On One-Pass Term Rewriting *

Zoltán Fülöp[1], Eija Jurvanen[2], Magnus Steinby[3], and Sándor Vágvölgyi[4]

[1] József Attila University, Department of Computer Science, H-6701 Szeged, P. O. Box 652, Hungary, fulop@inf.u-szeged.hu
[2] Turku Centre for Computer Science, DataCity, Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland, jurvanen@utu.fi
[3] Turku Centre for Computer Science, and Department of Mathematics, University of Turku, FIN-20014 Turku, Finland, steinby@utu.fi
[4] József Attila University, Department of Applied Informatics, H-6701 Szeged, P. O. Box 652, Hungary, vagvolgy@inf.u-szeged.hu

Reducing a term with a term rewriting system (TRS) is a highly nondeterministic process and usually no bound for the lengths of the possible reduction sequences can be given in advance. Here we consider two very restrictive strategies of term rewriting, *one-pass root-started rewriting* and *one-pass leaf-started rewriting*. If the former strategy is followed, rewriting starts at the root of the given term t and proceeds continuously towards the leaves without ever rewriting any part of the current term which has been produced in a previous rewrite step. When no more rewriting is possible, a *one-pass root-started normal form* of the term t has been reached. The leaf-started version is similar, but the rewriting is initiated at the leaves and proceeds towards the root. The requirement that rewriting should always concern positions immediately adjacent to parts of the term rewritten in previous steps distinguishes our rewriting strategies from the IO and OI rewriting schemes considered in [5] or [2]. It also implies that the top-down and bottom-up cases are different even for a linear TRS.

Let $\mathcal{R} = (\Sigma, R)$ be a TRS over a ranked alphabet Σ. For any Σ-tree language T, we denote the sets of one-pass root-started sentential forms, one-pass root-started normal forms, one-pass leaf-started sentential forms and one-pass leaf-started normal forms of trees in T by $1r\mathrm{S}_{\mathcal{R}}(T)$, $1r\mathrm{N}_{\mathcal{R}}(T)$, $1\ell\,\mathrm{S}_{\mathcal{R}}(T)$ and $1\ell\,\mathrm{N}_{\mathcal{R}}(T)$, respectively. We show that the following inclusion problems, where $\mathcal{R} = (\Sigma, R)$ is a left-linear TRS and T_1 and T_2 are two regular Σ-tree languages, are decidable.

The one-pass root-started sentential form inclusion problem: $1r\mathrm{S}_{\mathcal{R}}(T_1) \subseteq T_2$?
The one-pass root-started normal form inclusion problem: $1r\mathrm{N}_{\mathcal{R}}(T_1) \subseteq T_2$?
The one-pass leaf-started sentential form inclusion problem: $1\ell\,\mathrm{S}_{\mathcal{R}}(T_1) \subseteq T_2$?
The one-pass leaf-started normal form inclusion problem: $1\ell\,\mathrm{N}_{\mathcal{R}}(T_1) \subseteq T_2$?

In [9] the inclusion problem for ordinary sentential forms is called the second-order reachability problem and the problem is shown to be decidable for a TRS $\mathcal{R}$ which preserves recognizability, i.e. if the set of sentential forms of the trees of

* This research was supported by the exchange program of the University of Turku and the József Attila University, and by the grants MKM 665/96 and FKFP 0095/97.

Luboš Prim et al. (Eds.): MFCS'98, LNCS 1450, pp. 248–256, 1998.

any recognizable tree language T is also recognizable. In our problems the sets of normal forms or sentential forms are not necessarily regular.

Many questions concerning term rewriting systems have been studied using tree automata; cf. [2], [4], [8], [9], [10], [11], [12], for example. We also prove the decidability of the four inclusion problems by reducing them to the emptiness problem of certain finite tree recognizers.

We thank the referees for useful comments.

1 Preliminaries

Here we introduce the basic notions used in the paper, but for more about term rewriting and tree automata, we refer the reader to [1], [3], [6] and [7].

In what follows Σ is a *ranked alphabet.* For each $m \geq 0$, the set of m-ary symbols in Σ is denoted by Σ_m, and Σ is *unary* if $\Sigma = \Sigma_1$. If Y is an alphabet disjoint with Σ, the set $T_\Sigma(Y)$ of *Σ-terms with variables* in Y is the smallest set U including Y such that $f(t_1, \ldots, t_m) \in U$ whenever $m \geq 0$, $f \in \Sigma_m$ and $t_1, \ldots, t_m \in U$. If $c \in \Sigma_0$, we write just c for $c()$. The set $T_\Sigma(\emptyset)$ of *ground Σ-terms* is denoted by T_Σ. Terms are also called *trees.* Ground Σ-terms and subsets of T_Σ are called *Σ-trees* and *Σ-tree languages*, respectively. The *height* $\mathrm{hg}(t)$ of a tree $t \in T_\Sigma(Y)$ is defined so that $\mathrm{hg}(t) = 0$ for $t \in Y \cup \Sigma_0$, and $\mathrm{hg}(t) = \max\{\mathrm{hg}(t_1), \ldots, \mathrm{hg}(t_m)\} + 1$ for $t = f(t_1, \ldots, t_m)$. The set $\mathrm{var}(t)$ $(\subseteq Y)$ of variables appearing in t is also defined as usual (cf. [7]).

Let $X = \{x_1, x_2, \ldots\}$ be a set of variables. For each $n \geq 0$, we put $X_n = \{x_1, \ldots, x_n\}$ and abbreviate $T_\Sigma(X_n)$ to $T_{\Sigma,n}$. A tree $t \in T_{\Sigma,n}$ is *linear* if no variable appears twice in t. The subset $\widetilde{T}_{\Sigma,n}$ of $T_{\Sigma,n}$ is defined so that $t \in T_{\Sigma,n}$ belongs to $\widetilde{T}_{\Sigma,n}$ if and only if each of $x_1, \ldots, x_n$ occurs in t exactly once and their left-to-right order is $x_1, \ldots, x_n$. Also, let $\widetilde{T}_{\Sigma,X} = \bigcup_{n=0}^{\infty} \widetilde{T}_{\Sigma,n}$. If $f \in \Sigma_m$, $m \geq 1$ and $t_1, \ldots, t_m \in \widetilde{T}_{\Sigma,X}$, then $\|f(t_1, \ldots, t_m)\|$ is the tree in $\widetilde{T}_{\Sigma,X}$ obtained from $f(t_1, \ldots, t_m)$ by renaming the variables. If $t \in T_{\Sigma,n}$ and $\sigma\colon X \to T_\Sigma(X)$ is a substitution such that $\sigma(x_i) = t_i$ $(i = 1, \ldots, n)$, we write $\sigma(t) = t[t_1, \ldots, t_n]$.

A *term rewriting system* (TRS) over Σ is a system $\mathcal{R} = (\Sigma, R)$, where R is a finite set of *rewrite rules* $p \to r$ such that $p, r \in T_\Sigma(X)$, $\mathrm{var}(r) \subseteq \mathrm{var}(p)$ and $p \notin X$. A rule $p \to r$ is *ground* if $p, r \in T_\Sigma$. The rewrite relation $\Rightarrow_\mathcal{R}$ on T_Σ induced by $\mathcal{R}$ is defined so that $t \Rightarrow_\mathcal{R} u$ if u is obtained from t by replacing an occurrence of a subtree of t of the form $p[t_1, \ldots, t_n]$ by $r[t_1, \ldots, t_n]$, where $p \to r \in R$, $p, r \in T_{\Sigma,n}$ and $t_1, \ldots, t_n \in T_\Sigma$. The reflexive, transitive closure of $\Rightarrow_\mathcal{R}$ is denoted by $\Rightarrow_\mathcal{R}^*$. Hence $s \Rightarrow_\mathcal{R}^* t$ iff there exists a *reduction sequence*

$$t_0 \Rightarrow_\mathcal{R} t_1 \Rightarrow_\mathcal{R} \ldots \Rightarrow_\mathcal{R} t_n$$

in $\mathcal{R}$ such that $n \geq 0$, $t_0 = s$ and $t_n = t$. Note that we apply a TRS to ground terms only. For any TRS $\mathcal{R} = (\Sigma, R)$, let $\mathrm{lhs}(\mathcal{R}) = \{\, p \mid (\exists r)\, p \to r \in R \,\}$. The TRS $\mathcal{R}$ is *left-linear* if every p in $\mathrm{lhs}(\mathcal{R})$ is linear, and it is then *in standard form* if $\mathrm{lhs}(\mathcal{R}) \subseteq \widetilde{T}_{\Sigma,X}$. A tree $s \in T_\Sigma$ is *irreducible* with respect to $\mathcal{R}$ if $s \Rightarrow_\mathcal{R} u$ for no u, and it is a *normal form* of a Σ-tree t if it is irreducible and $t \Rightarrow_\mathcal{R}^* s$.

In a *top-down Σ-recognizer* $\mathcal{A} = (A, \Sigma, P, a_0)$ (1) A is a (finite) unary ranked alphabet of *states* such that $A \cap \Sigma = \emptyset$, (2) P is a finite set of *transition rules*, each of the form $a(f(x_1, \ldots, x_m)) \to f(a_1(x_1), \ldots, a_m(x_m))$, also written simply $a(f) \to f(a_1, \ldots, a_m)$, where $m \geq 0$, $f \in \Sigma_m$ and $a, a_1, \ldots, a_m \in A$, and (3) $a_0 \in A$ is the *initial state*. We treat $\mathcal{A}$ as the TRS $(\Sigma \cup A, P)$ and the rewrite relation $\Rightarrow_{\mathcal{A}} \subseteq T_{\Sigma \cup A} \times T_{\Sigma \cup A}$ is defined accordingly. For each $a \in A$, let $T(\mathcal{A}, a) = \{ t \in T_\Sigma \mid a(t) \Rightarrow_{\mathcal{A}}^* t \}$. The tree language *recognized* by $\mathcal{A}$ is the set $T(\mathcal{A}) = T(\mathcal{A}, a_0)$. A tree language $T \subseteq T_\Sigma$ is *recognizable*, or *regular*, if $T(\mathcal{A}) = T$ for a top-down Σ-recognizer $\mathcal{A}$.

In a *generalized top-down Σ-recognizer* $\mathcal{A} = (A, \Sigma, P, a_0)$ the rewrite rules of P are of the form $a(t(x_1, \ldots, x_n)) \to t[a_1(x_1), \ldots, a_n(x_n)]$, where $n \geq 0$, $a, a_1, \ldots, a_n \in A$, and $t \in \widetilde{T}_{\Sigma,n}$. The relations $\Rightarrow_{\mathcal{A}}$, $\Rightarrow_{\mathcal{A}}^*$, and the set $T(\mathcal{A})$ are defined as in a top-down Σ-recognizer.

A *bottom-up Σ-recognizer* is a quadruple $\mathcal{A} = (A, \Sigma, P, A_f)$, where (1) A is a finite set of *states* of rank 0, $\Sigma \cap A = \emptyset$, (2) P is a finite set of *transition rules*, of the form $f(a_1, \ldots, a_m) \to a$ with $m \geq 0$, $f \in \Sigma_m$, $a_1, \ldots, a_m$, $a \in A$, and (3) A_f $(\subseteq A)$ is the set of *final states*. We say that $\mathcal{A}$ is *total deterministic* if for all $f \in \Sigma_m$, $m \geq 0$, $a_1, \ldots, a_m \in A$, there is exactly one rule of the form $f(a_1, \ldots, a_m) \to a$. We treat $\mathcal{A}$ as the rewriting system $(\Sigma \cup A, P)$, and the tree language recognized by it can be defined as the set $T(\mathcal{A}) = \{ t \in T_\Sigma \mid (\exists\, a \in A_f)\ t \Rightarrow_{\mathcal{A}}^* a \}$. For any bottom-up Σ-recognizer $\mathcal{A}$, one can effectively construct a total deterministic bottom-up Σ-recognizer $\mathcal{B}$ such that $T(\mathcal{A}) = T(\mathcal{B})$.

In a *generalized bottom-up Σ-recognizer* $\mathcal{A} = (A, \Sigma, P, A_f)$ P is a finite set of rewrite rules $t[a_1, \ldots, a_n] \to a$, where $n \geq 0$, $t \in \widetilde{T}_{\Sigma,n}$ and $a_1, \ldots, a_n, a \in A$. The tree language recognized by $\mathcal{A}$ is $T(\mathcal{A}) = \{ t \in T_\Sigma \mid (\exists\, a \in A_f)\ t \Rightarrow_{\mathcal{A}}^* a \}$.

It is easy to see that both generalized top-down and bottom-up Σ-recognizers recognize exactly the regular Σ-tree languages. Moreover, the emptiness problem "$T(\mathcal{A}) = \emptyset$?" is obviously decidable for both types of automata.

2 One-Pass Term Rewriting

The first of our two modes of one-pass rewriting may be described as follows.

Let $\mathcal{R} = (\Sigma, R)$ be a TRS and t the Σ-tree to be rewritten. The portion of t first rewritten should include the root. Rewriting then proceeds towards the leaves so that each rewrite step applies to a root segment of a maximal unprocessed subtree but never involves any part of the tree produced by a previous rewrite step. For the formal definition we associate with $\mathcal{R}$ a TRS in which a new special symbol forces this mode of rewriting.

Definition 2.1. *The* one-pass root-started TRS *associated with a given TRS $\mathcal{R} = (\Sigma, R)$ is the TRS $\mathcal{R}_\# = (\Sigma \cup \{\#\}, R_\#)$, where $\#$ is a new unary symbol, the* separator mark, *and $R_\#$ is the set of all rewrite rules*

$$\#(p(x_1, \ldots, x_n)) \to r[\#(x_1), \ldots, \#(x_n)]$$

obtained from a rule $p \to r$ in R, where $p, r \in T_{\Sigma,n}$, by adding $\#$ to the root of the left-hand side and above the variables in the right-hand side.

Example 2.1. If $R = \{ f(g(x_1), x_2) \to f(x_1, g(x_2)),\ g(x_1) \to g(c) \}$, where $f \in \Sigma_2$, $g \in \Sigma_1$, and $c \in \Sigma_0$, then

$$R_\# = \{ \#(f(g(x_1), x_2)) \to f(\#(x_1), g(\#(x_2))),\ \#(g(x_1)) \to g(c) \} \ .$$

For any TRS $\mathcal{R}$, the associated one-pass root-started TRS $\mathcal{R}_\#$ is terminating. For recovering the one-pass root-started reduction sequences of $\mathcal{R}$ from the reduction sequences of $\mathcal{R}_\#$, we introduce the tree homomorphism $\delta\colon T_{\Sigma\cup\{\#\}} \to T_\Sigma$ which just erases the separator marks. If

$$\#(t) \Rightarrow_{\mathcal{R}_\#} t_1 \Rightarrow_{\mathcal{R}_\#} t_2 \Rightarrow_{\mathcal{R}_\#} \ldots \Rightarrow_{\mathcal{R}_\#} t_k$$

is a reduction sequence with $\mathcal{R}_\#$ starting from some $t \in T_\Sigma$, then

$$t \Rightarrow_{\mathcal{R}} \delta(t_1) \Rightarrow_{\mathcal{R}} \delta(t_2) \Rightarrow_{\mathcal{R}} \ldots \Rightarrow_{\mathcal{R}} \delta(t_k)$$

is a *one-pass root-started reduction sequence* with $\mathcal{R}$. The terms $t, \delta(t_1), \ldots, \delta(t_k)$ are called *one-pass root-started sentential forms* of t in $\mathcal{R}$. If t_k is irreducible in $\mathcal{R}_\#$, then $\delta(t_k)$ is a *one-pass root-started normal form* of t in $\mathcal{R}$. The sets of all one-pass root-started sentential forms and normal forms of a Σ-tree t are denoted by $1r\mathrm{S}_\mathcal{R}(t)$ and $1r\mathrm{N}_\mathcal{R}(t)$, respectively. This notation is extended to sets of Σ-trees in the natural way.

Note that for any TRS $\mathcal{R} = (\Sigma, R)$ and any $t \in T_\Sigma$, the sets $1r\mathrm{S}_\mathcal{R}(t)$ and $1r\mathrm{N}_\mathcal{R}(t)$ are finite and effectively computable but that $1r\mathrm{S}_\mathcal{R}(T)$ and $1r\mathrm{N}_\mathcal{R}(T)$ are not necessarily regular even for a regular Σ-tree language T.

The one-pass TRS used for defining the one-pass leaf-started rewriting mode of a given TRS is constructed in two stages.

Definition 2.2. *Let $\mathcal{R} = (\Sigma, R)$ be a TRS. First we extend R to the set R_e of all rules*

$$p[y_1, \ldots, y_n] \to r[y_1, \ldots, y_n]$$

such that $p \to r \in R$ with $p,\ r \in T_{\Sigma,n}$, and for each i, $1 \le i \le n$, either $y_i \in X$ or $y_i \in \Sigma_0$, and $p[y_1, \ldots, y_n] \in \widetilde{T}_{\Sigma,X}$. Now let $\Sigma' = \{ f' \mid f \in \Sigma \}$ be a disjoint copy of Σ such that for any $f \in \Sigma$, f and f' have the same rank. The one-pass leaf-started TRS *associated with $\mathcal{R}$ is the TRS $\mathcal{R}^\# = (\Sigma \cup \Sigma' \cup \{\#\}, R^\#)$, where $\#$ is a new unary symbol, the* separator mark, *and $R^\#$ consists of all rules*

$$p[\#(x_1), \ldots, \#(x_n)] \to \#(r'(x_1, \ldots, x_n)) \ ,$$

where $p \to r \in R_e$, with $p,\ r \in T_{\Sigma,n}$, and r' is obtained from r by replacing every symbol $f \in \Sigma$ by the corresponding symbol f' in Σ'.

Example 2.2. Let $R = \{f(g(x_1), x_2) \to f(x_1, c),\ g(c) \to c\}$, where $\Sigma = \{f, g, c\}$, $f \in \Sigma_2$, $g \in \Sigma_1$, and $c \in \Sigma_0$. Then $\Sigma' = \{f', g', c'\}$ and the one-pass leaf-started TRS associated with $\mathcal{R} = (\Sigma, R)$ is the TRS $\mathcal{R}^\# = (\Sigma \cup \Sigma' \cup \{\#\}, R^\#)$ where $R^\#$ consists of the five rules

$$f(g(\#(x_1)), \#(x_2)) \to \#(f'(x_1, c')),\ f(g(c), \#(x_1)) \to \#(f'(c', c')),$$
$$f(g(\#(x_1)), c) \to \#(f'(x_1, c')),\ f(g(c), c) \to \#(f'(c', c')),\ g(c) \to \#(c') \ .$$

Clearly, $\Rightarrow_{\mathcal{R}_e} = \Rightarrow_{\mathcal{R}}$. The reduction sequences of $\mathcal{R}^{\#}$ represent reduction sequences of $\mathcal{R}$ which start at the leaves of a term and proceed towards the root of it so that symbols introduced by a previous rewrite step never form a part of the left-hand side of the rule applied next. Moreover, $\mathcal{R}^{\#}$ passes only once over the term because the left-hand sides and the right-hand sides of its rules share only the symbol #. The corresponding one-pass reduction sequence of $\mathcal{R}$ is recovered by applying the tree homomorphism $\delta : T_{\Sigma \cup \Sigma' \cup \{\#\}} \to T_{\Sigma}$ which erases the #-marks and the primes from the symbols $f' \in \Sigma'$. Then each reduction sequence

$$t \Rightarrow_{\mathcal{R}^{\#}} t_1 \Rightarrow_{\mathcal{R}^{\#}} t_2 \Rightarrow_{\mathcal{R}^{\#}} \ldots \Rightarrow_{\mathcal{R}^{\#}} t_k$$

with $\mathcal{R}^{\#}$ yields the *one-pass leaf-started reduction sequence*

$$t \Rightarrow_{\mathcal{R}} \delta(t_1) \Rightarrow_{\mathcal{R}} \delta(t_2) \Rightarrow_{\mathcal{R}} \ldots \Rightarrow_{\mathcal{R}} \delta(t_k)$$

with $\mathcal{R}$. The terms $t, \delta(t_1), \ldots, \delta(t_k)$ are called *one-pass leaf-started sentential forms* of t in $\mathcal{R}$. If t_k is irreducible in $\mathcal{R}^{\#}$, then $\delta(t_k)$ is a *one-pass leaf-started normal form* of t in $\mathcal{R}$. The sets of all one-pass leaf-started sentential forms and normal forms of a Σ-tree t are denoted by $1\ell \mathrm{S}_{\mathcal{R}}(t)$ and $1\ell \mathrm{N}_{\mathcal{R}}(t)$, respectively. This notation is extended to sets of Σ-trees in the natural way.

Note that without the new rules of the extended TRS $\mathcal{R}_e$ many natural one-pass leaf-started rewriting sequences of $\mathcal{R}$ could be missed.

3 The One-Pass Root-Started Inclusion Problems

First we consider the one-pass root-started normal form inclusion problem. It is assumed that the tree languages are given as tree recognizers.

Theorem 3.1. *For any left-linear TRS $\mathcal{R} = (\Sigma, R)$, the following* one-pass root-started normal form inclusion problem *is decidable.*

Instance: *Recognizable Σ-tree languages T_1 and T_2.*
Question: $1r\mathrm{N}_{\mathcal{R}}(T_1) \subseteq T_2$*?*

For proving Theorem 3.1, we need the following auxiliary notation. For a set A of unary symbols such that $A \cap \Sigma = \emptyset$ and any alphabet Y, let $T_{\Sigma}(A(Y))$ be the least subset T of $T_{\Sigma \cup A}(Y)$ for which (1) $a(y) \in T$ for all $a \in A$, $y \in Y$, and (2) $m \geq 0$, $f \in \Sigma_m$, $t_1, \ldots, t_m \in T$ implies $f(t_1, \ldots, t_m) \in T$.

Let $\mathcal{A} = (A, \Sigma, P, a_0)$ be a top-down Σ-recognizer. For any $a \in A$, $n \geq 0$ and any $t \in T_{\Sigma,n}$, the set $\mathcal{A}(a,t)$ $(\subseteq T_{\Sigma}(A(X_n)))$ is defined so that (1) for $x_i \in X_n$, $\mathcal{A}(a, x_i) = \{a(x_i)\}$, (2) for $c \in \Sigma_0$, $\mathcal{A}(a,c) = \{c\}$ if $a(c) \to c \in P$, and $\mathcal{A}(a,c) = \emptyset$ otherwise, and (3) for $t = f(t_1, \ldots, t_m)$, $\mathcal{A}(a,t) =$

$$\{f(s_1, \ldots, s_m) \mid s_1 \in \mathcal{A}(a_1, t_1), \ldots, s_m \in \mathcal{A}(a_m, t_m),\ a(f) \to f(a_1, \ldots, a_m) \in P\}\ .$$

For any $s \in T_{\Sigma}(A(X))$ and any variable $x_i \in X$, we denote by $\mathrm{st}(s, x_i)$ the set of states $b \in A$ such that $b(x_i)$ appears as a subterm in s.

Clearly, $\mathcal{A}(a,t) \neq \emptyset$ iff there is a computation of $\mathcal{A}$ which starts in state a at the root of t, continues to the leaves of t, and if $\mathcal{A}$ reaches in a state b a leaf labelled by a nullary symbol c, then $b(c) \to c$ is in P. Each $s \in \mathcal{A}(a,t)$ represents the situation when such a successful computation has been completed so that all leaves labelled with a nullary symbol have also been processed. If $t \in \widetilde{T}_{\Sigma,n}$, then every $s \in \mathcal{A}(a_0,t)$ is of the form $s = t[a_1(x_1),\ldots,a_n(x_n)]$ and for any $t_1, \ldots, t_n \in T_\Sigma$, the tree s appears in a computation of $\mathcal{A}$ on $t[t_1,\ldots,t_n]$ of the form

$$a_0(t[t_1,\ldots,t_n]) \Rightarrow_{\mathcal{A}}^* t[a_1(t_1),\ldots,a_n(t_n)] = s[t_1,\ldots,t_n] \Rightarrow_{\mathcal{A}}^* \cdots$$

in which each subterm t_i is processed starting in the corresponding state a_i. However, if t is not linear, then a variable x_i may appear in a term $s \in \mathcal{A}(a_0,t)$ together with more than one state symbol, and then the corresponding subterm t_i should be accepted by a computation starting with each $a \in \mathrm{st}(s,x_i)$.

Proof (of Theorem 3.1). Consider a left-linear TRS $\mathcal{R} = (\Sigma, R)$ and any recognizable Σ-tree languages T_1 and T_2. Let $\mathcal{A} = (A, \Sigma, P_1, a_0)$ and $\mathcal{B} = (B, \Sigma, P_2, b_0)$ be top-down Σ-recognizers for which $T(\mathcal{A}) = T_1$ and $T(\mathcal{B}) = T_2^c$ $(= T_\Sigma \setminus T_2)$. We construct a generalized top-down Σ-recognizer $\mathcal{C}$ such that for any $t \in T_\Sigma$,

$$t \in T(\mathcal{C}) \quad \text{iff} \quad t \in T(\mathcal{A}) \text{ and } s \in T(\mathcal{B}) \text{ for some } s \in 1r\mathrm{N}_{\mathcal{R}}(t) \ . \tag{1}$$

Then $1r\mathrm{N}_{\mathcal{R}}(T_1) \subseteq T_2$ iff $T(\mathcal{C}) = \emptyset$, and the latter condition is decidable.

Let $\mathcal{C} = (C, \Sigma, P, (a_0, \{b_0\}))$ be the generalized top-down Σ-recognizer with the state set $C = (A \times \wp(B)) \cup (\bar{A} \times \wp(B))$, where $\wp(B)$ is the power set of B and $\bar{A} = \{\, \bar{a} \mid a \in A \,\}$ is a disjoint copy of A, and the set P of transition rules is defined as follows. The rules are of three different types.

Type 1. If $p \to r$ is a rule in R and $(a,H) \in A \times \wp(B)$, where $H = \{b_1,\ldots,b_k\}$, we include in P any rule

$$(a,H)(p(x_1,\ldots,x_n)) \to p[(a_1,H_1)(x_1),\ldots,(a_n,H_n)(x_n)] \ ,$$

where $p[a_1(x_1),\ldots,a_n(x_n)] \in \mathcal{A}(a,p)$ and there are terms $s_1 \in \mathcal{B}(b_1,r)$, $\ldots$, $s_k \in \mathcal{B}(b_k,r)$ such that $H_i = \mathrm{st}(s_1,x_i) \cup \ldots \cup \mathrm{st}(s_k,x_i)$ for all $i = 1,\ldots,n$. For $H = \emptyset$ $(k = 0)$, this is interpreted to mean that $H_1 = \ldots = H_n = \emptyset$ should hold, and if $p \to r$ is a ground rule $(n = 0)$, we include $(a,H)(p) \to p$ in P iff $a(p) \Rightarrow_{\mathcal{A}}^* p$ and $b_i(r) \Rightarrow_{\mathcal{B}}^* r$ for all $i = 1,\ldots,k$.

Type 2. Let NI be the set of all terms $q \in \widetilde{T}_{\Sigma,X}$ such that (1) $\mathrm{hg}(q) \leq \max\{\, \mathrm{hg}(p) \mid p \in \mathrm{lhs}(\mathcal{R}) \,\} + 1$, and (2) $\sigma(q) \neq \sigma'(p)$ for all $p \in \mathrm{lhs}(\mathcal{R})$ and all substitutions σ and σ'. For each $p(x_1,\ldots,x_n) \in \mathrm{NI}$ and any $(a,H) \in A \times \wp(B)$ with $H = \{b_1,\ldots,b_k\}$, we include in P any rule

$$(a,H)(p(x_1,\ldots,x_n)) \to p[(\bar{a}_1,H_1)(x_1),\ldots,(\bar{a}_n,H_n)(x_n)] \ ,$$

where $p[a_1(x_1),\ldots,a_n(x_n)] \in \mathcal{A}(a,p)$, and there are terms $s_1 \in \mathcal{B}(b_1,p)$, $\ldots$, $s_k \in \mathcal{B}(b_k,p)$ such that $H_i = \mathrm{st}(s_1,x_i) \cup \ldots \cup \mathrm{st}(s_k,x_i)$ for all $i = 1,\ldots,n$. The cases $H = \emptyset$ and $n = 0$ are treated similarly as above.

Type 3. For each $(\bar{a},H) \in \bar{A} \times \wp(B)$, where $H = \{b_1,\ldots,b_k\}$, we add to P rules as follows.

(i) For $c \in \Sigma_0$, we include in P the rule $(\bar{a}, H)(c) \to c$ iff $a(c) \to c$ is in P_1 and P_2 contains $b_i(c) \to c$ for every $b_i \in H$.
(ii) For $f \in \Sigma_m$, $m > 0$, we add to P all rules

$$(\bar{a}, H)(f(x_1, \ldots, x_m)) \to f((\bar{a}_1, H_1)(x_1), \ldots, (\bar{a}_m, H_m)(x_m)) \ ,$$

where $a(f(x_1, \ldots, x_m)) \to f(a_1(x_1), \ldots, a_m(x_m))$ is in P_1, and there are rules $b_i(f(x_1, \ldots, x_m)) \to f(b_{i1}(x_1), \ldots, b_{im}(x_m))$ $(i = 1, \ldots, k)$ in P_2 such that $H_j = \{b_{1j}, \ldots, b_{kj}\}$ for each $j = 1, \ldots, m$.

We can show that $\mathcal{C}$ has the property described in (1). If $t \in T(\mathcal{C})$, then $(a_0, \{b_0\})(t) \Rightarrow_{\mathcal{C}}^* t$ and this derivation can be split into two parts

$$(a_0, \{b_0\})(t) \Rightarrow_{\mathcal{C}}^* \tilde{t}[(a_1, H_1)(t_1), \ldots, (a_n, H_n)(t_n)] \Rightarrow_{\mathcal{C}}^* \tilde{t}[t_1, \ldots, t_n] = t \ , \qquad (2)$$

where $n \geq 0$, $t \in \widetilde{T}_{\Sigma,n}$ and, for every $1 \leq i \leq n$, $t_i \in T_\Sigma$ and $(a_i, H_i) \in A \times \wp(B)$. In the first part of (2) only Type 1 rules are used, and hence $\tilde{t}[a_1(x_1), \ldots, a_n(x_n)] \in \mathcal{A}(a_0, \tilde{t})$. Moreover, for some $k \geq 0$, $\tilde{s} \in \widetilde{T}_{\Sigma,k}$, and $s_1, \ldots, s_k \in T_\Sigma$,

$$\#(t) = \#(\tilde{t}[t_1, \ldots, t_n]) \Rightarrow_{\mathcal{R}_\#} \ldots \Rightarrow_{\mathcal{R}_\#} \tilde{s}[\#(s_1), \ldots, \#(s_k)] = s \ ,$$

where every s_j is a copy of exactly one of the t_i. (Of course, s_j may be equal to more than one t_i.) For each $i = 1, \ldots, n$, let $K(i) = \{\, j \mid s_j \text{ is a copy of } t_i \,\}$. Then for some $u \in \mathcal{B}(b_0, \tilde{s})$, $H_i = \bigcup\{\, \mathrm{st}(u, x_j) \mid j \in K(i) \,\}$ for all $i = 1, \ldots, n$.

In the second part of (2), it is first checked using Type 2 rules that $\tilde{s}[s_1, \ldots, s_k] \in 1r\mathrm{N}_{\mathcal{R}}(t)$, and the computations $(a_i, H_i)(t_i) \Rightarrow_{\mathcal{C}}^* t_i$ are finished using Type 3 rules. That means for every $i = 1, \ldots, n$, that (a) $t_i \in T(\mathcal{A}, a_i)$ and (b) $t_i \in T(\mathcal{B}, b)$ for all $b \in H_i$. Therefore

$$a_0(t) \Rightarrow_{\mathcal{A}}^* \tilde{t}[a_1(t_1), \ldots, a_n(t_n)] \Rightarrow_{\mathcal{A}}^* \tilde{t}[t_1, \ldots, t_n] = t$$

and there are $b_1, \ldots, b_k \in B$ such that

$$b_0(\tilde{s}[s_1, \ldots, s_k]) \Rightarrow_{\mathcal{B}}^* \tilde{s}[b_1(s_1), \ldots, b_k(s_k)] \Rightarrow_{\mathcal{B}}^* \tilde{s}[s_1, \ldots, s_k] \ .$$

The converse of (1) can be proved similarly. □

The corresponding result for sentential forms can be proved by modifying suitably the definition of the recognizer $\mathcal{C}$.

Theorem 3.2. *For any left-linear TRS $\mathcal{R} = (\Sigma, R)$, the following* one-pass root-started sentential form inclusion problem *is decidable.*

Instance: *Recognizable Σ-tree languages T_1 and T_2.*
Question: $1r\mathrm{S}_{\mathcal{R}}(T_1) \subseteq T_2$*?* □

4 The One-Pass Leaf-Started Inclusion Problems

Now we consider the one-pass leaf-started sentential form inclusion problem. Again the tree languages are assumed to be given in the form of tree recognizers.

Theorem 4.1. *For any left-linear TRS* $\mathcal{R} = (\Sigma, R)$, *the following* one-pass leaf-started sentential form inclusion problem *is decidable.*

Instance: *Recognizable* Σ*-tree languages* T_1 *and* T_2.

Question: $1\ell\,\mathrm{S}_{\mathcal{R}}(T_1) \subseteq T_2$?

Proof. Let $\mathcal{A} = (A, \Sigma, P_1, A_f)$ and $\mathcal{B} = (B, \Sigma, P_2, B_f)$ be bottom-up Σ-recognizers that recognize T_1 and T_2, respectively. We may assume that $\mathcal{B}$ is total deterministic. We construct a generalized bottom-up Σ-recognizer $\mathcal{C} = (C, \Sigma, P, C_f)$ such that $T(\mathcal{C}) = \emptyset$ iff $1\ell\,\mathrm{S}_{\mathcal{R}}(T_1) \subseteq T_2$ as follows.

Let $C = (A \times B) \cup (\bar{A} \times \bar{B})$, where $\bar{A} = \{\, \bar{a} \mid a \in A \,\}$ and $\bar{B} = \{\, \bar{b} \mid b \in B \,\}$, and let $C_f = \{\, \bar{a} \mid a \in A_f \,\} \times \{\, \bar{b} \mid b \in (B \setminus B_f) \,\}$. The set P consists of the following rules which are of three different types.

Type 1. For every $p \to r \in R_e$ with $p, r \in T_{\Sigma,n}$, $n \geq 0$, and for all $a_1, \ldots, a_n$, $a \in A$, $b_1, \ldots, b_n$, $b \in B$ such that $p[a_1, \ldots, a_n] \Rightarrow_{\mathcal{A}}^* a$ and $r[b_1, \ldots, b_n] \Rightarrow_{\mathcal{B}}^* b$, let P contain the rule $p[(a_1, b_1), \ldots, (a_n, b_n)] \to (a, b)$.

Type 2. For all $a \in A$ and $b \in B$, let $(a, b) \to (\bar{a}, \bar{b})$ be in P.

Type 3. For all $f \in \Sigma_m$, $m \geq 0$, $f(a_1, \ldots, a_m) \to a \in P_1$ and $f(b_1, \ldots, b_m) \to b \in P_2$, let P contain $f((\bar{a}_1, \bar{b}_1), \ldots, (\bar{a}_m, \bar{b}_m)) \to (\bar{a}, \bar{b})$.

The way $\mathcal{C}$ processes a Σ-tree t can be described as follows. First $\mathcal{C}$, using rules of Type 1, follows some one-pass leaf-started rewriting sequences by $\mathcal{R}$ on subtrees of t computing in the first components of its states the evaluations by $\mathcal{A}$ of these subtrees and in the second components the evaluations by $\mathcal{B}$ of the translations of the subtrees produced by these one-pass leaf-started rewriting sequences. At any time $\mathcal{C}$ may switch by rules of Type 2 to a mode in which it by rules of Type 3 computes in the first components of its states the evaluation by $\mathcal{A}$ of t and in the second components the evaluation by $\mathcal{B}$ of the one-pass leaf-started sentential form of t produced by $\mathcal{R}$ when the rewriting sequences on the subtrees are combined. This means that for any $t \in T_\Sigma$, $a \in A$ and $b \in B$,

$$t \Rightarrow_{\mathcal{C}}^* (\bar{a}, \bar{b}) \quad \text{iff} \quad t \Rightarrow_{\mathcal{A}}^* a \text{ and } s \Rightarrow_{\mathcal{B}}^* b \text{ for some } s \in 1\ell\,\mathrm{S}_{\mathcal{R}}(t) \ ,$$

which, by recalling the definition of C_f, implies immediately that $T(\mathcal{C}) = \emptyset$ iff $1\ell\,\mathrm{S}_{\mathcal{R}}(T_1) \subseteq T_2$, as required. □

Finally, we turn to one-pass leaf-started normal forms.

Theorem 4.2. *For any left-linear TRS* $\mathcal{R} = (\Sigma, R)$, *the following* one-pass leaf-started normal form inclusion problem *is decidable.*

Instance: *Recognizable* Σ*-tree languages* T_1 *and* T_2.

Question: $1\ell\,\mathrm{N}_{\mathcal{R}}(T_1) \subseteq T_2$?

Proof. Let $\mathcal{A} = (A, \Sigma, P_1, A_f)$ and $\mathcal{B} = (B, \Sigma, P_2, B_f)$ be total deterministic bottom-up Σ-recognizers such that $T(\mathcal{A}) = T_1$ and $T(\mathcal{B}) = T_2$. We construct a generalized bottom-up Σ-recognizer $\mathcal{C} = (C, \Sigma, P, C_f)$ such that $T(\mathcal{C}) = \emptyset$ iff $1\ell\,\mathrm{N}_{\mathcal{R}}(T_1) \subseteq T_2$ as follows.

Let $mx = \max\{\, \mathrm{hg}(p) \mid p \in \mathrm{lhs}(\mathcal{R}_e) \,\}$ and $T_{mx} = \{\, t \in \widetilde{T}_{\Sigma,X} \mid \mathrm{hg}(t) \leq mx \,\}$. Now let $C = (A \times B) \cup (\bar{A} \times \bar{B} \times (T_{mx} \cup \{ok\}))$, where $\bar{A} = \{\, \bar{a} \mid a \in A \,\}$ and

$\bar{B} = \{\, \bar{b} \mid b \in B \,\}$, and $C_f = \{\, \bar{a} \mid a \in A_f \,\} \times \{\, \bar{b} \mid b \in (B \setminus B_f) \,\} \times (T_{mx} \cup \{ok\})$. The set P consists of the following rules of five different types.

Type 1. For every rule $p \to r \in R_e$ with $p,\ r \in T_{\Sigma,n}$, $n \geq 0$, and any states $a_1, \ldots, a_n$, $a \in A$, $b_1, \ldots, b_n$, $b \in B$ such that $p[a_1, \ldots, a_n] \Rightarrow_{\mathcal{A}}^* a$ and $r[b_1, \ldots, b_n] \Rightarrow_{\mathcal{B}}^* b$, let P contain the rule $p[(a_1, b_1), \ldots, (a_n, b_n)] \to (a, b)$.

Type 2. For all $a \in A$ and $b \in B$, let $(a, b) \to (\bar{a}, \bar{b}, x_1)$ be in P.

Type 3. For all $f \in \Sigma_m$, $m \geq 0$, $u_1, \ldots, u_m$, $u \in T_{mx}$, $f(a_1, \ldots, a_m) \to a \in P_1$ and $f(b_1, \ldots, b_m) \to b \in P_2$ such that $u = \|f(u_1, \ldots, u_m)\|$ and $u \in (T_{mx} \setminus \mathrm{lhs}(\mathcal{R}_e))$, let P contain $f((\bar{a}_1, \bar{b}_1, u_1), \ldots, (\bar{a}_m, \bar{b}_m, u_m)) \to (\bar{a}, \bar{b}, u)$. For $m = 0$, we get $f \to (\bar{a}, \bar{b}, f)$.

Type 4. For any $f \in \Sigma_m$, $m \geq 0$, $f(a_1, \ldots, a_m) \to a \in P_1$, $f(b_1, \ldots, b_m) \to b \in P_2$ and $u_1, \ldots, u_m \in T_{mx}$ such that $\|f(u_1, \ldots, u_m)\| \notin T_{mx}$, let P contain the rule $f((\bar{a}_1, \bar{b}_1, u_1), \ldots, (\bar{a}_m, \bar{b}_m, u_m)) \to (\bar{a}, \bar{b}, ok)$.

Type 5. For any $f \in \Sigma_m$ with $m \geq 1$, $a_1, \ldots, a_m$, $a \in A$, $b_1, \ldots, b_m$, $b \in B$, and sequence $y_1, \ldots, y_m \in T_{mx} \cup \{ok\}$ such that $ok \in \{y_1, \ldots, y_m\}$, $f(a_1, \ldots, a_m) \to a \in P_1$, $f(b_1, \ldots, b_m) \to b \in P_2$, let P contain the rule $f((\bar{a}_1, \bar{b}_1, y_1), \ldots, (\bar{a}_m, \bar{b}_m, y_m)) \to (\bar{a}, \bar{b}, ok)$.

It can now be shown that for any $t \in T_\Sigma$, $a \in A$, $b \in B$ and $y \in T_{mx} \cup \{ok\}$,

$$t \Rightarrow_{\mathcal{C}}^* (\bar{a}, \bar{b}, y) \quad \text{iff} \quad t \Rightarrow_{\mathcal{A}}^* a \text{ and } s \Rightarrow_{\mathcal{B}}^* b \text{ for some } s \in 1\ell\mathrm{N}_{\mathcal{R}}(t)\ ,$$

and hence $T(\mathcal{C}) = \emptyset$ iff $1\ell\mathrm{N}_{\mathcal{R}}(T_1) \subseteq T_2$. □

References

1. J. Avenhaus. *Reduktionssysteme.* Springer, 1995.
2. M. Dauchet and F. De Comite. A gap between linear and non-linear term-rewriting systems. In *RTA-87*, LNCS **256**. Springer, 1987, 95–104.
3. N. Dershowitz and J.-P. Jouannaud. *Rewrite Systems*, volume B of *Handbook of Theoretical Computer Science*, chapter 6, pages 243–320. Elsevier, 1990.
4. A. Deruyver and R. Gilleron. The reachability problem for ground TRS and some extensions. In *TAPSOFT'89*, LNCS **351**. Springer, 1989, 227–243.
5. J. Engelfriet and E. M. Schmidt. IO and OI. Part I. *J. Comput. Syst. Sci.*, 15(3):328–353, 1977. Part II. *J. Comput. Syst. Sci.*, 16(1):67–99, 1978.
6. F. Gécseg and M. Steinby. *Tree automata.* Akadémiai Kiadó, Budapest, 1984.
7. F. Gécseg and M. Steinby. *Tree Languages*, volume 3 of *Handbook of Formal Languages*, chapter 1, pages 1–68. Springer, 1997.
8. R. Gilleron. Decision problems for term rewriting systems and recognizable tree languages. In *STACS'91*, LNCS **480**. Springer, 1991, 148–159.
9. R. Gilleron and S. Tison. Regular tree languages and rewrite systems. *Fundam. Inf.*, 24(1,2):157–175, 1995.
10. D. Hofbauer and M. Huber. Linearizing term rewriting systems using test sets. *J. Symb. Comput.*, 17(1):91–129, 1994.
11. G. Kucherov and M. Tajine. Decidability of regularity and related properties of ground normal form languages. *Inf. Comput.*, 118(1):91–100, 1995.
12. S. Vágvölgyi and R. Gilleron. For a rewrite system it is decidable whether the set of irreducible, ground terms is recognizable. *Bull. EATCS*, 48:197–209, 1992.