ELSEVIER

Theoretical Computer Science 194 (1998) 87–122

Theoretical Computer Science

Linear generalized semi-monadic rewrite systems effectively preserve recognizability[1]

Pál Gyenizse[a], Sándor Vágvölgyi[b,*]

[a] *Department of Computer Science, József Attila University, Szeged, Árpád tér 2, H-6720, Hungary*
[b] *Department of Applied Informatics, József Attila University, Szeged, Árpád tér 2, H-6720, Hungary*

Received June 1995; revised September 1996
Communicated by A. Salomaa

Abstract

We introduce the notion of the generalized semi-monadic rewrite system, which is a generalization of well-known rewrite systems: the ground rewrite system, the monadic rewrite system, and the semi-monadic rewrite system. We show that linear generalized semi-monadic rewrite systems effectively preserve recognizability. We show that a tree language L is recognizable if and only if there exists a rewrite system R such that $R \cup R^{-1}$ is a linear generalized semi-monadic rewrite system and that L is the union of finitely many $\leftrightarrow_R^*$-classes. We show several decidability and undecidability results on rewrite systems effectively preserving recognizability and on generalized semi-monadic rewrite systems. For example, we show that for a rewrite system R effectively preserving recognizability, it is decidable if R is locally confluent. Moreover, we show that preserving recognizability and effectively preserving recognizability are modular properties of linear collapse-free rewrite systems. Finally, as a consequence of our results on trees we get that restricted right–left overlapping string rewrite systems effectively preserve recognizability.

1. Introduction

Tree automata and recognizable tree languages proved to be an efficient tool in the theory of rewrite systems, see [15] for an overview. Let Σ be a ranked alphabet, let R be a rewrite system over Σ, and let L be a tree language over Σ. Then $R_\Sigma^*(L) = \{p \mid q \rightarrow_R^* p \text{ for some } q \in L\}$ is the set of descendants of trees in L. When Σ is apparent from the context, we simply write $R^*(L)$ rather than $R_\Sigma^*(L)$. A rewrite system R over Σ preserves Σ-recognizability, if for each recognizable tree language L over Σ, $R_\Sigma^*(L)$ is recognizable. The signature $sign(R)$ of a rewrite system R is the ranked alphabet consisting of all symbols appearing in the rules of R. In [14] Gilleron showed that for a rewrite system R it is not decidable if R preserves $sign(R)$-recognizability.

* Correspondence author. E-mail: vagvolgy@inf.u-szeged.hu.

[1] This research was supported by the grants OTKA F012852 of the Hungarian Academy of Sciences and MKM 223/95 of the Hungarian Cultural and Educational Ministry.

PII S0304-3975(96)00333-7

A rewrite system R over Δ preserves recognizability, if for each ranked alphabet Σ with $\Delta \subseteq \Sigma$, R preserves Σ-recognizability. It is not known yet whether or not it is decidable for a rewrite system R whether R preserves recognizability. We show that there is a ranked alphabet Σ and there is a linear rewrite system R over Σ such that R preserves Σ-recognizability but does not preserve recognizability. Let R be a rewrite system over $sign(R)$, and let $\Sigma = \{f, \sharp\} \cup sign(R)$, where $f \in \Sigma_2 - sign(R)$ and $\sharp \in \Sigma_0 - sign(R)$. We show that R preserves Σ-recognizability if and only if R preserves recognizability.

Let R be a rewrite system over a ranked alphabet Σ. We say that R effectively preserves Σ-recognizability if for a given tree automaton $\mathscr{B}$ over Σ, we can effectively construct a tree automaton $\mathscr{C}$ over Σ such that $L(\mathscr{C}) = R_{\Sigma}^{*}(L(\mathscr{B}))$. Let R be a rewrite system over a ranked alphabet Δ. We say that R effectively preserves recognizability if for a given ranked alphabet Σ with $\Delta \subseteq \Sigma$ and a given tree automaton $\mathscr{B}$ over Σ, we can effectively construct a tree automaton $\mathscr{C}$ over Σ such that $L(\mathscr{C}) = R_{\Sigma}^{*}(L(\mathscr{B}))$. Let R be a rewrite system over $sign(R)$, and let $\Sigma = \{f, \sharp\} \cup sign(R)$, where $f \in \Sigma_2 - sign(R)$ and $\sharp \in \Sigma_0 - sign(R)$. We show that R effectively preserves Σ-recognizability if and only if R effectively preserves recognizability.

In spite of Gilleron's undecidability results, we know several rewrite systems which preserve recognizability. Brainerd [2] showed that ground rewrite systems over any ranked alphabet Σ effectively preserve Σ-recognizability. Gallier and Book [11] introduced the notion of a monadic rewrite system, and Salomaa [20] showed that linear monadic rewrite systems over any ranked alphabet Σ effectively preserve Σ-recognizability. A rewrite system is monadic if each left-hand side is of depth at least 1 and each right-hand side is of depth at most 1. Coquidé et al. [4] defined the concept of a semi-monadic rewrite system generalizing the notion of a monadic rewrite system and the notion of a ground rewrite system. A rewrite system R over Σ is semi-monadic if, for every rule $l \to r$ in R, $depth(l) \geqslant 1$ and either $depth(r) = 0$ or $r = f(y_1, \ldots, y_k)$, where $f \in \Sigma_k$, $k \geqslant 1$, and for each $i \in \{1, \ldots, k\}$, either y_i is a variable (i.e., $y_i \in X$) or y_i is a ground term (i.e., $y_i \in T_{\Sigma}$). It is immediate that each monadic rewrite system is semi-monadic as well. Coquidé et al. [4] showed that linear semi-monadic rewrite systems over any ranked alphabet Σ effectively preserve Σ-recognizability. We generalize even further the concept of a semi-monadic rewrite system introducing the concept of a generalized semi-monadic rewrite system (gsm rewrite system for short). A rewrite system R is gsm if there is no rule $l \to r$ in R with $l \in X$ and the following holds. For any rules $l_1 \to r_1$ and $l_2 \to r_2$ in R, for any occurrences $\alpha \in O(r_1)$ and $\beta \in O(l_2)$, and for any supertree $l_3 \in T_{\Sigma}(X)$ of l_2/β with $var(l_3) \cap var(l_1) = \emptyset$, if

(i) $\alpha = \lambda$ or $\beta = \lambda$,

(ii) r_1/α and l_3 are unifiable, and

(iii) σ is a most general unifier of r_1/α and l_3,

then

(a) $l_2/\beta \in X$ or

(b) for each $\gamma \in O(l_3)$, if $l_2/\beta\gamma \in X$, then $\sigma(l_3/\gamma) \in X \cup T_{\Sigma}$.

We show that a linear gsm (lgsm) rewrite system R over Δ effectively preserves recognizability in the following way. Let L be a recognizable tree language over Σ with $\Delta \subseteq \Sigma$, and let $\mathscr{B} = (\Sigma, B, R_{\mathscr{B}}, B')$ be a tree automaton recognizing L. Similarly to the constructions of Salomaa [20] and Coquidé et al. [4], we construct a sequence of bottom-up tree automata $\mathscr{C}_i = (\Sigma, C, R_i, B')$, $i \geqslant 0$, having the same ranked alphabet, state set, and final state set. The rule set R_0 contains $R_{\mathscr{B}}$. Moreover, R_0 contains rules which enable R_0 to recognize the right-hand sides of rules in R. For each $i \geqslant 0$, R_{i+1} contains R_i, and for each rule $l \to r$ in R, $\mathscr{C}_{i+1}$ simulates, on the right-hand side r, the computation of $\mathscr{C}_i$ on the left-hand side l. There is a least integer $M \geqslant 0$ such that $R_M = R_{M+1}$. Hence $\mathscr{C}_M = \mathscr{C}_{M+1}$. We show that $L(\mathscr{C}_M) = R^*(L)$.

Brainerd [2], Kozen [17], and Fülöp and Vágvölgyi [10] showed that a tree language L is recognizable if and only if there exists a ground rewrite system R such that L is the union of finitely many $\leftrightarrow_R^*$-classes. We show that a tree language L is recognizable if and only if there exists a rewrite system R such that $R \cup R^{-1}$ is an lgsm rewrite system and that L is the union of finitely many $\leftrightarrow_R^*$-classes.

It is well known that the symbols of an alphabet Σ can be considered as unary function symbols, and hence words over Σ can be considered as unary trees over the ranked alphabet $\Sigma \cup \{\#\}$, where $\# \notin \Sigma$ is a symbol of rank 0. For example the word *apple* can be considered as the tree $a(p(p(l(e(\#)))))$, where $\# \notin \Sigma$ is a symbol of rank 0. Then recognizable string languages over Σ are the same as recognizable tree languages over the ranked alphabet $\Sigma \cup \{\#\}$. Let R be a string rewrite system over Σ. We can consider R as a rewrite system as follows. The left-hand sides and the right-hand sides of the rules in R can be considered as trees containing the variable x_1 instead of #.

Hence our concepts and results carry over to strings as well. Let R be a string rewrite system. We say that R is restricted right-left overlapping if there is no rule $\lambda \to r$ in R, and the following holds. For any rules $l_1 \to r_1$ and $l_2 \to r_2$ in R, for any nonempty suffix $u \in \Sigma^+$ of r_1 and any nonempty suffix $v \in \Sigma^+$ of l_2, if $u = r_1$ or $v = l_2$, then v cannot be a proper prefix of u. It should be clear what we mean when we say that a string rewrite system R effectively preserves recognizability. The following statement is an interesting consequence of our results on rewrite systems, and a generalization of the well-known result that monadic rewrite systems effectively preserve recognizability. Restricted right–left overlapping string rewrite systems effectively preserve recognizability. Moreover, a string language L is recognizable if and only if there exists a rewrite system R such that $R \cup R^{-1}$ is a restricted right-left overlapping string rewrite system and that L is the union of finitely many $\leftrightarrow_R^*$-classes.

We show the following decidability results.

(1) Let R_1, R_2 be rewrite systems. Let R_1 effectively preserve recognizability. Then it is decidable if $\to_{R_2}^* \subseteq \to_{R_1}^*$.

(2) Let R_1 and R_2 be rewrite systems effectively preserving recognizability. Then it is decidable which one of the following four mutually excluding conditions holds.

(i) $\to_{R_1}^* \subset \to_{R_2}^*$,

(ii) $\to_{R_2}^* \subset \to_{R_1}^*$,

(iii) $\to_{R_1}^* = \to_{R_2}^*$,
(iv) $\to_{R_1}^* \bowtie \to_{R_2}^*$,
where "$\bowtie$" stands for the incomparability relationship.

(3) For an lgsm rewrite system R, it is decidable whether R is left-to-right minimal. (A rewrite system R is left-to-right minimal if for each rule $l \to r$ in R, $\to_{R-\{l\to r\}}^* \subset \to_R^*$.)

(4) Let R_1 and R_2 be rewrite systems such that $R_1 \cup R_1^{-1}$ and $R_2 \cup R_2^{-1}$ are rewrite systems and effectively preserve recognizability. Then it is decidable which one of the following four mutually excluding conditions holds.
(i) $\leftrightarrow_{R_1}^* \subset \leftrightarrow_{R_2}^*$,
(ii) $\leftrightarrow_{R_2}^* \subset \leftrightarrow_{R_1}^*$,
(iii) $\leftrightarrow_{R_1}^* = \leftrightarrow_{R_2}^*$,
(iv) $\leftrightarrow_{R_1}^* \bowtie \leftrightarrow_{R_2}^*$.

(5) Let R be a rewrite system such that $R \cup R^{-1}$ is an lgsm rewrite system. Then it is decidable whether R is two-way minimal. (A rewrite system R is two-way minimal if for each rule $l \to r$ in R, $\leftrightarrow_{R-\{l\to r\}}^* \subset \leftrightarrow_R^*$.)

(6) Let R_1, R_2 be rewrite systems over a ranked alphabet Σ. Let R_1 effectively preserve recognizability. Let $g \in \Sigma - \Sigma_0$ be such that g does not occur on the left-hand side of any rule in R_1, and let $\sharp \in \Sigma_0$ be irreducible for R_1. Then it is decidable if $\to_{R_2}^* \cap (T_\Sigma \times T_\Sigma) \subseteq \to_{R_1}^* \cap (T_\Sigma \times T_\Sigma)$.

(7) Let R_1 and R_2 be rewrite systems over Σ effectively preserving recognizability. Moreover, let $g_1, g_2 \in \Sigma - \Sigma_0$ be such that for each $i \in \{1,2\}$, g_i does not occur on the left-hand side of any rule in R_i. Let $\sharp_1, \sharp_2 \in \Sigma_0$ be such that for each $i \in \{1,2\}$, $\sharp_i$ is irreducible for R_i. Then it is decidable which one of the following four mutually excluding conditions holds.
(i) $\to_{R_1}^* \cap (T_\Sigma \times T_\Sigma) \subset \to_{R_2}^* \cap (T_\Sigma \times T_\Sigma)$,
(ii) $\to_{R_2}^* \cap (T_\Sigma \times T_\Sigma) \subset \to_{R_1}^* \cap (T_\Sigma \times T_\Sigma)$,
(iii) $\to_{R_1}^* \cap (T_\Sigma \times T_\Sigma) = \to_{R_2}^* \cap (T_\Sigma \times T_\Sigma)$,
(iv) $\to_{R_1}^* \cap (T_\Sigma \times T_\Sigma) \bowtie \to_{R_2}^* \cap (T_\Sigma \times T_\Sigma)$.

(8) Let R be an lgsm rewrite system over Σ. Moreover, let $g \in \Sigma - \Sigma_0$ be such that g does not occur on the left-hand side of any rule in R, and let $\sharp \in \Sigma_0$ be irreducible for R. Then it is decidable whether R is left-to-right ground minimal. (A rewrite system R over Σ is left-to-right ground minimal if for each rule $l \to r$ in R, $\to_{R-\{l\to r\}}^* \cap (T_\Sigma \times T_\Sigma) \subset \to_R^* \cap (T_\Sigma \times T_\Sigma)$.)

(9) Let R_1 and R_2 be rewrite systems over Σ such that $R_1 \cup R_1^{-1}$ and $R_2 \cup R_2^{-1}$ are rewrite systems and effectively preserve recognizability. Moreover, let $g_1, g_2 \in \Sigma - \Sigma_0$ be such that for each $i \in \{1,2\}$, g_i does not occur in R_i. Let $\sharp_1, \sharp_2 \in \Sigma_0$ be such that for each $i \in \{1,2\}$, $\sharp_i$ is irreducible for $R_i \cup R_i^{-1}$. Then it is decidable which one of the following four mutually excluding conditions holds.
(i) $\leftrightarrow_{R_1}^* \cap (T_\Sigma \times T_\Sigma) \subset \leftrightarrow_{R_2}^* \cap (T_\Sigma \times T_\Sigma)$,
(ii) $\leftrightarrow_{R_2}^* \cap (T_\Sigma \times T_\Sigma) \subset \leftrightarrow_{R_1}^* \cap (T_\Sigma \times T_\Sigma)$,

(iii) $\leftrightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma) = \leftrightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$,

(iv) $\leftrightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma) \bowtie \leftrightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$,

where "$\bowtie$" stands for the incomparability relationship.

(10) Let R be a rewrite system over Σ such that $R \cup R^{-1}$ is an lgsm rewrite system. Moreover, let $g \in \Sigma - \Sigma_0$ be such that g does not occur in any rule of R, and let $\sharp \in \Sigma_0$ be irreducible for $R \cup R^{-1}$. Then it is decidable whether R is two-way ground minimal. (A rewrite system R over Σ is two-way ground minimal if for each rule $l \to r$ in R, $\leftrightarrow^*_{R-\{l \to r\}} \cap (T_\Sigma \times T_\Sigma) \subset \leftrightarrow^*_R \cap (T_\Sigma \times T_\Sigma)$.)

(11) Let R be a rewrite system over Σ effectively preserving recognizability, and let $p, q \in T_\Sigma(X)$. Then it is decidable if there exists a tree $r \in T_\Sigma(X)$ such that $p \to^*_R r$ and $q \to^*_R r$.

(12) Let R be a rewrite system over Σ effectively preserving recognizability. Then it is decidable if R is locally confluent.

By direct inspection we obtain that for any deterministic top-down tree transducer $\mathscr{A} = (\Sigma, \Delta, A, a_0, R)$ with $\Sigma \cap \Delta = \emptyset$, R is a convergent left-linear gsm rewrite system over the ranked alphabet $A \cup \Sigma \cup \Delta$. Hence Fülöp's [8] undecidability results on deterministic top-down tree transducers simply imply the following. Each of the following questions is undecidable for any convergent left-linear gsm rewrite systems R_1 and R_2 over a ranked alphabet Ω, for any recognizable tree language $L \subseteq T_\Omega$ given by a tree automaton over Ω recognizing L, where Γ is the smallest ranked alphabet for which $R_1(L) \subseteq T_\Gamma$. (Here $R_1(L)$ is the set of ground R_1-normal forms of L, i.e. $R_1(L) = R_1^*(L) \cap IRR(R_1)$.)

(i) Is $R_1(L) \cap R_2(L)$ empty?

(ii) Is $R_1(L) \cap R_2(L)$ infinite?

(iii) Is $R_1(L) \cap R_2(L)$ recognizable?

(iv) Is $T_\Gamma - R_1(L)$ empty?

(v) Is $T_\Gamma - R_1(L)$ infinite?

(vi) Is $T_\Gamma - R_1(L)$ recognizable?

(vii) Is $R_1(L)$ recognizable?

(viii) Is $R_1(L) = R_2(L)$?

(ix) Is $R_1(L) \subseteq R_2(L)$?

Fülöp and Gyenizse [9] showed that it is undecidable for a tree function induced by a deterministic homomorphism if it is injective. Hence the following holds. Let R be a convergent left-linear gsm rewrite system over Σ. Let $L \subseteq T_\Sigma$ be a recognizable tree language. Then it is undecidable if the tree function $\to^*_R \cap (L \times R(L))$ is injective.

We say that a rewrite system R is collapse-free if there is no rule $l \to r$ in R such that $l \in X$ or $r \in X$. Finally, we show that preserving recognizability and effectively preserving recognizability are modular properties of linear collapse-free rewrite systems. That is, the following results hold. Let R and S be linear collapse-free rewrite systems over disjoint ranked alphabets. Then R and S preserve recognizability if and only if the disjoint union $R \oplus S$ of R and S also preserves recognizability. Moreover, R and S effectively preserve recognizability if and only if the disjoint union $R \oplus S$ of R and S also effectively preserves recognizability. These results imply that preserving recognizability and effectively preserving recognizability are modular properties of λ-free string rewrite systems.

This paper is divided into six sections. In Section 2, we recall the necessary notions and notations. In Section 3, we show that lgsm rewrite systems effectively preserve recognizability. In Section 4, we illustrate by an example the constructions presented in Section 3. In Section 5, we study rewrite systems preserving recognizability and gsm rewrite systems. Finally, in Section 6, we present our concluding remarks, and some open problems.

2. Preliminaries

We recall and invent some notations, basic definitions and terminology which will be used in the rest of the paper. Nevertheless the reader is assumed to be familiar with the basic concepts of rewrite systems and of tree language theory (see, e.g. [7, 12, 13]).

The cardinality of a set A is denoted by $|A|$. The domain and the range of a binary relation ρ is denoted by $dom(\rho)$ and by $ran(\rho)$, respectively. We denote by ρ^{-1} the inverse of ρ. The composition of relations ρ and τ is denoted by $\rho \circ \tau$.

The set of nonnegative integers is denoted by N, and N^* stands for the free monoid generated by N with empty word λ as identity element. Consider the words $\alpha, \beta, \gamma \in N^*$ such that $\alpha = \beta\gamma$. Then we say that α is an extension of β, β is a prefix of α and γ is a suffix of α. Moreover, if $\alpha \neq \beta$, then β is a proper prefix of α. For a word $\alpha \in N^*$, $length(\alpha)$ stands for the length of α.

A ranked alphabet is a finite set Σ in which every symbol has a unique rank in N. For $m \geqslant 0$, Σ_m denotes the set of all elements of Σ which have rank m. The elements of Σ_0 are called constants. We assume that all ranked alphabets Σ and Δ that we consider have the following property. If $\sigma \in \Sigma_i$, and $\sigma \in \Delta_j$, then $i = j$. In other words, σ has the same rank in Σ as in Δ.

For a set of variables Y and ranked alphabet Σ, the set $T_\Sigma(Y)$ of Σ-terms (or Σ-trees) over Y is the smallest set satisfying

(a) $\Sigma_0 \cup Y \subseteq T_\Sigma(Y)$, and

(b) $f(t_1, \ldots, t_m) \in T_\Sigma(Y)$ whenever $m \geqslant 1$, $f \in \Sigma_m$ and $t_1, \ldots, t_m \in T_\Sigma(Y)$.

If $Y = \emptyset$, then $T_\Sigma(Y)$ is written as T_Σ. A term $t \in T_\Sigma(Y)$ is a ground term if $t \in T_\Sigma$ also holds. A tree $t \in T_\Sigma(Y)$ is linear if any variable of Y occurs at most once in t. We specify a countable set $X = \{x_1, x_2, \ldots\}$ of variables which will be kept fixed in this paper. Moreover, we put $X_m = \{x_1, \ldots, x_m\}$, for $m \geqslant 0$. Hence $X_0 = \emptyset$.

We shall need a few functions on terms. For a term $t \in T_\Sigma(X)$, the depth $depth(t) \in N$, the set of variables $var(t)$ of t, and the set of occurrences $O(t) \subseteq N^*$ are defined by recursion:

(a) if $t \in \Sigma_0 \cup X$, then

$depth(t) = 0$,

$var(t) = \emptyset$ if $t \in \Sigma_0$ and $var(t) = t$ if $t \in X$, and

$O(t) = \{\lambda\}$;

(b) if $t=f(t_1,\dots,t_m)$ with $m\geqslant 1$ and $f\in\Sigma_m$, then

$depth(t)=1+max\{depth(t_i)\mid 1\leqslant i\leqslant m\}$,

$var(t)=var(t_1)\cup\dots\cup var(t_m)$, and

$O(t)=\{\lambda\}\cup\{i\alpha\mid 1\leqslant i\leqslant m \text{ and } \alpha\in O(t_i)\}$.

We note that $depth(t)=max\{length(\alpha)\mid\alpha\in O(t)\}$.

For each $t\in T_\Sigma(X)$ and $\alpha\in O(t)$, we introduce the subterm $t/\alpha\in T_\Sigma(X)$ of t at α as follows:

(a) for $t\in\Sigma_0\cup X$, $t/\lambda=t$;

(b) for $t=f(t_1,\dots,t_m)$ with $m\geqslant 1$ and $f\in\Sigma_m$, if $\alpha=\lambda$ then $t/\alpha=t$, otherwise, if $\alpha=i\beta$ with $1\leqslant i\leqslant m$, then $t/\alpha=t_i/\beta$. Moreover, we say that p is a subtree of t if $p=t/\alpha$ for some $\alpha\in O(t)$.

For $t\in T_\Sigma$, $\alpha\in O(t)$, and $r\in T_\Sigma$, we define $t[\alpha\leftarrow r]\in T_\Sigma$ as follows.

(i) If $\alpha=\lambda$, then $t[\alpha\leftarrow r]=r$.

(ii) If $\alpha=i\beta$, for some $i\in N$ and $\beta\in N^*$, then $t=f(t_1,\dots,t_m)$ with $f\in\Sigma_m$ and $1\leqslant i\leqslant m$. Then $t[\alpha\leftarrow r]=f(t_1,\dots,t_{i-1},t_i[\beta\leftarrow r],t_{i+1},\dots,t_m)$.

For any $m\geqslant 1$, we distinguish a subset $\bar{T}_\Sigma(X_m)$ of $T_\Sigma(X_m)$ as follows: a tree $t\in T_\Sigma(X_m)$ is in $\bar{T}_\Sigma(X_m)$ if and only if each variable in X_m appears exactly once in t and the order of the variables from left to right in t is $x_1,\dots,x_m$. For example, if $\Sigma=\Sigma_0\cup\Sigma_2$ with $\Sigma_0=\{a\}$ and $\Sigma_2=\{f\}$, then $f(x_1,f(a,x_1))\in T_\Sigma(X_1)$ but $f(x_1,f(a,x_1))\notin\bar{T}_\Sigma(X_1)$. On the other hand, $f(x_1,f(a,x_2))\in\bar{T}_\Sigma(X_2)$. Moreover, for any $m\geqslant 0$, we define the subset $\hat{T}_\Sigma(X_m)$ of $T_\Sigma(X_m)$ as follows: a tree $t\in T_\Sigma(X_m)$ is in $\hat{T}_\Sigma(X_m)$ if and only if t is linear.

Let Σ be a ranked alphabet. Let $f\in\Sigma_1$, $t\in T_\Sigma$ be arbitrary. The tree $f^k(t)\in T_\Sigma$, $k\geqslant 0$, is defined by recursion: $f^0(t)=t$, and $f^{k+1}(t)=f(f^k(t))$ for $k\geqslant 0$.

A substitution is a mapping $\sigma:X\to T_\Sigma(X)$ which is different from the identity only for a finite subset $Dom(\sigma)$ of X. For any substitution σ with $Dom(\sigma)\subseteq X_m$, $m\geqslant 0$, the term $\sigma(t)$ is produced from t by replacing each occurrence of x_i with $\sigma(x_i)$ for $1\leqslant i\leqslant m$. For any trees $t\in\bar{T}_\Sigma(X_k)$, $t_1,\dots,t_k\in T_\Sigma(X)$ and for the substitution σ with $Dom(\sigma)\subseteq X_k$ and $\sigma(x_i)=t_i$ for $i=1,\dots,k$, we denote the term $\sigma(t)$ by $t[t_1,\dots,t_k]$ as well. Moreover, for any k,m with $1\leqslant m\leqslant k$, for any tree $t\in T_\Sigma(\{x_m,\dots,x_k\})$ and for any substitution σ with $\sigma(x_m)=t_m,\dots,\sigma(x_k)=t_k$, we denote $\sigma(t)$ also by $t[x_m\leftarrow t_m,\dots,x_k\leftarrow t_k]$.

We say that the pair (l_1,r_1) is a variant of the pair (l_2,r_2) if there is a substitution $\sigma:X\to X$ such that

(i) $\sigma(l_2)=l_1$ and $\sigma(r_2)=r_1$, and that

(ii) for all $x_i,x_j\in var(l_2)\cup var(r_2)$, $\sigma(x_i)=\sigma(x_j)$ implies that $x_i=x_j$.

Let Σ be a ranked alphabet and let $s,t\in T_\Sigma(X)$. A unifier of s and t is a substitution θ such that $\theta(s)=\theta(t)$. A most general unifier of s and t is a unifier θ of s and t such that for each unifier σ of s and t, there is a substitution σ' satisfying that $\sigma'(\theta(s))=\sigma(s)$ and $\sigma'(\theta(t))=\sigma(t)$. It is decidable if s and t are unifiable. Moreover, if s and t are unifiable, then one can effectively construct a most general unifier of s and t, see Theorem 4.3 in [18]. Throughout the paper we shall consider the most general unifiers of two arbitrary unifiable linear terms $s,t\in T_\Sigma(X)$ with $var(s)\cap var(t)=\emptyset$. We construct a most general unifier of s and t as follows. Let the substitution $\sigma : X \to T_\Sigma(X)$ be defined in the following way. Let $x \in var(s)$ be arbitrary and let $\alpha \in O(s)$ be such that $s/\alpha \in X$. If $\alpha \in O(t)$, then let $\sigma(x) = t/\alpha$, otherwise let $\sigma(x) = x$. Moreover, let $x \in var(t)$ be arbitrary and let $\alpha \in O(t)$ be such that $t/\alpha \in X$. If $\alpha \in O(s)$ and $s/\alpha \notin X$, then let $\sigma(x) = s/\alpha$, otherwise let $\sigma(x) = x$. It should be clear that σ is a most general unifier of s and t. It is well known that a most general unifier of s and t is unique up to renaming of variables. Hence for any most general unifier σ_1 of s and t and for any variable $x \in var(s) \cup var(t)$, if $\sigma(x) \in T_\Sigma$ or $\sigma_1(x) \in T_\Sigma$ then $\sigma(x) = \sigma_1(x)$.

Let Σ be a ranked alphabet and let $u, v \in T_\Sigma(X)$. The tree u is a supertree of v if u is linear and there is a substitution σ such that $v = \sigma(u)$. We illustrate the concept of a supertree by an example. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, $\Sigma_0 = \{\sharp\}$, $\Sigma_1 = \{f\}$, $\Sigma_2 = \{g\}$. Trees $f(x_2)$, $f(g(x_2, x_1))$, $f(g(\sharp, x_2))$ are supertrees of $f(g(\sharp, \sharp))$. On the other hand, $f(f(x_1))$ is not a supertree of $f(g(\sharp, \sharp))$, since there is no substitution σ such that $\sigma(f(f(x_1))) = f(g(\sharp, \sharp))$. Moreover, $f(g(x_1, x_1))$ is not a supertree of $f(g(\sharp, \sharp))$ as $f(g(x_1, x_1))$ is not linear.

Let Σ be a ranked alphabet. Then a rewrite system R over Σ is a finite subset of $T_\Sigma(X) \times T_\Sigma(X)$ such that for each $(l, r) \in R$, each variable of r also occurs in l. Elements (l, r) of R are called rules and are denoted by $l \to r$.

Note that for a rewrite system R, the set $R \cup R^{-1}$ is also a rewrite system if and only if for each rule $l \to r$ in R, each variable of l also occurs in r.

Let R be a rewrite system. We say that R is collapse-free if there is no rule $l \to r$ in R such that $l \in X$ or $r \in X$.

Let R be a rewrite system over Σ. Then $sign(R) \subseteq \Sigma$ is the ranked alphabet consisting of all symbols appearing in the rules of R.

Let R be a rewrite system over Σ. Given any two terms s and t in $T_\Sigma(X)$ and an occurrence $\alpha \in O(s)$, we say that s rewrites to t at α and denote this by $s \to_R t$ if there is some pair $(l, r) \in R$ and a substitution σ such that $s/\alpha = \sigma(l)$ and $t = s[\alpha \leftarrow \sigma(r)]$. Here we also say that R rewrites s to t applying the rule $l \to r$ at α. Relation $\to_R^*$ is the reflexive and transitive closure of $\to_R$, and $\leftrightarrow_R^*$ is the reflexive, symmetric, and transitive closure of $\to_R$. Finally, $\to_R^+$ is the transitive closure of $\to_R$. It should be clear that $\leftrightarrow_R^*$ is an equivalence relation. We denote by $[t]_R$ the $\leftrightarrow_R^*$-class of a tree $t \in T_\Sigma(X)$. Note that if $R \cup R^{-1}$ is a rewrite system and $t \in T_\Sigma$, then $[t]_R \subseteq T_\Sigma$.

A left-linear (linear, resp.) rewrite system is one in which no variable occurs more than once on any left-hand side (right-hand side and left-hand side, resp.). A ground rewrite system is one of which all rules are ground (i.e. elements of $T_\Sigma \times T_\Sigma$).

A rewrite system is monadic if each left-hand side is of depth at least 1 and each right-hand side is of depth at most 1. Coquidé et al. [4] defined the concept of a semi-monadic rewrite system generalizing the notion of a monadic rewrite system and the notion of a ground rewrite system. A rewrite system R over Σ is semi-monadic if, for every rule $l \to r$ in R, $depth(l) \geqslant 1$ and either $depth(r) = 0$ or $r = f(y_1, \ldots, y_k)$, where $f \in \Sigma_k$, $k \geqslant 1$, and for each $i \in \{1, \ldots, k\}$, either y_i is a variable (i.e., $y_i \in X$) or y_i is a ground term (i.e., $y_i \in T_\Sigma$). It is immediate that each monadic rewrite system is semi-monadic as well.

Let R be a rewrite system over Σ.

(a) R is left-to-right minimal if for each rule $l \to r$ in R, $\to^*_{R-\{l\to r\}} \subset \to^*_R$.

(b) R is left-to-right ground minimal if for each rule $l \to r$ in R, $\to^*_{R-\{l\to r\}} \cap (T_\Sigma \times T_\Sigma) \subset \to^*_R \cap (T_\Sigma \times T_\Sigma)$.

(c) R is two-way minimal if for each rule $l \to r$ in R, $\leftrightarrow^*_{R-\{l\to r\}} \subset \leftrightarrow^*_R$.

(d) R is two-way ground minimal if for each rule $l \to r$ in R, $\leftrightarrow^*_{R-\{l\to r\}} \cap (T_\Sigma \times T_\Sigma) \subset \leftrightarrow^*_R \cap (T_\Sigma \times T_\Sigma)$.

Let $\to$ be a binary relation on a set A. We say that $\to$ is

(i) confluent if, for every $u, v_1, v_2 \in A$, it holds that if $u \to^* v_1$ and $u \to^* v_2$, then there exists a $v_3 \in A$ such that $v_1 \to^* v_3$ and $v_2 \to^* v_3$;

(ii) noetherian if there is no infinite sequence $v_1 \to v_2 \to v_3 \to \cdots$;

(iii) convergent if $\to$ is confluent and noetherian.

A rewrite system R over Σ is confluent (noetherian, convergent) if the induced rewrite relation $\to_R$ is confluent (noetherian, convergent).

Let R be a rewrite system over Σ. A term $t \in T_\Sigma(X)$ is called irreducible for R if there does not exist $t' \in T_\Sigma(X)$ with $t \to_R t'$. The set of all ground terms that are irreducible for R is denoted by $IRR(R)$.

Let R be a convergent rewrite system over Σ, and let $p \in T_\Sigma(X)$. It is well known that there exists exactly one term $t \in T_\Sigma(X)$ irreducible for R such that $p \to^*_R t$. We call t the R-normal form of p. Let $p \in T_\Sigma$ be arbitrary, and let t be the R-normal form of p. It is obvious that $t \in IRR(R)$. Let $L \subseteq T_\Sigma$. The set of R-normal forms of the trees in the tree language L is denoted by $R(L)$. It should be clear that $R(L) = R^*(L) \cap IRR(R)$.

We adopt the concept of a critical pair from [16]. Let R be a rewrite system and assume that the rules $l_1 \to r_1$, $l_2 \to r_2$ are in R. Let us take a variant $l_2' \to r_2'$ of $l_2 \to r_2$ such that $var(l_1) \cap var(l_2') = \emptyset$. Let us assume that there is a tree $t = l_1/\alpha$, where $\alpha \in O(l_1)$, such that $t \notin X$, t and l_2' are unifiable. Let σ be a most general unifier of t and l_2'. Let $v_1 = \sigma(r_1)$ and define v_2 from $\sigma(l_1)$ by substituting $\sigma(r_2')$ for the subterm $\sigma(t) = \sigma(l_2')$ at the occurrence α. Then we call (v_1, v_2) a critical pair of R. Huet [16] showed the following result.

Proposition 2.1. *Let R be a rewrite system over Σ. Then R is locally confluent if and only if for every critical pair (v_1, v_2) of R there exists a tree $v \in T_\Sigma(X)$ such that $v_1 \to^*_R v$ and $v_2 \to^*_R v$.*

Let R and S be rewrite systems over the disjoint ranked alphabets Σ and Δ, respectively. Then the disjoint union $R \oplus S$ of R and S is the rewrite system $R \cup S$ over the ranked alphabet $\Sigma \cup \Delta$. Let $\mathbf{C}$ be a class of rewrite systems, let $\mathbf{C}$ be closed under disjoint union. A property $\mathscr{P}$ is modular for $\mathbf{C}$ if for any $R, S \in \mathbf{C}$ over disjoint ranked alphabets, $R \oplus S$ has the property $\mathscr{P}$ if and only if both R and S have the property $\mathscr{P}$. For a short survey on the disjoint union of rewrite systems, see the introduction of [3]. Moreover, see [3] also for recent results in this area.

Let Σ be a ranked alphabet, a bottom-up tree automaton over Σ is a quadruple $\mathscr{A}=(\Sigma,A,R,A_{\mathrm{f}})$, where A is a finite set of states of rank 0, $\Sigma\cap A=\emptyset$, $A_{\mathrm{f}}(\subseteq A)$ is the set of final states, R is a finite set of rules of the following two types:

(i) $\delta(a_1,\ldots,a_n)\rightarrow a$ with $n\geqslant 0$, $\delta\in\Sigma_n$, $a_1,\ldots,a_n,a\in A$.

(ii) $a\rightarrow a'$ with $a,a'\in A$ (λ-rules).

We consider R as a ground rewrite system over $\Sigma\cup A$. The tree language recognized by $\mathscr{A}$ is $L(\mathscr{A})=\{t\in T_\Sigma \mid (\exists a\in A_{\mathrm{f}})\ t\rightarrow_R^* a\}$. A tree language L is recognizable if there exists a bottom-up tree automaton $\mathscr{A}$ such that $L(\mathscr{A})=L$ (see [12]).

The bottom-up tree automaton $\mathscr{A}=(\Sigma,A,R,A_{\mathrm{f}})$ is deterministic if R has no λ-rules and R has no two rules with the same left-hand side. We say that the bottom-up tree automaton $\mathscr{A}$ is connected if for every $a\in A$ there exists $t\in T_\Sigma$ such that $t\rightarrow_R^* a$. Every recognizable tree language can be recognized by a deterministic connected bottom-up tree automaton (see [12]).

The following important result was shown by Brainerd [2], Kozen [17], and Fülöp and Vágvölgyi [10].

Proposition 2.2. *A tree language L is recognizable if and only if there exists a ground rewrite system R such that L is the union of finitely many $\leftrightarrow_R^*$-classes.*

Let Σ be a ranked alphabet, let R be a rewrite system over Σ, and let L be a tree language over Σ. Then $R_\Sigma^*(L)=\{p \mid q\rightarrow_R^* p$ for some $q\in L\}$ is the set of descendants of trees in L. When Σ is apparent from the context, we simply write $R^*(L)$ rather than $R_\Sigma^*(L)$. A rewrite system R over Σ preserves Σ-recognizability, if for each recognizable tree language L over Σ, $R_\Sigma^*(L)$ is recognizable. A rewrite system R over Δ preserves recognizability, if for each ranked alphabet Σ with $\Delta\subseteq\Sigma$, R preserves Σ-recognizability.

Let R be a rewrite system over a ranked alphabet Σ. We say that R effectively preserves Σ-recognizability if for a given tree automaton $\mathscr{B}$ over Σ, we can effectively construct a tree automaton $\mathscr{C}$ over Σ such that $L(\mathscr{C})=R_\Sigma^*(L(\mathscr{B}))$. Let R be a rewrite system over a ranked alphabet Δ. We say that R effectively preserves recognizability if for a given ranked alphabet Σ with $\Delta\subseteq\Sigma$ and a given tree automaton $\mathscr{B}$ over Σ, we can effectively construct a tree automaton $\mathscr{C}$ over Σ such that $L(\mathscr{C})=R_\Sigma^*(L(\mathscr{B}))$. The proofs of the following results are straightforward.

Lemma 2.3. *Let R be a rewrite system over Δ. Then the following statements are equivalent.*

(i) *R preserves recognizability.*

(ii) *For each ranked alphabet Σ with sign$(R)\subseteq\Sigma$, R preserves Σ-recognizability.*

Lemma 2.4. *Let R be a rewrite system over Δ. Then the following statements are equivalent.*

(i) *R effectively preserves recognizability.*

(ii) *For each ranked alphabet Σ with sign$(R)\subseteq\Sigma$, R effectively preserves Σ-recognizability.*

A top-down tree transducer is a 5-tuple $\mathscr{A} = (\Sigma, \Delta, A, a_0, R)$, where
(a) Σ and Δ are the input and output ranked alphabets,
(b) A, the set of states, is a ranked alphabet containing only 1-ary elements,
(c) $a_0 \in A$ is the initial state, and
(d) R is a rewrite system over the ranked alphabet $A \cup \Sigma \cup \Delta$, R consists of rules of the form

$$a(\sigma(x_1, \ldots, x_m)) \to u,$$

where $m \geqslant 0$, $\sigma \in \Sigma_m$, $a \in A$, $u \in T_{\Sigma \cup A}(X_m)$, $u = p[a_1(x_{i_1}), \ldots, a_n(x_{i_n})]$, $n \geqslant 0$, $p \in \bar{T}_\Sigma(X_n)$, and for each $1 \leqslant j \leqslant n$, $a_j \in A$, $1 \leqslant i_j \leqslant m$.

The tree transformation induced by $\mathscr{A}$ is the relation

$$\tau_{\mathscr{A}} = \left\{ (s,t) \in T_\Sigma \times T_\Delta \;\middle|\; q_0(s) \xrightarrow[R]{*} t \right\}.$$

We say that $\mathscr{A}$ is deterministic if different rules in R have different left-hand sides. In this case $\tau_{\mathscr{A}}$ is a partial function from T_Σ to T_Δ.

Proposition 2.5 (Fülöp [8]). *Let $\mathscr{A} = (\Sigma, \Delta, A, a_0, R)$ be a deterministic top-down tree transducer. Then R is a convergent rewrite system over the ranked alphabet $A \cup \Sigma \cup \Delta$. Moreover, $ran(\tau_{\mathscr{A}}) = R(L)$, where L is the recognizable tree language $\{a_0(s) \mid s \in dom(\tau_{\mathscr{A}})\}$.*

Fülöp [8] have shown the following undecidability results.

Proposition 2.6. *Each of the following problems is undecidable for arbitrary deterministic top-down tree transducers $\mathscr{A}_1 = (\Sigma, \Delta, A_1, a_1, R_1)$ and $\mathscr{A}_2 = (\Sigma, \Delta, A_2, a_2, R_2)$, where we denote $L_1 = ran(\tau_{\mathscr{A}_1})$ and $L_2 = ran(\tau_{\mathscr{A}_2})$.*
(i) *Is $L_1 \cap L_2$ empty?*
(ii) *Is $L_1 \cap L_2$ infinite?*
(iii) *Is $L_1 \cap L_2$ recognizable?*
(iv) *Is $T_\Delta - L_1$ empty?*
(v) *Is $T_\Delta - L_1$ infinite?*
(vi) *Is $T_\Delta - L_1$ recognizable?*
(vii) *Is L_1 recognizable?*
(viii) *Is $L_1 = L_2$?*
(ix) *Is $L_1 \subseteq L_2$?*

Applying the results of Proposition 2.6, Fülöp [8] have shown the following undecidability results.

Proposition 2.7. *Each of the following questions is undecidable for any convergent left-linear rewrite systems R_1 and R_2 over a ranked alphabet Ω, for any recognizable tree language $L \subseteq T_\Omega$ given by a tree automaton over Ω recognizing L, where Γ is the smallest ranked alphabet for which $R_1(L) \subseteq T_\Gamma$.*

(i) *Is* $R_1(L) \cap R_2(L)$ *empty*?
(ii) *Is* $R_1(L) \cap R_2(L)$ *infinite*?
(iii) *Is* $R_1(L) \cap R_2(L)$ *recognizable*?
(iv) *Is* $T_\Gamma - R_1(L)$ *empty*?
(v) *Is* $T_\Gamma - R_1(L)$ *infinite*?
(vi) *Is* $T_\Gamma - R_1(L)$ *recognizable*?
(vii) *Is* $R_1(L)$ *recognizable*?
(viii) *Is* $R_1(L) = R_2(L)$?
(ix) *Is* $R_1(L) \subseteq R_2(L)$?

3. Generalized semi-monadic rewrite systems

In this section we introduce the notion of a gsm rewrite system and show that linear gsm rewrite systems effectively preserve recognizability.

Definition 3.1. Let R be a rewrite system over Σ. We say that R is a generalized semi-monadic rewrite system (gsm rewrite system for short) if there is no rule $l \to r$ in R with $l \in X$ and the following holds. For any rules $l_1 \to r_1$ and $l_2 \to r_2$ in R, for any occurrences $\alpha \in O(r_1)$ and $\beta \in O(l_2)$, and for any supertree $l_3 \in T_\Sigma(X)$ of l_2/β with $var(l_3) \cap var(l_1) = \emptyset$, if
(i) $\alpha = \lambda$ or $\beta = \lambda$,
(ii) r_1/α and l_3 are unifiable, and
(iii) σ is a most general unifier of r_1/α and l_3,
then
(a) $l_2/\beta \in X$ or
(b) for each $\gamma \in O(l_3)$, if $l_2/\beta\gamma \in X$, then $\sigma(l_3/\gamma) \in X \cup T_\Sigma$.

Notice that Condition (a) implies that $l_3 \in X$. We abbreviate the expression linear gsm to lgsm.

Example 3.2. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, $\Sigma_0 = \{\sharp\}$, $\Sigma_1 = \{f\}$, and $\Sigma_2 = \{g\}$. Let the rewrite system R over Σ consist of the rule

$$g(x_1, x_2) \to f(g(x_1, \sharp)) .$$

We obtain by direct inspection that R is lgsm.

Definition 3.3. A rewrite system R over Σ is restricted right-left overlapping if there is no rule $l \to r$ in R with $l \in X$ and the following holds. For any rules $l_1 \to r_1$ and $l_2 \to r_2$ in R, for any occurrences $\alpha \in O(r_1)$ and $\beta \in O(l_2)$, and for any supertree $l_3 \in T_\Sigma(X)$ of l_2/β with $var(l_3) \cap var(l_1) = \emptyset$, if (i)–(iii) in Definition 3.1 hold, then (a′), (b′), or (c′) hold.
(a′) $\alpha = \lambda$, $l_2/\beta \in X$.
(b′) $\alpha = \lambda$ and for each $\gamma \in O(l_3)$, if $l_2/\beta\gamma \in X$, then $\sigma(l_3/\gamma) \in X \cup T_\Sigma$.
(c′) $\beta = \lambda$ and for each $\gamma \in O(l_3)$, if $l_2/\gamma \in X$, then $\sigma(l_3/\gamma) \in X \cup T_\Sigma$.

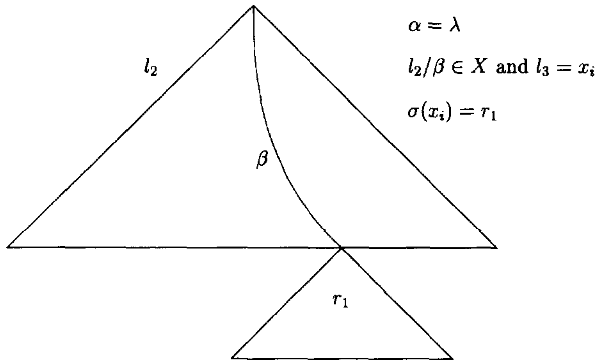

Fig. 1. The unification of r_1/α and the supertree l_3 of l_2/β by the most general unifier σ, when Condition (a′) holds

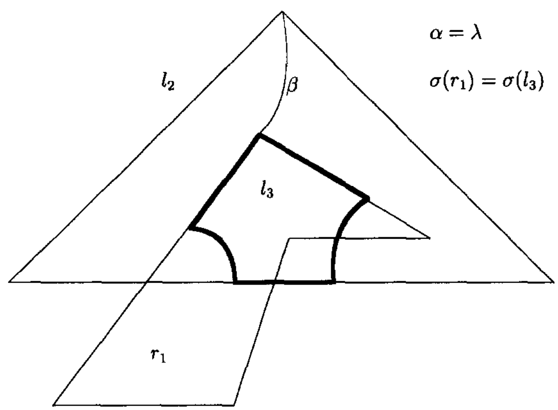

Fig. 2. The unification of r_1/α and the supertree l_3 of l_2/β by the most general unifier σ, when Condition (b′) holds

Note that Condition (a′) implies that $l_3 \in X$. We visualize the unification of r_1/α and the supertree l_3 of l_2/β by the most general unifier σ, when Condition (a′) (Condition (b′), Condition (c′), respectively) holds on Fig. 1 (Fig. 2, Fig. 3, respectively).

The proofs of the following two results are straightforward.

Observation 3.4. *A rewrite system R is gsm if and only if R is restricted right–left overlapping.*

Observation 3.5. *Each semi-monadic rewrite system is gsm as well.*

We obtain the following result by direct inspection.

Lemma 3.6. *For each top-down tree transducer $\mathscr{A} = (\Sigma, \Delta, A, a_0, R)$, there exists a top-down tree transducer $\mathscr{B} = (\Sigma', \Delta, A, a_0, R')$ such that $\Sigma' \cap \Delta = \emptyset$ and that $ran(\tau_{\mathscr{A}}) = ran(\tau_{\mathscr{B}})$. Moreover, let $\mathscr{A} = (\Sigma, \Delta, A, a_0, R)$ be a deterministic top-down tree transducer with $\Sigma \cap \Delta = \emptyset$. Then R is a left-linear gsm rewrite system.*

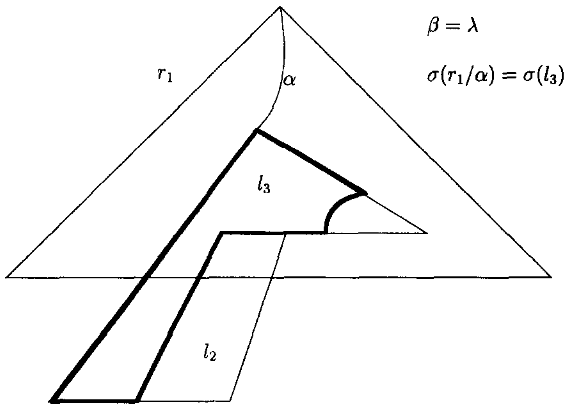

Fig. 3. The unification of r_1/α and the supertree l_3 of l_2 by the most general unifier σ, when Condition (c′) holds

We now show that lgsm rewrite systems effectively preserve recognizability. To this end, from now on in this section, let R be an lgsm rewrite system over some ranked alphabet Δ, let Σ be an arbitrary ranked alphabet such that $\Delta \subseteq \Sigma$. Moreover, let $L = L(\mathscr{B})$ be a recognizable tree language over Σ, where $\mathscr{B} = (\Sigma, B, R_B, B')$ is a deterministic connected bottom-up tree automaton over Σ. Via a series of Theorems and lemmas we show that $R^*_\Sigma(L)$ is recognizable. In fact we construct a tree automaton $\mathscr{C}$ over Σ such that $L(\mathscr{C}) = R^*_\Sigma(L)$. Our construction is illustrated by an example in Section 4. As we are interested in the tree language $R^*_\Sigma(L)$ rather than in $R^*_\Delta(L)$, by $R^*(L)$ we always mean $R^*_\Sigma(L)$.

Let E be the set of all ground terms u over Σ such that there are rules $l_1 \to r_1$ and $l_2 \to r_2$ in R, and there are occurrences $\alpha \in O(r_1)$ and $\beta \in O(l_2)$, and there is a supertree $l_3 \in T_\Sigma(X) - X$ of l_2/β with $var(l_3) \cap var(l_1) = \emptyset$ such that

(i) $\alpha = \lambda$ or $\beta = \lambda$,

(ii) r_1/α and l_3 are unifiable, and

(iii) σ is a most general unifier of r_1/α and l_3, and

(iv) there is an occurrence $\gamma \in O(l_3)$ such that $l_2/\beta\gamma \in X$ and $\sigma(l_3/\gamma) \in T_\Sigma$, and that u is a subterm of $\sigma(l_3/\gamma)$.

It should be clear that E is finite and is effectively constructable.

Recall that $\mathscr{B} = (\Sigma, B, R_B, B')$ is a deterministic connected bottom-up tree automaton such that $L(\mathscr{B}) = L$. We lose no generality by assuming that $B \cap N = \emptyset$. Moreover, without loss of generality we may assume that for each rule $l \to r$ in R, $l \in \bar{T}_\Sigma(X_n)$ for some $n \geqslant 0$. Let

$$D = B \cup \{p[a_1, \ldots, a_n] \mid n \geqslant 0,\ p \in T_\Sigma(X_n),\ a_1, \ldots, a_n \in B \cup E, p \text{ is a subtree of the right-hand side } r \text{ of some rule } l \to r \text{ in } R\}.$$

It should be clear that $B \cup E \subseteq D$. Let

$$C = B \cup \{1, \ldots, |D - B|\}.$$

We consider C as a ranked alphabet, for each $c \in C$ the rank of c is 0. Let $\langle\,\rangle : D \to C$ be a bijection such that $\langle b \rangle = b$ for each $b \in B$.

For each $i \geqslant 1$, consider the bottom-up tree automaton $\mathscr{C}_i = (\Sigma, C, R_i, B')$, where R_i is defined by recursion on i (for an example see Section 4).

We define R_0 as follows.

(i) $R_{\mathscr{B}} \subseteq R_0$.

(ii) For all $n \geqslant 0$, $f \in \Sigma_n$, $t_1, \ldots, t_n \in D$, if $f(t_1, \ldots, t_n) \in D$, then we put the rule $f(\langle t_1 \rangle, \ldots, \langle t_n \rangle) \to \langle f(t_1, \ldots, t_n) \rangle$ in R_0.

We shall refer to a rule appearing in (ii) as a (ii)-type rule of R_0.

Let us assume that $i \geqslant 1$ and we have defined the set R_{i-1}. Then we define R_i as follows.

(a) $R_{i-1} \subseteq R_i$.

(b) For any rule $l \to r$ in R with $n \geqslant 0$, $l \in \bar{T}_\Sigma(X_n)$, for all $a_1, \ldots, a_n \in B \cup E$, if $l[\langle a_1 \rangle, \ldots, \langle a_n \rangle] \to^*_{R_{i-1}} c$ for some $c \in C$, then we put the rule $\langle r[a_1, \ldots, a_n] \rangle \to c$ in R_i.

As $\mathscr{B}$ is connected, all states in B are reachable in $\mathscr{C}_0$. By (ii) in the definition of R_0, all states in $\{1, \ldots, |D - B|\}$ are reachable in R_0. Hence $\mathscr{C}_0$ is connected. As $R_i \subseteq R_{i+1}$ for $i \geqslant 0$, $\mathscr{C}_i$ is connected for $i \geqslant 1$.

It should be clear that there is an integer $M \geqslant 0$ such that $R_M = R_{M+1}$. Let M be the least integer such that $R_M = R_{M+1}$. Let $\mathscr{C} = \mathscr{C}_M$. Let $S = R_M$, and from now on we write $\mathscr{C} = (\Sigma, C, S, B')$, rather than $\mathscr{C}_M = (\Sigma, C, R_M, B')$.

Our aim is to show that $R^*(L) = L(\mathscr{C})$. To this end, first we show five preparatory lemmas, then the inclusion $L(\mathscr{C}) \subseteq R^*(L)$, then again five preparatory lemmas, and finally the inclusion $R^*(L) \subseteq L(\mathscr{C})$.

Lemma 3.7. $L = L(\mathscr{C}_0)$.

Proof. By direct inspection of the set R_0 of rules. □

Lemma 3.8. *For any $p \in T_\Sigma$, if $p \to^*_{R_0} \langle r[a_1, \ldots, a_n] \rangle$ for some $r \in \bar{T}_\Sigma(X_n)$, $n \geqslant 0$, and $a_1, \ldots, a_n \in B \cup E$, then $p = r[p_1, \ldots, p_n]$, where $p_i \in T_\Sigma$ and $p_i \to^*_{R_0} \langle a_i \rangle$ for $1 \leqslant i \leqslant n$.*

Proof. By direct inspection of the rules of R_0. □

The following statement is a simple consequence of Lemma 3.8.

Lemma 3.9. *For any $p \in T_\Sigma$, if $p \to^*_{R_0} \langle r[a_1, \ldots, a_n] \rangle$ for some $r \in \hat{T}_\Sigma(X_n)$, $n \geqslant 0$, and $a_1, \ldots, a_n \in B \cup E$, then $p = r[p_1, \ldots, p_n]$, where for each $1 \leqslant i \leqslant n$, if the variable x_i appears in the tree r, then $p_i \in T_\Sigma$ and $p_i \to^*_{R_0} \langle a_i \rangle$.*

Lemma 3.10. *For any $i \geqslant 1$, $p \in T_\Sigma$, $q, t \in T_{\Sigma \cup C}$, $k \geqslant 1$, and $v_1, \ldots, v_k \in T_{\Sigma \cup C}$, if*

$$p = v_1 \underset{R_0}{\to} v_2 \underset{R_0}{\to} \cdots \underset{R_0}{\to} v_k = q \underset{R_i}{\to} t, \tag{1}$$

and $\mathscr{C}_i$ applies an $(R_i - R_{i-1})$-rule in the last step $q \to_{R_i} t$ of (1), then there exists an $s \in T_\Sigma$ such that

$$s \underset{R}{\to} p \quad \text{and} \quad s \overset{*}{\underset{R_{i-1}}{\to}} t. \tag{2}$$

Proof. Let α be the occurrence where $\mathscr{C}_i$ applies an $(R_i - R_{i-1})$-rule

$$\langle r[a_1,\dots,a_n]\rangle \to c$$

in the last step $q \to_{R_i} t$ of (1). Then

$$q = u[\langle r[a_1,\dots,a_n]\rangle],$$

where $u \in \bar{T}_\Sigma(X_1)$, $u/\alpha = x_1$, $r \in \hat{T}_\Sigma(X_n)$, $n \geqslant 0$, and $a_1,\dots,a_n \in B \cup E$. By Lemma 3.9,

$$p = u[r[p_1,\dots,p_n]],$$

for each $1 \leqslant i \leqslant n$, if the variable x_i appears in the tree r, then $p_i \in T_\Sigma$ and $p_i \to_{R_0}^{*} \langle a_i\rangle$. Finally, $t = u[c]$. By (b) of the definition of rules of R_i, $i \geqslant 1$, there is a rule $l \to r$ in R with $l \in \bar{T}_\Sigma(X_n)$, $n \geqslant 0$, and there are states and trees $a_i' \in B \cup E$ for $1 \leqslant i \leqslant n$ such that for each $1 \leqslant i \leqslant n$, $a_i' = a_i$ if x_i appears in the tree r, and that

$$l[\langle a_1'\rangle,\dots,\langle a_n'\rangle] \xrightarrow[R_{i-1}]{*} c.$$

As $\mathscr{C}_{i-1}$ is connected, there are trees $q_1,\dots,q_n \in T_\Sigma$ such that for each $1 \leqslant i \leqslant n$, if x_i appears in the tree r, then $q_i = p_i$, and that $q_i \to_{R_{i-1}}^{*} a_i'$. Let

$$s = u[l[q_1,\dots,q_n]].$$

Then

$$s \xrightarrow[R]{} p$$

and

$$s = u[l[q_1,\dots,q_n]] \xrightarrow[R_0]{*} u[l[a_i',\dots,a_n']] \xrightarrow[R_{i-1}]{*} u[c] = t.$$

Hence (2) holds. $\square$

Lemma 3.11. *For every $i \geqslant 0$, $p \in T_\Sigma$, $q \in T_{\Sigma \cup C}$, if $p \to_{R_i}^{*} q$, then there is an $s \in T_\Sigma$ such that*

$$s \xrightarrow[R]{*} p \quad \textit{and} \quad s \xrightarrow[R_0]{*} q.$$

Proof. We proceed by induction on i. For $i = 0$ the statement is trivial. Let us suppose that $i \geqslant 1$ and that we have shown the statement for $1, 2, \dots, i-1$. Let

$$p \xrightarrow[R_i]{*} q, \tag{3}$$

and let m be the number of $(R_i - R_{i-1})$-rules applied by $\mathscr{C}_i$ along (3). We show by induction on m that

$$\text{there is an } s \in T_\Sigma \text{ such that } s \xrightarrow[R]{*} p \text{ and } s \xrightarrow[R_0]{*} q. \tag{4}$$

If $m = 0$, then $p \to_{R_{i-1}}^{*} q$ and hence by the induction hypothesis on i, (4) holds.

Let us suppose that $m \geqslant 1$ and that for $0, 1, \ldots, m-1$, we have shown (4). Let $p \rightarrow_{R_i}^* q$ where $\mathscr{C}$ applies m $(R_i - R_{i-1})$-rules. Then there are integers n, k, $1 \leqslant k \leqslant n$, and there are trees $t_1, t_2, u_1, u_2, \ldots, u_n \in T_{\Sigma \cup C}$ such that (I), (II), (III), and (IV) hold.

(I) $p = u_1 \rightarrow_{R_i} \cdots \rightarrow_{R_i} u_k = t_1 \rightarrow_{R_i} u_{k+1} = t_2 \rightarrow_{R_i} \cdots \rightarrow_{R_i} u_n = q$.

(II) along the reduction subsequence $p = u_1 \rightarrow_{R_i} \cdots \rightarrow_{R_i} u_k = t_1$ of (I), $\mathscr{C}_i$ applies no $(R_i - R_{i-1})$-rule.

(III) in the rewrite step $u_k \rightarrow_{R_i} u_{k+1}$ $\mathscr{C}_i$ applies an $(R_i - R_{i-1})$-rule.

(IV) along the reduction subsequence $t_2 = u_{k+1} \rightarrow_{R_i} \cdots \rightarrow_{R_i} u_n = q$ of (I), $\mathscr{C}_i$ applies $m-1$ $(R_i - R_{i-1})$-rules.

By the induction hypothesis on i, there is a tree $s_1 \in T_\Sigma$ such that

$$s_1 \overset{*}{\underset{R}{\rightarrow}} p \quad \text{and} \quad s_1 \overset{*}{\underset{R_0}{\rightarrow}} t_1. \tag{5}$$

Hence

$$s_1 \overset{*}{\underset{R_0}{\rightarrow}} t_1 \underset{R_i}{\rightarrow} t_2.$$

By Lemma 3.10, there is a tree $s_2 \in T_\Sigma$ such that

$$s_2 \underset{R}{\rightarrow} s_1 \quad \text{and} \quad s_2 \overset{*}{\underset{R_{i-1}}{\rightarrow}} t_2. \tag{6}$$

Hence there is $j \geqslant 0$ and there are $w_1, \ldots, w_j \in T_{\Sigma \cup C}$ such that

$$s_2 = w_1 \underset{R_{i-1}}{\rightarrow} w_2 \underset{R_{i-1}}{\rightarrow} \cdots \underset{R_{i-1}}{\rightarrow} w_j = t_2 = u_{k+1} \underset{R_i}{\rightarrow} \cdots \underset{R_i}{\rightarrow} u_n = q, \tag{7}$$

and along (7), $\mathscr{C}_i$ applies $m-1$ $(R_i - R_{i-1})$-rules. By the induction hypothesis on m, there is a tree $s_3 \in T_\Sigma$ such that

$$s_3 \overset{*}{\underset{R}{\rightarrow}} s_2 \quad \text{and} \quad s_3 \overset{*}{\underset{R_0}{\rightarrow}} q.$$

Hence by (5) and (6),

$$s_3 \overset{*}{\underset{R}{\rightarrow}} s_2 \underset{R}{\rightarrow} s_1 \overset{*}{\underset{R}{\rightarrow}} p.$$

Thus (4) holds. □

Theorem 3.12. $L(\mathscr{C}) \subseteq R^*(L)$.

Proof. Let $p \in L(\mathscr{C})$. Then $p \rightarrow_S^* b$ for some $b \in B'$. Hence by Lemma 3.11, there is an $s \in T_\Sigma$ such that

$$s \overset{*}{\underset{R}{\rightarrow}} p \quad \text{and} \quad s \overset{*}{\underset{R_0}{\rightarrow}} b. \tag{8}$$

Hence $s \in L(\mathscr{C}_0)$. By Lemma 3.7, $s \in L$. Thus by (8), $p \in R^*(L)$. □

Now we show the inclusion $R^*(L) \subseteq L(\mathscr{C})$. To this end, first we prove five lemmas.

Lemma 3.13. *Let $l_1 \to r_1$ and $l_2 \to r_2$ be rules in R. Let $\alpha \in O(r_1)$, where $r_1/\alpha \in T_\Sigma(X_j)$, $j \geqslant 0$. Let $\beta \in O(l_2)$, where $l_2/\beta \in T_\Sigma(X) - X$, and let $s \in \bar{T}_\Sigma(X_k) - X$, $k \geqslant 1$, be a supertree of l_2/β. Let $\alpha = \lambda$ or $\beta = \lambda$. Let*

$$(r_1/\alpha)[a_1,\ldots,a_j] = s[z_1,\ldots,z_k], \tag{9}$$

where $a_1,\ldots,a_j \in B \cup E$, $z_1,\ldots,z_k \in T_{\Sigma \cup B}$. Let $\gamma \in O(s)$ be such that $l_2/\beta\gamma \in X$, and $s/\gamma = x_v$, for some $1 \leqslant v \leqslant k$. Then $z_v \in B \cup E$.

Proof. Let $l_1 \in T_\Sigma(X_m)$ for some $m \geqslant 0$. Let $l_3 = s[x_{m+1},\ldots,x_{m+k}]$. Then $l_3 \in T_\Sigma(\{x_{m+1},\ldots,x_{m+k}\})$ is a supertree of l_2/β, for each $m+1 \leqslant i \leqslant m+k$, x_i appears exactly once in l_3. Moreover, $var(l_1) \cap var(l_3) = \emptyset$, and by (9),

$$(r_1/\alpha)[a_1,\ldots,a_j] = l_3[x_{m+1} \leftarrow z_1,\ldots,x_{m+k} \leftarrow z_k]. \tag{10}$$

Let $\sigma_1 : X \to T_\Sigma(X)$ be a most general unifier of r_1/α and l_3. By (10), there is a substitution $\sigma_2 : X \to T_{\Sigma \cup B}(X)$ such that

$$\sigma_2(\sigma_1(r_1/\alpha)) = (r_1/\alpha)[a_1,\ldots,a_j] = l_3[x_{m+1} \leftarrow z_1,\ldots,x_{m+k} \leftarrow z_k] = \sigma_2(\sigma_1(l_3)),$$

where $\sigma_2(\sigma_1(x_i)) = a_i$ for $1 \leqslant i \leqslant j$ and $\sigma_2(\sigma_1(x_{m+i})) = z_i$ for $1 \leqslant i \leqslant k$. By Definition 3.1 and by the definition of E, $\sigma_1(x_{m+v}) \in X \cup E$. If $\sigma_1(x_{m+v}) \in X$, then $\sigma_2(\sigma_1(x_{m+v}))$ is a subtree of a_μ for some $\mu \in \{1,\ldots,j\}$. Hence by the definition of E, $z_v = \sigma_2(\sigma_1(x_{m+v})) \in B \cup E$. If $\sigma_1(x_{m+v}) \in E$, then $z_v = \sigma_2(\sigma_1(x_{m+v})) = \sigma_1(x_{m+v}) \in E$. □

Intuitively, the following lemma states that along a reduction sequence of S we can reverse the order of the consecutive application of a (ii)-type rule of R_0 at $\alpha \in N^*$ and the application of an $(S - R_0)$-rule at $\beta \in N^*$ if α is not a prefix of β and β is not a prefix of α.

Lemma 3.14. *Let*

$$u_1 \underset{S}{\rightarrow} u_2 \underset{S}{\rightarrow} u_3$$

be a reduction sequence of $\mathscr{C}$. Let $\alpha \in O(u_1)$, and $\beta \in O(u_2)$ be such that $u_1 \to_S u_2$ applying a (ii)*-type rule* **rule**$_1$ *of R_0 at α, and that $u_2 \to_S u_3$ applying an $(S - R_0)$-rule* **rule**$_2$ *at β. If α is not a prefix of β and β is not a prefix of α, then there is a tree $v \in T_{\Sigma \cup C}$ such that $u_1 \to_S v$ applying* **rule**$_2$ *at β, and $v \to_S u_3$ applying* **rule**$_1$ *at α.*

Proof. Straightforward. □

Lemma 3.15. *Let $i \geqslant 0$, $t \in \bar{T}_{\Sigma \cup C}(X_1)$, $\alpha \in O(t)$, $t/\alpha = x_1$, $c \in \{1,\ldots,|D - B|\}$, and $b \in B$. Let*

$$t[c] = u_1 \underset{R_i}{\rightarrow} u_2 \underset{R_i}{\rightarrow} \cdots \underset{R_i}{\rightarrow} u_n = b \tag{11}$$

with $n \geqslant 1$, $u_1,\ldots,u_n \in T_{\Sigma \cup C}$. Then along (11), $\mathscr{C}_i$ applies a rule in $R_i - R_0$ at some prefix β of α.

Proof. By direct inspection of the construction of the $\mathscr{C}_i$'s. $\square$

Lemma 3.16. *For any $n \geqslant 0$, $u \in \bar{T}_\Sigma(X_n)$, $v_1, \ldots, v_n, v \in D$, $m \geqslant 1$, and $w_1, \ldots, w_m \in T_{\Sigma \cup C}$, if*

$$u[\langle v_1 \rangle, \ldots, \langle v_n \rangle] = w_1 \underset{S}{\rightarrow} w_2 \underset{S}{\rightarrow} \cdots \underset{S}{\rightarrow} w_m = \langle v \rangle, \tag{12}$$

and $\mathscr{C}$ applies only (ii)*-type rules of R_0 along* (12), *then $u[v_1, \ldots, v_n] = v$.*

Proof. We proceed by induction on *depth*(u). The basis *depth*(u) = 0 of the induction is trivial. The induction step is a simple consequence of (ii) in the definition of R_0 and of the inclusion $R_0 \subseteq S$. $\square$

Lemma 3.17. *Let $t \in L(\mathscr{C})$, $m \geqslant 1$, $t_1, \ldots, t_m \in T_{\Sigma \cup C}$, $b \in B'$, and let*

$$t = t_1 \underset{S}{\rightarrow} t_2 \underset{S}{\rightarrow} t_3 \underset{S}{\rightarrow} \cdots \underset{S}{\rightarrow} t_m = b. \tag{13}$$

Let $l \rightarrow r$ be a rule in R, where $l \in \bar{T}_\Sigma(X_n)$ and $n \geqslant 1$. Moreover, let $1 \leqslant j \leqslant m$, and let

$$t_j/\alpha = l[\langle v_1 \rangle, \ldots, \langle v_n \rangle], \tag{14}$$

where $n \geqslant 1$, $v_1, \ldots, v_n \in D$, $\alpha \in O(t_j)$. Let $\alpha_1, \ldots, \alpha_n \in O(l)$ such that

$$l/\alpha_i = x_i \quad \text{for } 1 \leqslant i \leqslant n. \tag{15}$$

Consider the reduction subsequence

$$t_j \underset{S}{\rightarrow} t_{j+1} \underset{S}{\rightarrow} \cdots \underset{S}{\rightarrow} t_m = b \tag{16}$$

of (13). *If $\mathscr{C}$ does not apply any rules at the occurrences $\alpha\alpha_1, \ldots, \alpha\alpha_n$ along* (16), *then $v_1, \ldots, v_n \in B \cup E$.*

Proof. Let $1 \leqslant i \leqslant n$, and let us assume that $v_i \in D - B$. By (14) and (15),

$$t_j/\alpha\alpha_i = \langle v_i \rangle. \tag{17}$$

By Lemma 3.15, $\mathscr{C}$ applies a rule in $S - R_0$ at some prefix of $\alpha\alpha_i$ along (16). Let $\beta \in O(t_j)$ be the longest prefix of $\alpha\alpha_i$ such that $\mathscr{C}$ applies a rule **rule** in $S - R_0$ at β along (16). Then **rule** is of the form $\langle r_1[a_1, \ldots, a_\kappa] \rangle \rightarrow c$, where $\kappa \geqslant 0$, $r_1 \in \bar{T}_\Sigma(X_\kappa)$, $a_1, \ldots, a_\kappa \in B \cup E$, and there is a rule $l_1 \rightarrow r_1$ in R. Moreover there exists ξ, $j < \xi \leqslant m$, such that

$$t_j/\beta \overset{*}{\underset{S}{\rightarrow}} t_{j+1}/\beta \overset{*}{\underset{S}{\rightarrow}} \cdots \overset{*}{\underset{S}{\rightarrow}} t_\xi/\beta = \langle r_1[a_1, \ldots, a_\kappa] \rangle ,$$

where for each π, $j \leqslant \pi \leqslant \xi - 1$, $t_\pi/\beta = t_{\pi+1}/\beta$ or $t_\pi/\beta \rightarrow_S t_{\pi+1}/\beta$. We lose no generality by assuming that

$$t_j/\beta \underset{S}{\rightarrow} t_{j+1}/\beta \underset{S}{\rightarrow} \cdots \underset{S}{\rightarrow} t_\xi/\beta = \langle r_1[a_1, \ldots, a_\kappa] \rangle . \tag{18}$$

By Lemma 3.14 we may assume that there exists ν, $j \leqslant \nu \leqslant \xi$ such that

(a) along the reduction subsequence

$$t_j/\beta \underset{S}{\rightarrow} \cdots \underset{S}{\rightarrow} t_\nu/\beta \tag{19}$$

of (18) no rule is applied at any prefix of $\alpha\alpha_i$, that

(b) along (19) each application of a (ii)-type rule of R_0 at some $\delta \in N^*$ is followed somewhere later by an application of an $S - R_0$-rule of S at a prefix ε of δ, and that

(c) along the reduction subsequence

$$t_\nu/\beta \underset{S}{\rightarrow} \cdots \underset{S}{\rightarrow} t_\xi/\beta = \langle r_1[a_1,\ldots,a_\kappa]\rangle$$

of (18), S applies only (ii)-type rules of R_0.

Then

$$t_\nu/\beta = s[\langle z_1\rangle,\ldots,\langle z_k\rangle] \tag{20}$$

for some $k \geqslant 1$, $s \in \bar{T}_\Sigma(X_k)$, and $\langle z_1\rangle,\ldots,\langle z_k\rangle \in C$. By (20), (c) of the definition of ν, and Lemma 3.16,

$$s[z_1,\ldots,z_k] = r_1[a_1,\ldots,a_\kappa]. \tag{21}$$

The word α is a prefix of β or β is a prefix of α. Hence we can distinguish two cases.

Case 1: α is a prefix of β, see Fig. 4. In this case,

$$\beta = \alpha\gamma \tag{22}$$

for some $\gamma \in N^*$, and hence t_ν/β is a subtree of t_ν/α. Now by (14), the definition of ν, and (20),

$$s \text{ is a supertree of } l/\gamma. \tag{23}$$

Let ω be the pefix of $\alpha\alpha_i$ with $length(\omega) = length(\alpha\alpha_i) - 1$. Observe that $\mathscr{C}$ applies a (ii)-type rule of R_0 at the occurrence ω along (16). Hence

$$s \notin X\,. \tag{24}$$

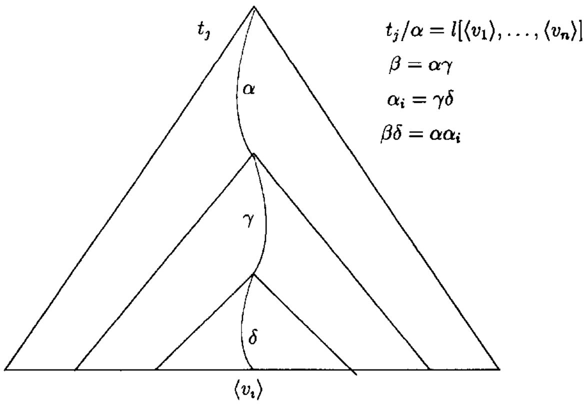

Fig. 4. Case 1.

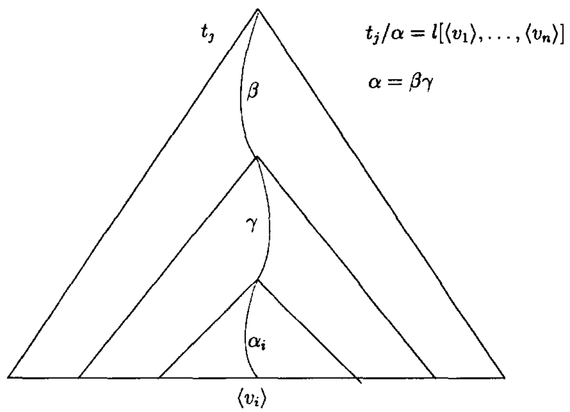

Fig. 5. Case 2.

Let $\delta \in N^*$ be defined by the equation $\gamma\delta = \alpha_i$. Then

$$\beta\delta = \alpha\alpha_i, \tag{25}$$

and by (a) of the definition of v,

$$\delta \in O(s), \qquad \delta \in O(l/\gamma) \quad \text{and} \quad (l/\gamma)/\delta = x_i. \tag{26}$$

By (25) and by (a) of the definition of v,

$$\beta\delta \in O(t_v).$$

By (17), (25), (a) of the definition of v, and (20),

$$\langle v_i\rangle = (t_j/\beta)/\delta = (t_v/\beta)/\delta = s[\langle z_1\rangle, \ldots, \langle z_k\rangle]/\delta = \langle z_\mu\rangle \tag{27}$$

for some $1 \leqslant \mu \leqslant k$. As R is gsm, by (23), (24), (26), (21), and Lemma 3.13, $z_\mu \in B \cup E$. By (27), $v_i \in B \cup E$.

Case 2: β is a prefix of α, see Fig. 5. In this case

$$\alpha = \beta\gamma \tag{28}$$

for some $\gamma \in N^*$, and hence t_j/α is a subtree of t_j/β. Now by (14), the definition of v, and (20),

$$s/\gamma \text{ is a supertree of } l. \tag{29}$$

Moreover, by (a) of the definition of v,

$$\alpha_i \in O(s/\gamma), \qquad l/\alpha_i \in X \quad \text{and} \quad (s/\gamma)/\alpha_i \in X. \tag{30}$$

Let ω be the pefix of $\alpha\alpha_i$ with $length(\omega) = length(\alpha\alpha_i) - 1$. Observe that $\mathscr{C}$ applies a (ii)-type rule of R_0 at the occurrence ω along (16). Hence

$$s/\gamma \notin X\ . \tag{31}$$

By (28) and by (a) of the definition of v,

$$\beta\gamma\alpha_i = \alpha\alpha_i \in O(t_v). \tag{32}$$

Then by (17), (32), (a) of the definition of v, and (20),

$$\langle v_i \rangle = (t_j/\beta)/\gamma\alpha_i = (t_v/\beta)/\gamma\alpha_i = s[\langle z_1 \rangle, \dots, \langle z_k \rangle]/\gamma\alpha_i = \langle z_\mu \rangle \tag{33}$$

for some $1 \leqslant \mu \leqslant k$. By (21),

$$(s/\gamma)[z_1, \dots, z_k] = s[z_1, \dots, z_k]/\gamma = r_1[a_1, \dots, a_\kappa]/\gamma. \tag{34}$$

As R is gsm, by (29), (31), (30), (33), (34), and Lemma 3.13, $z_\mu \in B \cup E$. By (33), $v_i \in B \cup E$. □

Theorem 3.18. $R^*(L) \subseteq L(\mathscr{C})$.

Proof. By (i) in the definition of R_0, $R_{\mathscr{B}} \subseteq R_0$. Hence $L \subseteq L(\mathscr{C}_0)$. As $R_{i-1} \subseteq R_i$ for $i \geqslant 1$, we have $L \subseteq L(\mathscr{C}_i)$ for $i \geqslant 0$. Hence $L \subseteq L(\mathscr{C})$. Thus it is sufficient to show that for each $t \in L(\mathscr{C})$, if $t \to_R t'$, then $t' \in L(\mathscr{C})$. To this end, let us suppose that $t \to_R t'$, applying the rule $l \to r$ in R at $\alpha \in O(t)$. Here $l \in \bar{T}_\Sigma(X_n)$ for some $n \geqslant 0$. Let $\alpha_1, \dots, \alpha_n \in O(l)$ be such that

$$l/\alpha_i = x_i \quad \text{for } 1 \leqslant i \leqslant n\,.$$

Then

$$t = s[l[u_1, \dots, u_n]],$$

where $s \in \bar{T}_\Sigma(X_1)$, $\alpha \in O(s)$, $s/\alpha = x_1$, and $u_1, \dots, u_n \in T_\Sigma$. Moreover,

$$t' = t[\alpha \leftarrow r[u_1, \dots, u_n]] = s[r[u_1, \dots, u_n]].$$

As $t \in L(\mathscr{C})$, there is a reduction sequence

$$t = t_1 \underset{S}{\to} t_2 \underset{S}{\to} t_3 \underset{S}{\to} \cdots \underset{S}{\to} t_m = b, \tag{35}$$

where $m \geqslant 1$, $b \in B'$, $t_1, \dots, t_m \in T_{\Sigma \cup C}$, and there are integers j, k with $1 \leqslant j, k \leqslant m$ such that

(i) $t_j = s[l[\langle v_1 \rangle, \dots, \langle v_n \rangle]]$, where $v_i \in D$ and $u_i \to_S^* \langle v_i \rangle$ for $1 \leqslant i \leqslant n$,

(ii) $t_k = s[c_0]$, for some $c_0 \in C$, where $l[\langle v_1 \rangle, \dots, \langle v_n \rangle] \to_S^* c_0$, and that

(iii) along the reduction subsequence $t_j \to_S t_{j+1} \to_S \cdots \to_S t_k$ of (35), $\mathscr{C}$ does not apply any rules at the occurrences $\alpha\alpha_1, \dots, \alpha\alpha_n$. By Lemma 3.17, $v_1, \dots, v_n \in B \cup E$. Hence by Condition (ii) in the definition of R_i, $i \geqslant 1$, and by the definition of $\mathscr{C}$, the rule $r[\langle v_1 \rangle, \dots, \langle v_n \rangle] \to c_0$ is in S. Thus we get

$$t' = s[r[u_1, \dots, u_n]] \underset{S}{\overset{*}{\to}} s[r[\langle v_1 \rangle, \dots, \langle v_n \rangle]] \underset{S}{\to} s[c_0] \underset{S}{\overset{*}{\to}} b.$$

As $b \in B'$, $t' \in L(\mathscr{C})$. □

By Theorems 3.12 and 3.18, we get the following.

Theorem 3.19. $R^*(L) = L(\mathscr{C})$.

As Δ, R, Σ ($\Delta \subseteq \Sigma$), and $\mathscr{B}$ are arbitrary, we have the following result.

Theorem 3.20. *Linear generalized semi-monadic rewrite systems effectively preserve recognizability.*

Theorem 3.21. *A tree language L is recognizable if and only if there exists a rewrite system R such that $R \cup R^{-1}$ is a rewrite system preserving recognizability and that L is the union of finitely many $\leftrightarrow_R^*$-classes.*

Proof. Let us assume that L is recognizable. Then by Proposition 2.2 there is a ground rewrite system R such that L is the union of finitely many $\leftrightarrow_R^*$-classes. Clearly, $R \cup R^{-1}$ is an lgsm rewrite system and hence, by Theorem 3.20, preserves recognizability.

Let us assume that there exists a rewrite sys tem R such that $R \cup R^{-1}$ is a rewrite system preserving recognizability and that L is the union of finitely many $\leftrightarrow_R^*$-classes. That is to say, $L = [t_1]_R \cup [t_2]_R \cup \cdots \cup [t_k]_R$ for some $k \geqslant 0$. As $\to_{R \cup R^{-1}}^* = \leftrightarrow_R^*$, $L = (R \cup R^{-1})^*(\{t_1, \ldots, t_k\})$. It should be clear that the tree language $\{t_1, \ldots, t_k\}$ is recognizable. Since $R \cup R^{-1}$ preserves recognizability, L is also recognizable. $\square$

The following result is a simple consequence of our results.

Theorem 3.22. *A tree language L is recognizable if and only if there exists a rewrite system R such that $R \cup R^{-1}$ is an lgsm rewrite system and that L is the union of finitely many $\leftrightarrow_R^*$-classes.*

Example 3.23. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, $\Sigma_0 = \{\sharp, \$\}$, $\Sigma_1 = \{f\}$, $\Sigma_2 = \{g\}$. Let R consist of the rules

$$g(g(x_1, \$), x_2) \to f(g(g(\$, x_1), x_2)),$$

$$g(g(\$, x_2), x_1) \to f(g(g(\$, x_1), x_2)).$$

Then $R \cup R^{-1}$ is an lgsm rewrite system. Hence, by Theorem 3.22, the union of finitely many arbitrary $\leftrightarrow_R^*$-classes is recognizable.

It is well known that the symbols of an alphabet Σ can be considered as unary function symbols, and hence words over Σ can be considered as unary trees. For example the word *apple* can be considered as the tree $a(p(p(l(e(\sharp)))))$, where $\sharp \notin \Sigma$ is a symbol of rank 0. Then recognizable string languages over Σ are the same as recognizable tree languages over the ranked alphabet $\Sigma \cup \{\sharp\}$.

Let R be a string rewrite system over Σ. We can consider R as a rewrite system as follows. The left-hand sides and the right-hand sides of the rules in R can be considered as trees containing the variable x_1 instead of $\sharp$. For example, the string rewrite rule

$$apple \to peach$$

can be considered as the rewrite rule

$$a(p(p(l(e(x_1))))) \to p(e(a(c(h(x_1))))).$$

Hence our concepts and results carry over to strings as well. Let R be a string rewrite system. We say that R is restricted right–left overlapping if there is no rule $\lambda \to r$ in R, and the following holds. For any rules $l_1 \to r_1$ and $l_2 \to r_2$ in R, for any nonempty suffix $u \in \Sigma^+$ of r_1 and any nonempty suffix $v \in \Sigma^+$ of l_2, if $u = r_1$ or $v = l_2$, then v cannot be a proper prefix of u. For example the string rewrite system

$$\{apple \to peach\}$$

is restricted right–left overlapping.

A string rewrite system R is monadic if $(l,r) \in R$ implies that $|l| > |r|$ and ($|r| = 1$ or $|r| = 0$). It is not hard to see that each monadic rewrite system is restricted right–left overlapping as well. It should be clear what we mean when we say that a string rewrite system R over Σ effectively preserves Σ-recognizability (recognizability, resp.). It is well known that monadic rewrite systems effectively preserve recognizability, see Theorem 4.1.2 in [1]. The following theorem is a generalization of this result and is an interesting consequence of our results on rewrite systems.

Theorem 3.24. *Restricted right–left overlapping string rewrite systems effectively preserve recognizability. Moreover, a string language L is recognizable if and only if there exists a string rewrite system R such that $R \cup R^{-1}$ is a restricted right–left overlapping string rewrite system and that L is the union of finitely many $\leftrightarrow_R^*$-classes.*

Let R be a string rewrite system. We say that R is λ-free if there is no rule $l \to r$ in R such that l or r is the empty word.

4. An example

In this section we illustrate the construction of $\mathscr{C}_j$, $j \geqslant 0$, appearing in the previous section by an example. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, $\Sigma_0 = \{\sharp\}$, $\Sigma_1 = \{f\}$, $\Sigma_2 = \{g\}$. Let the rewrite system R over Σ consist of the following two rules.

$$f(f(g(x_1, \sharp))) \to f(f(x_1)),$$

$$g(x_1, x_2) \to f(g(x_1, \sharp)).$$

By direct inspection we obtain that R is an lgsm rewrite system. Here $E = \{\sharp\}$. Let $L = \{g(\sharp, \sharp)\}$. It is not hard to see that

$$R^*(L) = \{f^n(g(\sharp, \sharp)) \mid n \geqslant 0\} \cup \{f^n(\sharp) \mid n \geqslant 2\}.$$

Consider the bottom-up tree automaton $\mathscr{B} = (\Sigma, B, R_{\mathscr{B}}, B')$, where $B = \{b_1, b_2\}$, $B' = \{b_2\}$, and $R_{\mathscr{B}}$ consists of the following two rules: $\sharp \to b_1$, $g(b_1, b_1) \to b_2$. It is not hard to show that $L=L(\mathscr{B})$. By direct inspection we obtain that the set of subterms of the right-hand sides of the rules of R is

$$\{x_1, f(x_1), f(f(x_1)), \sharp, g(x_1,\sharp), f(g(x_1,\sharp))\}.$$

Then

$$D=\{b_1,b_2,\sharp,f(b_1),f(b_2),f(\sharp),f(f(b_1)),f(f(b_2)),f(f(\sharp)),g(b_1,\sharp),g(b_2,\sharp),$$
$$g(\sharp,\sharp),f(g(b_1,\sharp)),f(g(b_2,\sharp)),f(g(\sharp,\sharp))\}.$$

Moreover, $C=\{b_1,b_2,1,\dots,13\}$. Let $\langle\ \rangle : D\to\{1,\dots,|D-B|\}$ be defined by

$$\begin{array}{lll}
\langle b_1\rangle=b_1, & \langle b_2\rangle=b_2, & \langle\sharp\rangle=1,\\
\langle f(b_1)\rangle=2, & \langle f(b_2)\rangle=3, & \langle f(\sharp)\rangle=4,\\
\langle f(f(b_1))\rangle=5, & \langle f(f(b_2))\rangle=6, & \langle f(f(\sharp))\rangle=7,\\
\langle g(b_1,\sharp)\rangle=8, & \langle g(b_2,\sharp)\rangle=9, & \langle g(\sharp,\sharp)\rangle=10,\\
\langle f(g(b_1,\sharp))\rangle=11, & \langle f(g(b_2,\sharp))\rangle=12, & \langle f(g(\sharp,\sharp))\rangle=13.
\end{array}$$

Then $\mathscr{C}_0=(\Sigma,C,R_0,B')$ is determined by the set R_0 of rules. R_0 consists of the following fifteen rules.

$$\begin{array}{ll}
\sharp\to b_1, & g(b_1,b_1)\to b_2,\\
\sharp\to\langle\sharp\rangle, & f(b_1)\to\langle f(b_1)\rangle,\\
f(b_2)\to\langle f(b_2)\rangle, & f(\langle\sharp\rangle)\to\langle f(\sharp)\rangle,\\
f(\langle f(b_1)\rangle)\to\langle f(f(b_1))\rangle, & f(\langle f(b_2)\rangle)\to\langle f(f(b_2))\rangle,\\
f(\langle f(\sharp)\rangle)\to\langle f(f(\sharp))\rangle, & g(b_1,\langle\sharp\rangle)\to\langle g(b_1,\sharp)\rangle,\\
g(b_2,\langle\sharp\rangle)\to\langle g(b_2,\sharp)\rangle, & g(\langle\sharp\rangle,\langle\sharp\rangle)\to\langle g(\sharp,\sharp)\rangle,\\
f(\langle g(b_1,\sharp)\rangle)\to\langle f(g(b_1,\sharp))\rangle, & f(\langle g(b_2,\sharp)\rangle)\to\langle f(g(b_2,\sharp))\rangle,\\
f(\langle g(\sharp,\sharp)\rangle)\to\langle f(g(\sharp,\sharp))\rangle. &
\end{array}$$

That is, R_0 consists of the following fifteen rules.

$$\begin{array}{lll}
\sharp\to b_1, & g(b_1,b_1)\to b_2, & \sharp\to 1,\\
f(b_1)\to 2, & f(b_2)\to 3, & f(1)\to 4,\\
f(2)\to 5, & f(3)\to 6, & f(4)\to 7,\\
g(b_1,1)\to 8, & g(b_2,1)\to 9, & g(1,1)\to 10,\\
f(8)\to 11, & f(9)\to 12, & f(10)\to 13.
\end{array}$$

The bottom–up tree automaton $\mathscr{C}_1 = (\Sigma, C, R_1, B')$ is determined by the set R_1 of rules. R_1 contains all rules of R_0 and the following five rules.

$$\langle f(f(b_1))\rangle \to \langle f(f(b_2))\rangle, \quad \langle f(g(b_1,\sharp))\rangle \to b_2,$$
$$\langle f(g(b_1,\sharp))\rangle \to \langle g(b_1,\sharp)\rangle, \quad \langle f(g(b_2,\sharp))\rangle \to \langle g(b_2,\sharp)\rangle,$$
$$\langle f(g(\sharp,\sharp))\rangle \to \langle g(\sharp,\sharp)\rangle.$$

That is, R_1 contains all rules of R_0 and the following five rules.

$$5\to 6, \; 11\to b_2, \; 11\to 8, \; 12\to 9, \; 13\to 10.$$

The bottom–up tree automaton $\mathscr{C}_2 = (\Sigma, C, R_2, B')$ is determined by the set R_2 of rules. R_2 contains all rules of R_1 and the following seven rules.

$$\langle f(f(b_1))\rangle \to \langle f(g(b_1,\sharp))\rangle, \quad \langle f(f(b_1))\rangle \to \langle g(b_1,\sharp)\rangle,$$
$$\langle f(f(b_1))\rangle \to b_2, \quad \langle f(f(b_2))\rangle \to \langle f(g(b_2,\sharp))\rangle,$$
$$\langle f(f(b_2))\rangle \to \langle g(b_2,\sharp)\rangle, \quad \langle f(f(\sharp))\rangle \to \langle f(g(\sharp,\sharp))\rangle,$$
$$\langle f(f(\sharp))\rangle \to \langle g(\sharp,\sharp)\rangle.$$

That is, R_2 contains all rules of R_1 and the following seven rules.

$$5\to 11, \; 5\to 8, \; 5\to b_2, \; 6\to 12, \; 6\to 9, \; 7\to 13, \; 7\to 10.$$

The bottom–up tree automaton $\mathscr{C}_3 = (\Sigma, C, R_3, B')$ is determined by the set R_3 of rules. R_3 contains all rules of R_2 and the following two rules.

$$\langle f(f(b_1))\rangle \to \langle f(g(b_2,\sharp))\rangle, \quad \langle f(f(b_1))\rangle \to \langle g(b_2,\sharp)\rangle.$$

That is, R_3 contains all rules of R_2 and the following two rules.

$$5\to 12, \quad 5\to 9.$$

Since $R_4 = R_3$, the bottom–up tree automaton $\mathscr{C}_4 = (\Sigma, C, R_4, B')$ is equal to $\mathscr{C}_3 = (\Sigma, C, R_3, B')$. Let $S = R_4$ and let us write $\mathscr{C} = (\Sigma, C, S, B')$ for $\mathscr{C}_4 = (\Sigma, C, R_4, B')$. Hence S consists of the following twenty-nine rules.

$$\begin{array}{lll}
\sharp\to b_1, & g(b_1,b_1)\to b_2, & \sharp\to 1,\\
f(b_1)\to 2, & f(b_2)\to 3, & f(1)\to 4,\\
f(2)\to 5, & f(3)\to 6, & f(4)\to 7,\\
g(b_1,1)\to 8, & g(b_2,1)\to 9, & g(1,1)\to 10,\\
f(8)\to 11, & f(9)\to 12, & f(10)\to 13,\\
5\to 6, & 11\to b_2, & 11\to 8,\\
12\to 9, & 13\to 10, & 5\to 11,\\
5\to 8, & 5\to b_2, & 6\to 12,\\
6\to 9, & 7\to 13, & 7\to 10,\\
5\to 12, & 5\to 9. &
\end{array}$$

By direct inspection we obtain that the states 3, 4, 6, 7, 9, 10, 12, and 13 are superfluous as no final state can be reached from any of them. Hence we drop all of them and also omit all rules in which they appear. In this way we obtain the bottom–up tree automaton $\mathscr{A}_1 = (\Sigma, C, S_1, B')$, where S_1 consists of the following twelve rules.

$$\begin{array}{lll} \sharp \to b_1, & g(b_1, b_1) \to b_2, & \sharp \to 1, \\ f(b_1) \to 2, & f(2) \to 5, & g(b_1, 1) \to 8, \\ f(8) \to 11, & 11 \to b_2, & 11 \to 8, \\ 5 \to 11, & 5 \to 8, & 5 \to b_2. \end{array}$$

It is not hard to see that the rule $5 \to 11$ is superfluous. We obtain the bottom–up tree automaton $\mathscr{A}_2 = (\Sigma, C, S_2, B')$, from $\mathscr{A}_1$ by dropping the rule $5 \to 11$. Thus S_2 consists of the following eleven rules.

$$\begin{array}{lll} \sharp \to b_1, & g(b_1, b_1) \to b_2, & \sharp \to 1, \\ f(b_1) \to 2, & f(2) \to 5, & g(b_1, 1) \to 8, \\ f(8) \to 11, & 11 \to b_2, & 11 \to 8, \\ 5 \to 8, & 5 \to b_2. & \end{array}$$

We define the deterministic bottom–up tree automaton $\mathscr{A}_3 = (\Sigma, C, S_3, A')$, from $\mathscr{A}_2$ by applying the subset construction. Then S_3 consists of the seven following rules.

$$\begin{array}{ll} \sharp \to \{b_1, 1\}, & g(\{b_1, 1\}, \{b_1, 1\}) \to \{b_2, 8\}, \\ f(\{b_1, 1\}) \to \{2\}, & f(\{b_2, 8\}) \to \{8, 11, b_2\}, \\ f(\{2\}) \to \{5, 8, b_2\}, & f(\{5, 8, b_2\}) \to \{8, 11, b_2\}, \\ f(\{8, 11, b_2\}) \to \{8, 11, b_2\}. & \end{array}$$

Moreover, A' consists of the three states $\{b_2, 8\}, \{8, 11, b_2\}, \{5, 8, b_2\}$. Let us redenote the states of $\mathscr{A}_3$ as follows. Let

$$a_1 = \{b_1, 1\},\ a_2 = \{b_2, 8\},\ a_3 = \{2\},\ a_4 = \{8, 11, b_2\},\ a_5 = \{5, 8, b_2\}.$$

Hence S_3 consists of the following seven rules:

$$\begin{array}{lll} \sharp \to a_1, & g(a_1, a_1) \to a_2, & f(a_1) \to a_3, \\ f(a_2) \to a_4, & f(a_3) \to a_5, & f(a_5) \to a_4, \\ f(a_4) \to a_4. & & \end{array}$$

Moreover, $A' = \{a_2, a_4, a_5\}$.

It should be clear that the states a_2, a_4, a_5 are equivalent. Finally we construct a minimal deterministic bottom–up tree automaton $\mathscr{A}_4 = (\Sigma, C, S_3, A'')$, from $\mathscr{A}_3$ by merging the equivalent states a_2, a_4, a_5. Hence S_4 consists of the following five rules.

$$\sharp \to a_1,\ g(a_1, a_1) \to a_2,\ f(a_1) \to a_3,\ f(a_2) \to a_2,\ f(a_3) \to a_2.$$

Moreover, $A'' = \{a_2\}$. We obtain by direct inspection that $L(\mathscr{A}_4) = R^*(L)$.

5. Rewrite systems preserving recognizability

In this section we study rewrite systems preserving recognizability and gsm rewrite systems. First we present a ranked alphabet Σ and a linear rewrite system R over Σ such that R preserves Σ-recognizability but does not preserve recognizability.

Theorem 5.1. *There is a ranked alphabet Σ and there is a linear rewrite system R over Σ such that R preserves Σ-recognizability but does not preserve recognizability.*

Proof. Let $\Sigma = \Sigma_1 \cup \Sigma_0$, $\Sigma_1 = \{f, g\}$, $\Sigma_0 = \{\sharp\}$. Let R consist of the following five rules.

$$f(g(x_1)) \to f(f(g(g(x_1)))),\ f(\sharp) \to \sharp,\ g(\sharp) \to \sharp,\ \sharp \to f(\sharp),\ \sharp \to g(\sharp).$$

It should be clear that for each tree $t \in T_\Sigma$, $t \to_R^* \sharp$, and $\sharp \to_R^* t$. Hence for each nonempty tree language $L \subseteq T_\Sigma$, $R^*(L) = T_\Sigma$. Thus R preserves Σ-recognizability.

Let $\Delta = \Sigma \cup \{h\}$, where $h \in \Delta_1$. Let $L = \{f(g(h(\sharp)))\}$. Since L is finite, L is recognizable. However, $R^*(L) = \{f^n(g^n(h(t))) \mid n \geqslant 0,\ t \in T_\Sigma\}$ is not recognizable. □

Theorem 5.2. *Let R be a rewrite system over $sign(R)$, and let $\Sigma = \{f, \sharp\} \cup sign(R)$, where $f \in \Sigma_2 - sign(R)$ and $\sharp \in \Sigma_0 - sign(R)$. R preserves Σ-recognizability if and only if R preserves recognizability.*

Proof. ($\Leftarrow$) Trivial.

($\Rightarrow$) Let Δ be an arbitrary ranked alphabet with $sign(R) \subseteq \Delta$. To each symbol $g \in \Delta_k - sign(R)$, $k \geqslant 0$, we assign a tree $t_g \in T_\Sigma(X_k)$. To this end, we number the symbols in $\Delta - sign(R)$ from 1 to $|\Delta - sign(R)|$. Then we define the nth left comb $left_n$ and the nth right comb $right_n$ as follows.

(i) $left_0 = f(\sharp, \sharp)$ and $right_0 = \sharp$,

(ii) for each $n \geqslant 0$, $left_{n+1} = f(left_n, x_{n+1})$, $right_{n+1} = f(\sharp, right_n)$.

Finally, to a symbol $g \in \Delta_k - sign(R)$, $k \geqslant 0$, with number l, we assign the tree $t_g = f(left_k, right_l)$.

Consider the rewrite system

$$S = \{g(x_1, \ldots, x_k) \to t_g \mid k \geqslant 0,\ g \in \Delta_k - sign(R), t_g \text{ is assigned to } g\}.$$

It should be clear that S is a convergent rewrite system. For each tree $p \in T_\Delta$, we denote by p', the S-normal form of p. For a tree language $L \subseteq T_\Delta$, let $L' = \{p' \mid p \in L\}$. It is not hard to show the following two statements.

Claim 5.3. *For any $r, s \in T_\Delta$,*

$$r \underset{R}{\to} s \text{ if and only if } r' \underset{R}{\to} s'.$$

Claim 5.4. *A tree language L over Δ is recognizable if and only if L' is recognizable over Σ.*

Let L be any recognizable tree language over Δ. By Claim 5.4, L' is a recognizable tree language over Σ. By Claim 5.3, $(R^*_\Delta(L))' = R^*_\Sigma(L')$. By Claim 5.4, $R^*_\Delta(L)$ is recognizable if and only if $R^*_\Sigma(L')$ is recognizable. Hence if R preserves recognizability over Σ, then R preserves recognizability over Δ. As Δ is an arbitrary ranked alphabet with $sign(R) \subseteq \Delta$, by Lemma 2.3, R preserves recognizability. □

The proof of the following result is similar to the proof of Theorem 5.2.

Theorem 5.5. *Let R be a rewrite system over $sign(R)$, and let $\Sigma = \{f, \sharp\} \cup sign(R)$, where $f \in \Sigma_2 - sign(R)$ and $\sharp \in \Sigma_0 - sign(R)$. R effectively preserves Σ-recognizability if and only if R effectively preserves recognizability.*

Consequence 5.6. *Let R be a rewrite system over Σ such that there is a symbol $f \in \Sigma_2 - sign(R)$ and there is a constant $\sharp \in \Sigma_0 - sign(R)$. Then R preserves recognizability if and only if R preserves Σ-recognizability. Moreover, R effectively preserves recognizability if and only if R effectively preserves Σ-recognizability.*

Theorem 5.7. *Let R, S be rewrite systems over a ranked alphabet Σ. Let R effectively preserve recognizability. Then it is decidable if $\to^*_S \subseteq \to^*_R$.*

Proof. Let $m \geqslant 0$ be such that for all variables x_i occurring on the left-hand side of some rule in S, $x_i \in X_m$, that is, $i \leqslant m$. Let us introduce new constant symbols $Z = \{z_1, \ldots, z_m\}$ with $Z \cap \Sigma = \emptyset$. For each $t \in T_\Sigma(X)$, let $t_z \in T_{\Sigma \cup Z}(X)$ be defined by $t_z = t[z_1, \ldots, z_m]$. By direct inspection we obtain that for all $u, v \in T_\Sigma(X)$,

$$u \underset{R}{\to} v \text{ if and only if } u_z \underset{R}{\to} v_z ,$$

hence

$$u \overset{*}{\underset{R}{\to}} v \text{ if and only if } u_z \overset{*}{\underset{R}{\to}} v_z .$$

Claim 5.8. *$\to^*_S \subseteq \to^*_R$ if and only if for each rule $l \to r$ in S, $r_z \in R^*_{\Sigma \cup Z}(\{l_z\})$.*

Proof. ($\Rightarrow$) Let $l \to r$ be an arbitrary rule in S. Clearly, $l \to^*_R r$. Thus $r_z \in R^*_{\Sigma \cup Z}(\{l_z\})$.

($\Leftarrow$) Let us suppose that $t_1, t_2 \in T_\Sigma(X)$, and that $t_1 \to_S t_2$ applying the rule $l \to r$. As $r_z \in R^*_{\Sigma \cup Z}(\{l_z\})$, $l_z \to^*_R r_z$ holds. Hence $l \to^*_R r$ implying that $t_1 \to^*_R t_2$ as well. □

Let $l \to r$ be an arbitrary rule in S. We can construct a tree automaton over $\Sigma \cup Z$ recognizing the singleton set $\{l_z\}$. As R effectively preserves recognizability, $R^*_{\Sigma \cup Z}(\{l_z\})$ is recognizable, and we can construct a tree automaton over $\Sigma \cup Z$ recognizing $R^*_{\Sigma \cup Z}(\{l_z\})$. Hence we can decide if $r_z \in R^*_{\Sigma \cup Z}(\{l_z\})$. Thus by Claim 5.8, we can decide if $\to^*_S \subseteq \to^*_R$. □

Consequence 5.9. *Let R_1 and R_2 be rewrite systems effectively preserving recognizability. Then it is decidable which one of the following four mutually excluding conditions holds.*
(i) $\rightarrow^*_{R_1} \subset \rightarrow^*_{R_2}$,
(ii) $\rightarrow^*_{R_2} \subset \rightarrow^*_{R_1}$,
(iii) $\rightarrow^*_{R_1} = \rightarrow^*_{R_2}$,
(iv) $\rightarrow^*_{R_1} \bowtie \rightarrow^*_{R_2}$,
where "$\bowtie$" stands for the incomparability relationship.

Observation 5.10. *If one omits a rule from an lgsm rewrite system, then the resulting rewrite system still remains lgsm.*

One can easily show the following result applying Theorem 3.20, Consequence 5.9, and Observation 5.10.

Consequence 5.11. *For an lgsm rewrite system R, it is decidable whether R is left-to-right minimal.*

Consequence 5.9 also implies the following.

Consequence 5.12. *Let R_1 and R_2 be rewrite systems such that $R_1 \cup R_1^{-1}$ and $R_2 \cup R_2^{-1}$ are rewrite systems and effectively preserve recognizability. Then it is decidable which one of the following four mutually excluding conditions holds.*
(i) $\leftrightarrow^*_{R_1} \subset \leftrightarrow^*_{R_2}$,
(ii) $\leftrightarrow^*_{R_2} \subset \leftrightarrow^*_{R_1}$,
(iii) $\leftrightarrow^*_{R_1} = \leftrightarrow^*_{R_2}$,
(iv) $\leftrightarrow^*_{R_1} \bowtie \leftrightarrow^*_{R_2}$.

Theorem 3.20, Observation 5.10, and Consequence 5.12 imply the following.

Consequence 5.13. *Let R be a rewrite system such that $R \cup R^{-1}$ is an lgsm rewrite system. Then it is decidable whether R is two-way minimal.*

Theorem 5.14. *Let R_1, R_2 be rewrite systems over a ranked alphabet Σ. Let R_1 effectively preserve recognizability. Let $g \in \Sigma - \Sigma_0$ be such that g does not occur on the left-hand side of any rule in R_1, and let $\sharp \in \Sigma_0$ be irreducible for R_1. Then it is decidable if $\rightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma) \subseteq \rightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma)$.*

Proof. We assume that $g \in \Sigma_1$. One can easily modify the proof of this case when proving the more general case $g \in \Sigma_k$, $k \geqslant 1$. For each $t \in T_\Sigma(X)$, let $t_g \in T_\Sigma$ be defined from t by substituting $g^i(\sharp)$ for all occurrences of the variable x_i for $i \geqslant 1$.

Claim 5.15. $\rightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma) \subseteq \rightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma)$ *if and only if for each rule $l \rightarrow r$ in R_2, $r_g \in R_1^*(\{l_g\})$.*

Proof. ($\Rightarrow$) Let $l \to r$ be an arbitrary rule in R_2. Clearly, $l_g \to_{R_2} r_g$. Thus by our assumption $l_g \to^*_{R_1} r_g$.

($\Leftarrow$) Let us suppose that $t_1, t_2 \in T_\Sigma$, and that $t_1 \to_{R_2} t_2$ applying the rule $l \to r$. As $r_g \in R_1^*(\{l_g\})$, $l_g \to^*_{R_1} r_g$ holds. Hence $l \to^*_{R_1} r$ implying that $t_1 \to^*_{R_1} t_2$ as well. □

For each rule $l \to r$ in R_2, the tree language $\{l_g\}$ is recognizable, and we can construct a tree automaton over Σ recognizing $\{l_g\}$. As R_1 effectively preserves recognizability, $R_1^*(\{l_g\})$ is also recognizable, and we can construct a tree automaton over Σ recognizing $R_1^*(\{l_g\})$. Hence for each rule $l \to r$ in R_2, we can decide whether or not $r_g \in R_1^*(\{l_g\})$. Thus by Claim 5.15, we can decide if $\to^*_{R_2} \cap (T_\Sigma \times T_\Sigma) \subseteq \to^*_{R_1} \cap (T_\Sigma \times T_\Sigma)$. □

Consequence 5.16. *Let R_1 and R_2 be rewrite systems over Σ effectively preserving recognizability. Moreover, let $g_1, g_2 \in \Sigma - \Sigma_0$ be such that for each $i \in \{1,2\}$, g_i does not occur on the left-hand side of any rule in R_i. Let $\sharp_1, \sharp_2 \in \Sigma_0$ be such that for each $i \in \{1,2\}$, $\sharp_i$ is irreducible for R_i. Then it is decidable which one of the following four mutually excluding conditions holds.*

(i) $\to^*_{R_1} \cap (T_\Sigma \times T_\Sigma) \subset \to^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$,
(ii) $\to^*_{R_2} \cap (T_\Sigma \times T_\Sigma) \subset \to^*_{R_1} \cap (T_\Sigma \times T_\Sigma)$,
(iii) $\to^*_{R_1} \cap (T_\Sigma \times T_\Sigma) = \to^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$,
(iv) $\to^*_{R_1} \cap (T_\Sigma \times T_\Sigma) \bowtie \to^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$.

One can easily show the following result applying Theorem 3.20, Observation 5.10, and Consequence 5.16.

Consequence 5.17. *Let R be an lgsm rewrite system over Σ. Moreover, let $g \in \Sigma - \Sigma_0$ such that g does not occur on the left-hand side of any rule in R, and let $\sharp \in \Sigma_0$ be irreducible for R. Then it is decidable whether R is left-to-right ground minimal.*

Consequence 5.16 also implies the following.

Consequence 5.18. *Let R_1 and R_2 be rewrite systems over Σ such that $R_1 \cup R_1^{-1}$ and $R_2 \cup R_2^{-1}$ are rewrite systems and effectively preserve recognizability. Moreover, let $g_1, g_2 \in \Sigma - \Sigma_0$ be such that for each $i \in \{1,2\}$, g_i does not occur in R_i. Let $\sharp_1, \sharp_2 \in \Sigma_0$ be such that for each $i \in \{1,2\}$, $\sharp_i$ is irreducible for $R_i \cup R_i^{-1}$. Then it is decidable which one of the following four mutually excluding conditions holds.*

(i) $\leftrightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma) \subset \leftrightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$,
(ii) $\leftrightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma) \subset \leftrightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma)$,
(iii) $\leftrightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma) = \leftrightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$,
(iv) $\leftrightarrow^*_{R_1} \cap (T_\Sigma \times T_\Sigma) \bowtie \leftrightarrow^*_{R_2} \cap (T_\Sigma \times T_\Sigma)$.

Theorem 3.20, Observation 5.10, and Consequence 5.18 imply the following.

Consequence 5.19. *Let R be a rewrite system over Σ such that $R \cup R^{-1}$ is an lgsm rewrite system. Moreover, let $g \in \Sigma - \Sigma_0$ be such that g does not occur in any rule of R, and let $\sharp \in \Sigma_0$ be irreducible for $R \cup R^{-1}$. Then it is decidable whether R is two-way ground minimal.*

Lemma 5.20. *Let R be a rewrite system over Σ effectively preserving recognizability, and let $p, q \in T_\Sigma(X)$. Then it is decidable if there exists a tree $r \in T_\Sigma(X)$ such that $p \to_R^* r$ and $q \to_R^* r$.*

Proof. Let $m \geqslant 0$ be such that $var(p) \subseteq X_m$, $var(q) \subseteq X_m$. Let us introduce new constant symbols $Z = \{z_1, \dots, z_m\}$ with $Z \cap \Sigma = \emptyset$. For each $t \in T_\Sigma(X_m)$, let $t_z \in T_{\Sigma \cup Z}$ be defined by $t_z = t[z_1, \dots, z_m]$.

The singleton sets $\{p_z\}$, $\{q_z\}$ are recognizable, and we can construct two tree automata over $\Sigma \cup Z$ which recognize $\{p_z\}$ and $\{q_z\}$, respectively. As R preserves recognizability, $R^*_{\Sigma \cup Z}(\{p_z\})$ and $R^*_{\Sigma \cup Z}(\{q_z\})$ are recognizable, and we can construct two tree automata over $\Sigma \cup Z$ which recognize $R^*_{\Sigma \cup Z}(\{p_z\})$ and $R^*_{\Sigma \cup Z}(\{q_z\})$, respectively. Hence we can decide if $R^*_{\Sigma \cup Z}(\{p_z\}) \cap R^*_{\Sigma \cup Z}(\{q_z\}) = \emptyset$, see [12]. Clearly, $R^*_{\Sigma \cup Z}(\{p_z\}) \cap R^*_{\Sigma \cup Z}(\{q_z\}) \neq \emptyset$ if and only if there exists a tree $r \in T_\Sigma(X)$ such that $p \to_R^* r$ and $q \to_R^* r$. $\square$

Theorem 5.21. *Let R be a rewrite system over Σ effectively preserving recognizability. Then it is decidable if R is locally confluent.*

Proof. By Proposition 2.1, R is locally confluent if and only if for every critical pair (v_1, v_2) of R there exists a tree $v \in T_\Sigma(X)$ such that $v_1 \to_R^* v$ and $v_2 \to_R^* v$. It is well known that all critical pairs of R are variants of finitely many critical pairs of R. Hence it is sufficient to inspect finitely many critical pairs. Thus the theorem follows from Lemma 5.20. $\square$

Theorem 5.22. *Each of the following questions is undecidable for any convergent left-linear gsm rewrite systems R_1 and R_2 over a ranked alphabet Ω, for any recognizable tree language $L \subseteq T_\Omega$ given by a tree automaton over Ω recognizing L, where Γ is the smallest ranked alphabet for which $R_1(L) \subseteq T_\Gamma$.*

(i) *Is $R_1(L) \cap R_2(L)$ empty?*
(ii) *Is $R_1(L) \cap R_2(L)$ infinite?*
(iii) *Is $R_1(L) \cap R_2(L)$ recognizable?*
(iv) *Is $T_\Gamma - R_1(L)$ empty?*
(v) *Is $T_\Gamma - R_1(L)$ infinite?*
(vi) *Is $T_\Gamma - R_1(L)$ recognizable?*
(vii) *Is $R_1(L)$ recognizable?*
(viii) *Is $R_1(L) = R_2(L)$?*
(ix) *Is $R_1(L) \subseteq R_2(L)$?*

Proof. Proposition 2.7 appeared as Theorem 5.2 in [8]. We can apply the proof of Theorem 5.2 in [8] with slight modifications. By Lemma 3.6, the proofs of (i)-(vii) and of (ix) carry over.

To adopt the proof of (viii), we observe the following. Let $\mathscr{A}=(\Sigma,\Delta,A,a_0,R)$ be a deterministic top–down tree transducer. Then by Lemma 3.6, we may assume that $\Sigma\cap\Delta=\emptyset$. Hence, by Lemma 3.6, R is a left-linear gsm rewrite system. Let $\star$ be a new symbol with rank 0, such that $\star\notin\Sigma\cup\Delta\cup A$. If we add a rule $a(x_1)\to\star$ (with $a\in A$) to R, then R remains a left-linear gsm rewrite system. □

A deterministic homomorphism tree transducer is a special deterministic top–down tree transducer, see [9]. It is undecidable for a tree function induced by a deterministic homomorphism tree transducer if it is injective, see [9]. Hence by Proposition 2.5 and Lemma 3.6, the following holds.

Theorem 5.23. *Let R be a convergent left-linear gsm rewrite system over Σ. Let $L\subseteq T_\Sigma$ be a recognizable tree language given by a tree automaton over Σ recognizing L. Then it is undecidable if the tree function $\to_R^*\cap(L\times R(L))$ is injective.*

Lemma 5.24. *Let R and S be linear collapse-free rewrite systems over the disjoint ranked alphabets Σ and Δ, respectively. Let Γ be a ranked alphabet with $\Sigma\cup\Delta\subseteq\Gamma$. Consider R and S as rewrite systems over Γ. Then*

(i) $\to_S\circ\to_R\subseteq\to_R\cup(\to_R\circ\to_S)$, *and*

(ii) $\to_{R\cup S}^*=\to_R^*\circ\to_S^*$.

Proof. The proof of (i) is straightforward. Condition (ii) is a simple consequence of (i). □

Theorem 5.25. *Let R and S be linear collapse-free rewrite systems over the disjoint ranked alphabets Σ and Δ, respectively. Let R and S preserve recognizability. Then $R\oplus S$ also preserves recognizability.*

Proof. Let L be a recognizable tree language over some ranked alphabet Γ, where $\Sigma\cup\Delta\subseteq\Gamma$. By Lemma 5.24, $(R\oplus S)_\Gamma^*(L)=S_\Gamma^*(R_\Gamma^*(L))$. As R preserves recognizability, $R_\Gamma^*(L)$ is recognizable. Moreover, since S preserves recognizability, $S_\Gamma^*(R_\Gamma^*(L))$ is also recognizable. □

Theorem 5.26. *Let R and S be linear collapse-free rewrite systems over the disjoint ranked alphabets Σ and Δ, respectively. Let $R\oplus S$ preserve recognizability. Then R and S also preserve recognizability.*

Proof. Let L be a recognizable tree language over some ranked alphabet Γ, where $\Sigma\subseteq\Gamma$. It is sufficient to show that $R_\Gamma^*(L)$ is recognizable. Without loss of generality we may rename the symbols of Γ such that $\Gamma\cap\Delta=\emptyset$. Thus $R_\Gamma^*(L)=(R\oplus S)_{\Gamma\cup\Delta}^*(L)$. Since $\Sigma\cup\Delta\subseteq\Gamma\cup\Delta$ and $R\oplus S$ preserves recognizability, $R_\Gamma^*(L)$ is recognizable. □

Consequence 5.27. *For linear collapse-free rewrite systems, the property of preserving recognizability is modular.*

The proof of the following result is similar to the proof of Consequence 5.27.

Theorem 5.28. *For linear collapse-free rewrite systems, the property of effectively preserving recognizability is modular.*

Let R and S be string rewrite systems over the disjoint alphabets Σ and Δ, respectively. Then the disjoint union $R \oplus S$ of R and S is the string rewrite system $R \cup S$ over the alphabet $\Sigma \cup \Delta$. A property $\mathscr{P}$ is modular if $R \oplus S$ has the property $\mathscr{P}$ if and only if both R and S has the property $\mathscr{P}$. Our results on linear collapse-free rewrite systems imply that preserving recognizability and effectively preserving recognizability are modular properties of λ-free string rewrite systems.

Theorem 5.29. *Let R and S be λ-free string rewrite systems over the disjoint alphabets Σ and Δ, respectively. Then R and S preserve recognizability if and only if $R \oplus S$ also preserves recognizability. Moreover, R and S effectively preserve recognizability if and only if $R \oplus S$ also effectively preserves recognizability.*

6. Conclusion and open problems

We have introduced the notion of the generalized semi-monadic rewrite system, which is a generalization of well-known rewrite systems: the ground rewrite system, the monadic rewrite system, and the semi-monadic rewrite system. We have shown that lgsm rewrite systems effectively preserve recognizability. We have shown that a tree language L is recognizable if and only if there exists a rewrite system R such that $R \cup R^{-1}$ is an lgsm rewrite system and that L is the union of finitely many $\leftrightarrow_R^*$-classes. We presented several decidability and undecidability results on gsm rewrite systems. Our results give rise to several open problems.

(1) Gilleron [14] showed that for a rewrite system R, it is not decidable if R preserves $sign(R)$-recognizability. Is it decidable for a rewrite system R over a ranked alphabet Σ, $sign(R) \subset \Sigma$, whether R preserves Σ-recognizability? Is it decidable for a rewrite system R whether R preserves recognizability?

(2) Generalize lgsm rewrite systems such that the obtained rewrite systems still effectively preserve recognizability.

(3) Let R_1 and R_2 be rewrite systems effectively preserving recognizability (lgsm rewrite systems, respectively) over Σ. Is it decidable if $\rightarrow_{R_1}^* \cap (T_\Sigma \times T_\Sigma) \subseteq \rightarrow_{R_2}^* \cap (T_\Sigma \times T_\Sigma)$? Is it decidable if $\leftrightarrow_{R_1}^* \cap (T_\Sigma \times T_\Sigma) \subseteq \leftrightarrow_{R_2}^* \cap (T_\Sigma \times T_\Sigma)$?

(4) Let R be a rewrite system effectively preserving recognizability. Is it decidable if R is left-to-right minimal? Is it decidable if R is two-way minimal? Is it decidable if R is left-to-right ground minimal? Is it decidable if R is two-way ground minimal? The last two questions are also open if R is an lgsm rewrite system.

(5) Dauchet and his colleagues [5, 6] have shown that for a ground rewrite system R, it is decidable if R is confluent and it is decidable if R is noetherian. Give subclasses $\mathbf{C}_1$ and $\mathbf{C}_2$ of lgsm rewrite systems which contain the class of ground rewrite systems such that for any rewrite system $R \in \mathbf{C}_1$ it is decidable if R is noetherian and that for any rewrite system $R \in \mathbf{C}_2$, it is decidable if R is confluent.

(6) Fülöp and Gyenizse [9] showed that for an arbitrary linear deterministic top–down tree transducer $\mathscr{A}$, it is decidable if the tree function $\tau_{\mathscr{A}}$ is injective. Hence in the light of Theorem 5.23, we raise the following question. Let R be a convergent lgsm rewrite system over Σ. Let $\Delta \subset \Sigma$ and $\Gamma \subset \Sigma$. Is it decidable if the function $\rightarrow_R^* \cap (T_\Delta \times T_\Gamma)$ is injective?

(7) A rewrite system R over Σ is tame if for all critical pairs (u, v) of R

(i) $R^*(\{u\}) \cup R^*(\{v\})$ is finite,

(ii) for each $w \in R^*(\{u\}) \cup R^*(\{v\})$, $w \rightarrow_R^+ w$ does not hold, and

(iii) for any $u' \in R^*(\{u\})$ and $v' \in R^*(\{v\})$, there is a $z \in T_\Sigma(X)$ such that $u' \rightarrow_R^* z$ and $v' \rightarrow_R^* z$.

If R effectively preserves recognizability, then it is decidable if R is tame. If R is convergent, then R is tame as well. It would be worth while studying tame rewrite systems preserving recognizability.

Recently, Otto [19] has proved the following result which appeared as a conjecture in a previous version of this paper.

Theorem 6.1 (Otto [19]). *A string rewrite system R over Σ preserves Σ-recognizability if and only if R preserves recognizability.*

References

[1] R.V. Book and F. Otto, *String-Rewriting Systems* (Springer, New York, 1993).

[2] W.S. Brainerd, Tree generating regular systems, *Inform and Control* **14** (1969) 217–231.

[3] A.C. Caron and J.L. Coquidé, Decidability of reachability for disjoint union of term rewriting systems, *Theoret. Comput. Sci.* **126** (1994) 31–52.

[4] J.L. Coquidé, M. Dauchet, R. Gilleron and S. Vágvölgyi, Bottom–up tree pushdown automata: classification and connection with rewrite systems, *Theoret. Comput. Sci.* **127** (1994) 69–98.

[5] M. Dauchet, P. Heuillard, P. Lescanne and S. Tison, Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems, *Inform. and Control* **88** (1990) 187–201.

[6] M. Dauchet and S. Tison, The theory of ground rewrite systems is decidable, in: *Proc. 5th IEEE Symp. on Logic in Computer Science* (Philadelphia, 1990) 242–248.

[7] N. Dershowitz and J.P. Jouannaud, Rewrite systems, in: J.V. Leeuwen, ed., *Handbook of Theoretical Computer Science*, Vol B (North Holland, Amsterdam, 1990) 243–320.

[8] Z. Fülöp, Undecidable properties of deterministic top–down tree transducers, *Theoret. Comput. Sci.* **134** (1994) 311–328.

[9] Z. Fülöp and P. Gyenizse, On injectivity of deterministic top–down tree transducers, *Inform. Process. Lett.* **48** (1993) 183–188.

[10] Z. Fülöp and S. Vágvölgyi, Congruential tree languages are the same as recognizable tree languages – A proof for a theorem of D. Kozen, *Bull. EATCS* **39** (1989) 175–185.

[11] J.H. Gallier and R.V. Book, Reductions in tree replacement systems, *Theoret. Comput. Sci.* **37** (1985) 123–150.

[12] F. Gécseg and M. Steinby, *Tree Automata* (Akadémiai Kiadó, Budapest, 1984).

[13] F. Gécseg and M. Steinby, Tree Languages, in: G. Rozenberg and A. Salomaa eds., *Handbook of Formal Language Theory* (Springer, Berlin), to be published.
[14] R. Gilleron, Decision problems for term rewriting systems and recognizable tree languages, *Proc. STACS'91, Lecture Notes in Computer Science*, Vol. 480 (Springer, Berlin, 1991) 148–159.
[15] R. Gilleron and S. Tison, Regular tree languages and rewrite systems, *Fund. Inform.* **24** (1995) 157–175.
[16] G. Huet, Confluent Reductions: Abstract properties and applications to term rewriting systems, *J. ACM* **27** (1980) 797–821.
[17] D. Kozen, Complexity of finitely presented algebras, in: *Proc. 9th Annual ACM Symp. on Theory of Computing* (Boulder, CO, 1977) 164–177.
[18] J.W. Lloyd, *Foundations of Logic Programming* (Springer, Berlin, extended edition, 1993).
[19] F. Otto, On recognizability preserving string-rewriting systems, Manuscript, 1995.
[20] K. Salomaa, Deterministic tree pushdown automata and monadic tree rewriting systems, *J. Comput. System Sci.* **37** (1988) 367–394.