}]{*}$ is the reflexive and transitive closure of $\xrightarrow[\mathcal{R}]{}$. A rewrite system $\mathcal{R}$ is non-ambiguous if there is no rule $l \to r \in \mathcal{R}$ with $\exists l' \; l \xrightarrow[\mathcal{R}\setminus\{l\to r\}]{} l'$

## 3 Tree Automata

*Tree automata* could also be called *term automata*. Following the custom, we use the former terminology, though it is in the use to talk about *terms* in rewriting theory. Moreover, though we didn't mention it in the previous section, the terms we consider are always finite, i.e. of finite domain. Thus we will also use *finite* tree automata.

**Definition 1.** A bottom-up finite tree automaton is a t-uple $(\mathcal{F}, Q, Q_f, \Delta)$ where $\mathcal{F}$ is a finite ranked alphabet, $Q$ is a finite set of states and $Q_f \subseteq Q$, $\Delta$ is a ground rewriting system over $\mathcal{F} \cup Q$ whose rules have the form $f(q_1 \dots q_n) \to q$ where $f \in \mathcal{F}$, the arity of $f$ is $n$ and $q_1 \dots q_n, q \in Q$.

The term $t \in \mathcal{T}(\mathcal{F})$ is *accepted* - recognized - by the automaton $(\mathcal{F}, Q, Q_f, \Delta)$ iff there exists $q \in Q_f$ such that $t \xrightarrow[\Delta]{*} q$ [2]. The language $L(\mathcal{A})$ of an automaton $\mathcal{A}$ is the set of ground terms in $\mathcal{T}(\mathcal{F})$ accepted by $\mathcal{A}$.

Let us recall some basic properties of finite tree automata – see the reference book of Gécseg and Steinby [9] for more details.

***Proposition 2.*** *The class of languages of ground terms recognized by a finite tree automaton is closed under Boolean operations.*

This means, for every automata $\mathcal{A}_1$ and $\mathcal{A}_2$, there are automata accepting respectively the languages $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$, $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ and $\mathcal{T}(\mathcal{F}) \setminus L(\mathcal{A}_1)$.

***Proposition 3.*** *Emptiness is decidable for finite tree automata.*

In other words, there is an algorithm which decides for any automaton $\mathcal{A}$ whether $L(\mathcal{A}) = \emptyset$ or not.

## 4 Recognizability of Sets of Normalizable Ground Terms

Now, we are ready to state our main result, concerning recognizability of set of ground normalizable terms.

**Definition 4.** A *growing* rewrite system is a linear rewrite system over an alphabet $\mathcal{F}$ such that for every rule $l \to r \in \mathcal{R}$ and every pair of positions $u \in \mathcal{P}os(l)$, $v \in \mathcal{P}os(r)$ such that $l(u) = r(v) \in \mathcal{X}$ we have $|u| \leq 1$.

[2] We may write also write $\xrightarrow[\mathcal{A}]{}$ for the relation $\xrightarrow[\Delta]{}$.

***Theorem 5.*** *Let $\mathcal{R}$ be a growing rewrite system. Then the set of ground terms which are normalizable w.r.t. $\mathcal{R}$ is recognizable by a tree automaton.*

To prove the theorem, we have to construct a tree automaton from the given rewrite system $\mathcal{R}$. For this purpose, the two following lemmas will be helpful.

***Lemma 6.*** *For each linear term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, there exists an automaton $\mathcal{A}_t$ which recognizes the ground instances of $t$.*

***Lemma 7.*** *There exists an automaton $\mathcal{A}_{\text{NF}}$ which recognizes the set of ground terms which are in normal form w.r.t. $\mathcal{R}$.*

Such constructions have been exposed by Gallier and Book [8].
Let $\mathcal{L} = \bigcup_{f(l_1 \ldots l_n) \to r \in \mathcal{R}} \{l_1 \ldots l_n\}$ and let $\mathcal{A}_0$ be the following sum of automata:

$$\mathcal{A}_0 = (\mathcal{F}, Q, Q_0, \Delta_0) := \biguplus_{l \in \mathcal{L}} \mathcal{A}_l \uplus \mathcal{A}_{\text{NF}}$$

where the (disjoint) sum of two automata with the same alphabet is the automaton whose set of states, set of final states and set of rules are the union of corresponding sets of the two automata, provided that they are all disjoint.

Then, we complete $\Delta_0$ using the following two inferences rules. Each application of 1 or 2 transforms $\Delta_k$ $(k \geq 0)$ into $\Delta_{k+1}$.

$$\frac{f(l_1 \ldots l_n) \to g(r_1 \ldots r_m) \in \mathcal{R} \quad g(q_1 \ldots q_m) \to q \in \Delta_k}{f(q'_1 \ldots q'_n) \to q \in \Delta_{k+1}} \tag{1}$$

with the conditions:

1. For all $1 \leq i \leq n$, if $l_i$ is a variable then $q'_i$ is any state of $Q$, otherwise $q'_i$ is a final state of $\mathcal{A}_{l_i}$.
2. For all $1 \leq j \leq m$ there exists a substitution $\theta : \mathcal{X} \to Q$ such that $r_j\theta \xrightarrow[\Delta_k]{*} q_j$ and for each $l_i \in \mathcal{X}$ occurring in $g(r_1 \ldots r_m)$ $(1 \leq i \leq n)$, we have $l_i\theta = q'_i$.

$$\frac{f(l_1 \ldots l_n) \to x \in \mathcal{R}_w \quad q \in \Delta_k}{f(q'_1 \ldots q'_n) \to q \in \Delta_{k+1}} \tag{2}$$

with the conditions:

1. $x$ is a variable
2. For all $1 \leq i \leq n$, if $l_i = x$ then $q'_i = q$
   if $l_i$ is a variable different from $x$ then $q'_i$ is any state of $Q$
   otherwise $q'_i$ is a final state of $\mathcal{A}_{l_i}$

The number of states of $\mathcal{A}_0$ $(|Q|)$ is finite and no new state is added in 1 or 2. So the process stops with a set $\Delta$ when there is no new rule to add.

The desired automaton is

$$\mathcal{A} := (\mathcal{F}, Q, Q_{\text{NF}}, \Delta)$$

where $Q_{\text{NF}}$ is the set of final states of $\mathcal{A}_{\text{NF}}$. Now, we show the two inclusions in the next two lemmas. Let $\mathcal{A}_k$ be the automaton $(\mathcal{F}, Q, Q_{\text{NF}}, \Delta_k)$.

***Lemma 8.*** *For each $k$, $L(\mathcal{A}_k) \subseteq \{t \in \mathcal{T}(\mathcal{F}) \mid t$ is normalizable w.r.t. $\mathcal{R}\}$.*

*Proof.* (sketch) First, we show by induction on $k$ that
**(fact1)** for every $k$, for every ground term $t \in \mathcal{T}(\mathcal{F})$ and for every final state $q_l$ of some automaton $\mathcal{A}_l$ $(l \in \mathcal{L})$ if $t \xrightarrow[\mathcal{A}_k]{*} q_l$ then $t \xrightarrow[\mathcal{R}]{*} l\sigma$ where $l\sigma$ is a ground instance of $l$.

1. For $k = 1$, by construction of $\mathcal{A}_1$ as a disjoint sum of automata, if $t \xrightarrow[\mathcal{A}_1]{*} q_l$, a final state of an $\mathcal{A}_l$ $(l \in \mathcal{L})$, then $t$ is a ground instance of $l$.
2. Assume this is true for $k$ and that $t \xrightarrow[\Delta_{k+1}]{*} q_l$, a final state of an $\mathcal{A}_l$ $(l \in \mathcal{L})$. We use an induction on the number $n$ of reduction steps using the rule $\Delta_{k+1} \setminus \Delta_k$ in the above reduction sequence.
   (a) If $n = 0$, we can directly use the induction hypothesis concerning $\mathcal{A}_k$.
   (b) If $n > 0$, we consider the first application of the rule $\Delta_{k+1} \setminus \Delta_k$ in the reduction sequence. Let $f(q'_1 \dots q'_n) \to q$ be this rule and assume it had been added to $\mathcal{A}_k$ by inference 1 with the hypothesis $f(l_1 \dots l_n) \to g(r_1 \dots r_m) \in \mathcal{R}$ and $g(q_1 \dots q_m) \to q \in \Delta_k$. Then the proof can be summarized in the following diagram:

$$
\begin{tikzcd}
t = t[f(t_1 \dots t_n)]_p \arrow[d, "*"', "\mathcal{R}"] & & & \\
t[f(l_1 \dots l_n)\sigma]_p \arrow[r, "*", "\mathcal{A}_k"'] \arrow[d, "\mathcal{R}"] & t'[f(q'_1 \dots q'_n)]_p \arrow[r, "\mathcal{A}_{k+1} \setminus \mathcal{A}_k"'] & t'[q]_p \arrow[r, "*", "\mathcal{A}_{k+1}"'] & q_l \\
t[g(r_1 \dots r_m)\sigma]_p \arrow[r, "*", "\mathcal{A}_k"'] & t'[g(r_1 \dots r_m)\theta]_p \arrow[r, "*", "\mathcal{A}_k"'] & t'[g(q_1 \dots q_m)]_p \arrow[u, "\mathcal{A}_k"'] &
\end{tikzcd}
$$

   For the reduction $t \xrightarrow[\mathcal{R}]{*} t[f(l_1 \dots l_n)\sigma]_p$, condition 1 of inference 1 says that if $l_i \notin \mathcal{X}$, $q'_i$ is a final state of $\mathcal{A}_{l_i}$ and by induction hypothesis, $t_i \xrightarrow[\mathcal{A}_k]{*} q'_i$ implies $t_i \xrightarrow[\mathcal{R}]{*} l_i\sigma$.
   The substitution $\theta$ is defined in condition 2. We have $r_i\sigma \xrightarrow[\mathcal{A}_k]{*} r_i\theta$ for each $1 \leq i \leq m$ because $l_j \in \mathcal{V}ar(r_i)$ implies $l_j\theta = q'_j$ (condition 2) and $l_j\sigma \xrightarrow[\mathcal{A}_k]{*} q'_j$. On the other hand, $\mathcal{V}ar(r_i) \cap \mathcal{V}ar(f(l_1 \dots l_n)) \subseteq \{l_1 \dots l_n\}$, because $\mathcal{R}$ is growing. Condition 2 also implies $r_i\theta \xrightarrow[\mathcal{A}_k]{*} q_i$. By induction hypothesis on the number of applications of $\Delta_{k+1} \setminus \Delta_k$, $t[g(r_1 \dots r_m)\sigma]_p$ reduces to some ground instance of $l$, thus so does $t$. The proof would have been almost the same if the rule $\Delta_{k+1} \setminus \Delta_k$ had been deduced using the inference 2.

Now, we prove the lemma 8 by induction on $k$ using **fact 1**.

1. The case $k = 1$ is obvious because $L(\mathcal{A}_1) = L(\mathcal{A}_{\text{NF}})$ by construction.
2. For $k+1$, assume $t \xrightarrow[\mathcal{A}_{k+1}]{*} q_f$ where $q_f$ is a final state of $\mathcal{A}_{k+1}$. We use the same induction as in the proof of **fact 1**. The diagram for the step (case of inference 1) is now:

$$\begin{tikzcd}
t \arrow[r, "*", "\mathcal{A}_k"'] \arrow[d, "*"', "\mathcal{R}"] & t'[f(q'_1 \dots q'_n)]_p \arrow[r, "\Delta_{k+1} \backslash \Delta_k"'] & t'[q]_p \arrow[r, "*", "\mathcal{A}_{k+1}"'] & q_f \\
t[f(l_1 \dots l_n)\sigma]_p \arrow[d, "\mathcal{R}"] & & & \\
t[g(r_1 \dots r_m)\sigma]_p \arrow[r, "*", "\mathcal{A}_k"'] & t'[g(r_1 \dots r_m)\theta]_p \arrow[r, "*", "\mathcal{A}_k"'] & t'[g(q_1 \dots q_m)]_p \arrow[uu, "\mathcal{A}_k"'] &
\end{tikzcd}$$

The reduction $* \downarrow \mathcal{R}$ follows from **fact 1**. The other reductions are deduced as in the proof of **fact 1**. By induction hypothesis – on the number of applications of $\Delta_{k+1} \setminus \Delta_k$, $t[g(r_1 \dots r_n)\sigma]_p$ reduces to a normal form w.r.t. $\mathcal{R}$, thus so does $t$. □

***Lemma 9.*** *$L(\mathcal{A}) \supseteq \{t \in \mathcal{T}(\mathcal{F}) \mid t$ is normalizable w.r.t. $\mathcal{R}\}$*

*Proof.* Assume that $t \xrightarrow[\mathcal{R}]{*} n$ for some normal form $n$ w.r.t. $\mathcal{R}$. We show by induction on the length $n$ of this reduction sequence that $t \in L(\mathcal{A})$.

1. If $n = 0$, $t \in L(\mathcal{A})$ because $L(\mathcal{A}_{\text{NF}}) \subseteq L(\mathcal{A})$.
2. If $n > 0$, assume that the first reduction step uses the rule $f(l_1 \dots l_n) \to g(r_1 \dots r_m) \in \mathcal{R}$ with substitution $\sigma$, that $r_i\sigma \xrightarrow[\mathcal{A}]{*} q_i$ and that $g(q_1 \dots q_m) \to q \in \Delta$.

$$\begin{tikzcd}
t = t[f(l_1 \dots l_n)\sigma]_p \arrow[r, "\mathcal{R}"'] \arrow[dd, "*"', "\mathcal{A}"] & t[g(r_1 \dots r_m)\sigma]_p \arrow[r, "*"', "\mathcal{A}"'] & t[g(r_1 \dots r_m)\theta]_p \arrow[d, "*"', "\mathcal{A}"] \\
& & t[g(q_1 \dots q_m)]_p \arrow[d, "\mathcal{A}"] \\
t[f(q'_1 \dots q'_n)]_p \arrow[rr, dashed, "\mathcal{A}"'] & & t[q]_p \arrow[d, "*"', "\mathcal{A}"] \\
& & q_f
\end{tikzcd}$$

The reduction $t[g(r_1 \dots r_m)\sigma]_p \xrightarrow[\mathcal{A}]{*} q_f$ follows from the induction hypothesis ($q_f$ is a final state of $\mathcal{A}$). The substitution $\theta$ is deduced from this reduction by $\mathcal{A}$ and has the right form for condition 2. For each $l_i$ which is a variable ($1 \le i \le n$), the corresponding state $q'_i$ is defined by $\theta$ and then $q'_i$ fulfills the condition 1. Thus, the inference rule 1 has been applied during the construction of $\Delta$ and the transition $t[f(q'_1 \dots q'_n)]_p \xrightarrow[\mathcal{A}]{} t[q]_p$ is possible. The proof would be the same if the first reduction step would have used a rule $f(l_1 \dots l_n) \to x$. □

# 5 Decision of Reachability

We will see how the result of the previous section applies to some decision problems in rewriting. The reachability problem can be stated as follows:

> Given $\mathcal{R}$ a rewrite system over $\mathcal{F}$ and two ground terms $t$ and $t'$ in $\mathcal{T}(\mathcal{F})$, do we have $t \xrightarrow[\mathcal{R}]{*} t'$ ?

It is well known that this problem is undecidable in general. It has been shown decidable for ground rewrite systems by Togachi and Noguchi [22], for left-linear and right-ground rewrite systems by Dauchet et al. [5] and Oyamaguchi [17] and for right-ground rewrite systems by Oyamaguchi [18]. Now, applying Theorem 5, we have:

***Theorem 10.*** *Reachability is decidable for growing rewrite systems.*

Note that this result extends (strictly) former results for left-linear systems.

*Proof.* (sketch) Let $\mathcal{R}$ be a rewrite system as in the theorem, over an alphabet $\mathcal{F}$ and $t, t' \in \mathcal{T}(\mathcal{F})$. Let $\mathcal{F}' := \mathcal{F} \uplus \{c, d\}$ and $\mathcal{R}'$ be the following system over $\mathcal{F}'$:

$$\mathcal{R}' \; := \; \mathcal{R} \;\cup\; t' \to d \;\cup\; \bigcup_{f \in \mathcal{F}} f(x_1 \dots x_n) \to c \;\;\cup\; \{c \to c\} \; .$$

Note that $d$ is the only normal form of $\mathcal{R}'$. Then we have $t \xrightarrow[\mathcal{R}]{*} t'$ iff $t$ is normalizable w.r.t. $\mathcal{R}'$. By Theorem 5, there is an automaton $\mathcal{A}$ which recognizes the set of ground terms which are normalizable w.r.t. $\mathcal{R}'$. So the reachability problem for $\mathcal{R}$, $t$, $t'$ reduces to the membership of $t$ to $L(\mathcal{A})$ which is decidable. □

# 6 Decision of Word Problem

As a consequence, symetryzing the condition of theorem 10 to both left and right members of rewrite rules, we can state:

***Corollary 11.*** *The word problem*[3] *is decidable for linear rewrite systems $\mathcal{R}$ such that for each rule $l \to r \in \mathcal{R}$ and each pair of positions $u \in \mathcal{P}os(l)$, $v \in \mathcal{P}os(r)$ such that $l(u) = r(v) = x \in \mathcal{X}$ we have $|u|, |v| \leq 1$.*

*Proof.* If $\mathcal{R} = \{l_1 \to r_1 \dots l_n \to t_n\}$ is as in the statements of Corollary 11, then $\{l_1 \to r_1, r_1 \to l_1 \dots l_n \to r_n, r_n \to l_n\}$ is a growing system, thus it has a decidable reachability problem, by Theorem 10. □

This is a slight generalization for linear systems of the decidability proof of the word problem by Comon, Haberstrau and Jouannaud [2] for shallow systems.

The section 8 shows that Theorem 10 and Corollary 11 cannot be easily generalized.

## 7 Sequentiality

The property of sequentiality for left-linear rewrite systems was introduced by Huet and Lévy [14]. The main property of a sequential rewrite system is that there exists an effective call-by-need normalizing strategy.

### 7.1 Definition

Our purpose is less to study theoretical and practical implications of sequentiality than to give decidability results. Moreover, the decidability proof refers to a coding of Comon [1]. For all these reasons, we will go quickly with the definitions concerning sequentiality. For a more complete presentation of sequentiality, the reader is referred to Huet and Levy [14] and Klop and Middeldorp [15].

Informally, a rewrite system is sequential if any normalizable term contains a redex which is necessarily contracted in a reduction to a normal form. In the literature, these redexes are called *index*. In order to define formally the notion of necessity, we classically use *partially evaluated terms* also called *$\Omega$-terms*. An $\Omega$-term is a term on the signature $\mathcal{F}$ augmented by a new constant $\Omega$. The $\Omega$'s represent subterms which have not been evaluated so far. A position $p$ in an $\Omega$-term $t$ such that $t(p) = \Omega$ is an index if it needs to be evaluated (replaced) in order to get a normal form. The formal notion of "more evaluated" is given by the relation $\sqsubseteq$ on $\Omega$-terms and defined by: $\Omega \sqsubseteq t$ for every $\Omega$-term $t$ and if $t_i \sqsubseteq u_i$ for each $1 \leq i \leq n$ then $f(t_1 \dots t_n) \sqsubseteq f(u_1 \dots u_n)$. An index of an $\Omega$-term $t$ for a rewrite system $\mathcal{R}$ is a position $p \in \mathcal{P}os(t)$ such that $t(p) = \Omega$ and for each $\Omega$-term $u \sqsupseteq t$ which reduces by $\mathcal{R}$ to a normal form in $\mathcal{T}(\mathcal{F})$, we have $u(p) \neq \Omega$. Then, a left-linear rewrite system $\mathcal{R}$ is sequential if for each $\Omega$-term $t$ which cannot be reduced by $\mathcal{R}$ into a normal form in $\mathcal{T}(\mathcal{F})$ but such that there exists $u \sqsupseteq t$ with $u \xrightarrow[\mathcal{R}]{*} n \in \mathcal{T}(\mathcal{F})$ ($n$ is in normal form), then $t$ has an index.

[3] The word problem for a rewrite system $\{l_1 \to r_1 \dots l_n \to t_n\}$ over $\mathcal{F}$, and two terms $t, t' \in \mathcal{T}(\mathcal{F})$ is the reachability problem for $\{l_1 \to r_1, r_1 \to l_1 \dots l_n \to r_n, r_n \to l_n\}$, $t$ and $t'$.

*Example 1.* Let $\mathcal{R}$ be the left-linear rewriting system defining the parallel or $\vee$: $\mathcal{R} = \vee(\top, x_1) \to \top,\ \ \vee(x_2, \top) \to \top$. Consider the partially evaluated term $t = \vee(\Omega, \Omega)$, which is in normal form w.r.t. $\mathcal{R}$. We have $t \sqsubseteq \vee(\top, \Omega) \underset{\mathcal{R}}{\longrightarrow} \top$ where the constant $\top$ is a normal form of $\mathcal{R}$. By the above reduction, the position 2 is not an index. This is neither the case of the position 1, because $t \sqsubseteq \vee(\Omega, \top) \underset{\mathcal{R}}{\longrightarrow} \top$. Thus, $\mathcal{R}$ is not sequential.

In the general case, sequentiality is undecidable [14]. The reachability – see Sect. 5 – indeed reduces to the sequentiality decision, and this problem is known to be undecidable even for linear rewrite systems. Thus we are left to check sufficient conditions, namely properties stronger than sequentiality which are decidable. When a non-ambiguous left-linear rewrite system fulfills such a property, the normalizing strategy is still applicable. Toyama investigates in [23] some case of ambiguous left-linear systems for which the index reduction strategy is normalizing. We will give some conditions of the literature in term of *approximations.*

### 7.2 Approximations of Rewrite System

An approximation of a rewrite system is a rougher system, in the sense that the associated binary relation on terms is rougher.

**Definition 12.** An approximation of a rewrite system $\mathcal{R}$ is another system $\mathcal{R}'$ such that $\underset{\mathcal{R}}{\longrightarrow} \subseteq \underset{\mathcal{R}'}{\longrightarrow}$.

*Example 2.* $\{f(x_1, x_2) \to h(x_1', x_2'),\ f(x_3, 0) \to x_3'\}$ and $\{f(x_1, x_2) \to x_1',\ f(x_3, 0) \to x_3\}$ are approximations of $\{f(x_1, x_2) \to h(g(x_1), x_1),\ f(x_3, 0) \to x_3\}$.

In the following, an approximation is understood as a mapping $\tau$ which associates to each rewrite system $\mathcal{R}$ one of its approximations $\tau(\mathcal{R})$ in the sense of definition 12. If an approximation $\tau$ is such that for every $\mathcal{R}$, $\tau(\mathcal{R})$ is right-linear, as in the case presented in example 2, then we say that $\tau$ is a *right-linear approximation.*

The idea is to find a right-linear approximation $\tau$ such that given a left-linear rewrite system $\mathcal{R}$, it is decidable whether $\tau(\mathcal{R})$ is sequential. In this case, we may say for short that sequentiality for $\tau$ is decidable. The fact that these conditions really imply sequentiality is given by the following lemma:

***Lemma 13.*** *For any right-linear approximation $\tau$ and any left-linear system $\mathcal{R}$, if $\tau(\mathcal{R})$ is sequential, then so is $\mathcal{R}$.*

### 7.3 Decidable Approximations and Sequentiality

Historically, the first approximation which has been considered consists in replacing the right hand sides of the rules by distincts fresh variables. The sequentiality for such an approximation is Huet and Lévy's *strong sequentiality* [13]. These two authors showed the decidability of strong sequentiality for left-linear and non-overlapping (any rule cannot reduce another) systems in [14]. Sadfi [21] and Comon [1] have shown the decidability in the possibly overlapping and left-linear case.

Nagaya et al. introduced in [16] NVNF-sequentiality which is sequentiality of a finer approximation for left-linear and non overlapping systems and extends NV-sequentiality of Oyamaguchi [19]. This consists in renaming all variables of all right members of rules by pairwise distincts fresh variables.

*Example 3.* If $\mathcal{R}$ is $\{f(x_1, f(x_2, x_3)) \to f(g(x_1, x_2), x_3), f(x_4, 0) \to x_4\}$ its NVNF approximation is $\{f(x_1, f(x_2, x_3)) \to f(g(x'_1, x'_2), x'_3), f(x_4, 0) \to x'_4\}$.

Comon [1] proved this result for left-linear systems.

At last, Comon [1] also proved that sequentiality is decidable for shallow linear rewrite systems, this is for systems whose all left and right hand sides of rules are a variable or have all their variables occurring at depth 1. This result could be generalized to show that the sequentiality for a *shallow approximation* (shallow sequentiality for short) is decidable, where shallow approximating a rewrite system consists in renaming by fresh variables in right members all the sharing variables except those which are at depth less or equal than 1 in both left members and right members.

### 7.4 Growing Sequentiality

We consider here *growing approximation* generalizing former approximations:

**Definition 14 Growing Approximation.** The growing approximation of a left-linear rewrite system $\{l_1 \to r_1 \dots l_n \to r_n\}$ is a growing rewrite system $\{l_1 \to r'_1 \dots l_n \to r'_n\}$ where $r'_i$ $(1 \leq i \leq n)$ is obtained by renaming the variables of $r_i$ which do not match the conditions of growing systems.

*Example 4.* Let us go back to the system of example 3. Its growing approximation is $\{f(x_1, f(x_2, x_3)) \to f(g(x_1, x'_2), x'_3),\ f(x_4, 0) \to x_4\}$.

Growing-sequentiality refers to sequentiality for the growing approximation. To prove the decidability of growing-sequentiality for left-linear systems, we use the following fundamental theorem of Comon [1], which we give a reformulation with the notations of our paper.

***Theorem 15 Comon [1].*** *Consider a right-linear approximation $\tau$. If for any left-linear rewrite system $\mathcal{R}$, the set of ground terms normalizable w.r.t. $\tau(\mathcal{R})$ is recognizable by a tree automaton, then the sequentiality for $\tau$ is decidable.*

To prove this theorem, Comon used Rabin's correspondence between tree automata and the decidable weak second order monadic theory of the finite $k$-ary tree WS$k$S. More precisely, he shows that sequentiality may be expressed in a WS$k$S formula with an additional unary predicate satisfied by grounds terms normalizable w.r.t. $\tau(\mathcal{R})$. This has been used by Comon [1] to show the decidability results mentioned in Sect. 7.3.

***Theorem 16.*** *The growing-sequentiality of any left-linear rewrite system is decidable.*

*Proof.* From Theorems 15 and 5.

### 7.5 A gap between growing sequentiality and the previous notions of sequentiality

To conclude this section and for those who are familiar with sequentiality, we give two examples illustrating the *strict* hierarchy between the approximations we have presented above. They are variation on Gustave example [11].

***Proposition 17.*** *The two rewrite systems of examples 5 and 6 below are respectively shallow-sequential and non NVNF-sequential and on the other hand growing-sequential and non shallow-sequential.*

The hierarchy is depicted in Fig. 1. In this figure, all the non-dashed sets of term rewrite systems are recursive. For a comparison between NV-sequentiality and NVNF-sequentiality, see [16].

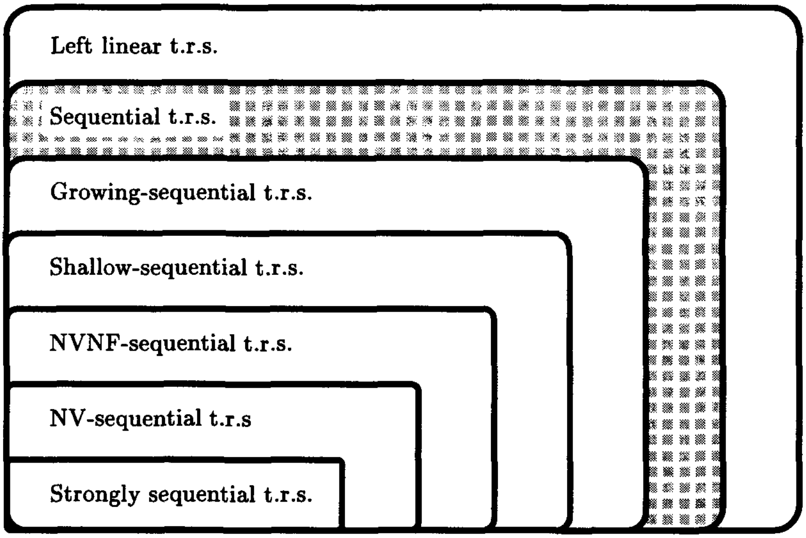


**Fig. 1.** A hierarchy between different sets of sequential rewrite systems

*Example 5.* Let $\mathcal{R} = \{f(x_1, a, b) \to f(a, x_1, a),\ f(b, x_2, a) \to f(a, a, x_2),$ $f(a, b, x_3) \to f(x_3, a, a)\}$ ($f$ is ternary, $a$, $b$ are constants). Its approximation for NVNF-sequentiality of Nagaya et al. [16] is $\mathcal{R}' = \{f(x_1, a, b) \to f(a, x_1', a),$

$f(b, x_2, a) \to f(a, a, x'_2)$, $f(a, b, x_3) \to f(x'_3, a, a)\}$ though its approximation for shallow sequentiality is $\mathcal{R}$ itself. The $\Omega$-term $t := f(\Omega, \Omega, \Omega)$ has no index w.r.t. $\mathcal{R}'$ and is a normal form of $\mathcal{R}'$. Thus $\mathcal{R}'$ is not sequential, $\mathcal{R}$ is not NVNF-sequential. It can be shown that $\mathcal{R}$ is sequential, thus $\mathcal{R}$ is shallow-sequential. To prove this, we may use a result from Huet and Lévy [13] saying that if a left-linear *non-ambiguous* rewrite system $\mathcal{R}$ is such that for each $\Omega$-term $t$ in normal form w.r.t. $\mathcal{R}$ which contains an occurrence of $\Omega$ and such that there exists $u \sqsupseteq t$ with $u \xrightarrow[\mathcal{R}]{*} n \in \mathcal{T}(\mathcal{F})$ ($n$ is in normal form), then $t$ has an index[4].

*Example 6.* Let $\mathcal{R} = \{f(x_1, a, b) \to f(a, f(a, x_1, a), a)$, $f(b, x_2, a) \to f(a, a, f(a, a, x_2))$, $f(a, b, x_3) \to f(f(x_3, a, a), a, a)\}$ Its approximation for shallow-sequentiality is $\mathcal{R}' = \{f(x_1, a, b) \to f(a, f(a, x'_1, a), a)$, $f(b, x_2, a) \to f(a, a, f(a, a, x'_2))$, $f(a, b, x_3) \to f(f(x'_3, a, a), a, a)\}$ though its growing approximation is $\mathcal{R}$ itself. As in example 5, we can show that $\mathcal{R}'$ is not sequential ($\mathcal{R}$ is not shallow-sequential) and that $\mathcal{R}$ is growing-sequential.

# 8 Undecidability Results

To conclude, we show that a further weakening of the conditions of Theorems 10, 16 and Corollary 11 yields in undecidability.

## 8.1 Non-Linear Case

If we relax the linearity condition in Theorem 10, then reachability becomes undecidable.

***Theorem 18.*** *Reachability is undecidable for rewrite systems $\mathcal{R}$ such that for each rule $l \to r \in \mathcal{R}$ and each pair of positions $u \in \mathcal{P}os(l)$, $v \in \mathcal{P}os(r)$ with $l(u) = r(v) \in \mathcal{X}$ we have $|u| \leq 1$.*

*Proof.* (sketch) We describe a reduction of an intance of the (undecidable) Post Correspondence Problem (PCP). Let $A$ be an alphabet and $u_1 \ldots u_n, v_1 \ldots v_n \in A^*$ an instance of PCP. The problem is to find a finite sequence $i_1 \ldots i_m$ of integers smaller than $n$ – which may contain repetitions – such that $u_{i_1} u_{i_2} \cdots u_{i_m} = v_{i_1} v_{i_2} \cdots v_{i_m}$. We let $\mathcal{F} := A \uplus \{f, 0, c, b\}$ where the symbols of $A$ are unary, $f$ is binary and 0, $c$ and $b$ are constants. We note $u_i(x)$ ($1 \leq i \leq n$) for the term $u_{i,1}(u_{i,2}(\ldots(u_{i,k}(x))\ldots))$, if $u_{i,1} \ldots u_{i,k} \in A$ and $u_i = u_{i,1} \ldots u_{i,k}$.

$$\mathcal{R} = \bigcup_{1=1}^{n} c \to f(u_i(0), v_i(0)) \cup \bigcup_{1=1}^{n} f(x_1, x_2) \to f(u_i(x_1), v_i(x_2)) \cup f(x, x) \to b\ .$$

Then the PCP has a solution iff $c \xrightarrow[\mathcal{R}]{*} b$. □

The Corollary 11 neither holds without the linearity condition. Oyamaguchi showed indeed in [18] that the word problem is undecidable for right-ground term rewriting systems, which constitutes a particular case of Corollary 11 without linearity.

[4] This result does not hold for ambiguous left-linear systems

### 8.2 Linear Case with a Weaker Restriction

If, in Theorem 10, we keep the linearity condition but we weaken the restriction about variables sharing, this also results in undecidability.

***Theorem 19.*** *Reachability is undecidable for linear rewrite systems $\mathcal{R}$ such that for each rule $l \to r \in \mathcal{R}$ we have either for each pair of positions $u \in \mathcal{P}os(l)$, $v \in \mathcal{P}os(r)$ such that $l(u) = r(v) \in \mathcal{X}$, $|u| \leq 1$ or for each pair $u \in \mathcal{P}os(l)$, $v \in \mathcal{P}os(r)$ such that $l(u) = r(v) \in \mathcal{X}$, $|v| \leq 1$.*

*Proof.* (sketch) Once again we reduce an instance $A, u_1, \ldots, u_n, v_1, \ldots, v_n \in A^*$ of PCP to a reachability problem for a system $\mathcal{R}$ as in statement of Theorem 19. Let $\mathcal{F} = A \uplus \{f, g, 0, c\}$ with arities where the symbols of $A$ are unary, $f$ and $g$ are binary and $0$ and $c$ are constants.

$$\begin{aligned}\mathcal{R} = \biguplus_{i=1}^{n} c \to f(u_i(0), v_i(0)) \;\cup\; \biguplus_{i=1}^{n} f(x_1, x_2) \to f(u_i(x_1), v_i(x_2)) \\ \cup f(x_1, x_2) \to g(x_1, x_2) \;\cup\; \biguplus_{a \in A} g(a(x_1), a(x_2)) \to g(x_1, x_2)\ .\end{aligned}$$

Then the above PCP has a solution iff $c \xrightarrow[\mathcal{R}]{*} g(0, 0)$. □

Because of the reduction of reachability to sequentiality in [13], sequentiality is also undecidable for the kind of linear rewrite systems of Theorem 19.

## Acknowledgments

I am very grateful to Hubert Comon for his helpful suggestions and remarks as well as for his careful reading of a first version of this paper, and also to the referees for their useful comments.

## References

1. H. 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 Comp. Soc. Press.
2. H. Comon, M. Haberstrau, and J.-P. Jouannaud. Decidable properties of shallow equational theories. In *Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science.* IEEE Comp. Soc. Press, 1992.
3. M. Dauchet. Simulation of a turing machine by a left-linear rewrite rule. In *Proc. 3rd Rewriting Techniques and Applications, Chapel Hill, LNCS 355*, 1989.
4. M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *Proc. 5th IEEE Symp. Logic in Computer Science, Philadelphia*, 1990.
5. M. Dauchet, S. Tison, T. Heuillard, and P. Lescanne. Decidability of the confluence of ground term rewriting systems. Technical Report IT-102, LIFL, Université de Lille, May 1987. Actes de logic in Computer Science Second Annual Conference New-York Juin 87.

6. N. Dershowitz. Termination of linear rewriting systems. In *Proceedings of the Eihgth Int. Colloquium on Automata, Languages and Programming*, pages 448–458, Acre, Israel, 1981.
7. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. North-Holland, 1990.
8. J. Gallier and R. Book. Reductions in tree replacement systems. *Theorical Computer Science*, 37:123–150, 1985.
9. M. Gécseg and M. Steinby. *Tree Automata*. Akademia Kiadó, Budapest, 1984.
10. G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *J. ACM*, 27(4):797–821, Oct. 1980.
11. G. Huet. Formal structures for computation and deduction. In: Working material for the International Summer School on Logic of Programming and Calculi of Discrete Design, Marktoberdorf, May 1986.
12. G. Huet and D. S. Lankford. On the uniform halting problem for term rewriting systems. Research Report 283, INRIA, Mar. 1978.
13. G. Huet and J.-J. Lévy. Computations in orthogonal term rewriting systems. In G. Plotkin and J.-L. Lassez, editors, *Computational Logic: essays in Honour of Alan Robinson*. MIT Press, 1990.
14. G. Huet and J.-J. Lévy. Call by need computations in non-ambiguous linear term rewriting systems. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*. MIT Press, 1991.
15. J. W. Klop and A. Middeldorp. Sequentiality in orthogonal term rewriting systems. *Journal of Symbolic Computation*, 12:161–195, 1991.
16. T. Nagaya, M. Sakai, and Y. Toyama. NVNF-sequentiality of left-linear term rewriting systems. In *Proc. Japanese Workhop on Term Rewriting*, Kyoto, July 1995.
17. M. Oyamaguchi. The reachability problem for quasi-ground term rewriting systems. *J. Inf. Process.*, pages 232–236, 1986.
18. M. Oyamaguchi. On the word problem for right-ground term rewriting systems. *Trans. IEICE Japan*, E73:718–723, 1990.
19. M. Oyamaguchi. NV-sequentiality: a decidable condition for call-by-need computations in term rewriting systems. *SIAM J. Comput.*, 22(1):114–135, 1993.
20. E. L. Post. Recursive unsolvability of a problem of Thue. *Journal of Symbolic Logic*, 13:1–11, 1947.
21. W. Sadfi. *Contribution à l'étude de la séquentialité forte des définitions de fonctions par règles*. Thèse de doctorat, Université Paris-Sud, Orsay, France, Dec. 1993.
22. A. Togashi and S. Noguchi. Some decision problems and their time complexity for term rewriting systems. *Trans. IECE Japan*, J66(D):1177–1184, 1983.
23. Y. Toyama. Strong sequentiality of left linear overlapping term rewriting systems. In *Proc. 7th IEEE Symp. on Logic in Computer Science*, Santa Cruz, CA, 1992.