Acta Cybernetica **22** (2016) 633–655.

One-Pass Reductions[*]

Sándor Vágvölgyi[†]

Abstract

We study OI and IO one-pass reduction sequences with term rewrite systems. We present second order decidability and undecidability results on recognizable tree languages and one-pass reductions. For left-linear TRSs, the second order OI inclusion problem and the second order OI reachability problem are decidable, the second order OI joinability problem is undecidable. For right-linear TRSs, the second order common IO ancestor problem is undecidable.

Keywords: term rewrite systems, OI and IO one-pass reductions, tree automata

1 Introduction

A term rewrite system (TRS for short) R reduces a term in a nondeterministic way along which it does many choices. Traditional term rewriting is the exhaustive application of R to a term until no more rules apply. However, this procedure is usually not adequate for most applications, for example for program transformation. Because, in general, there is no bound for the lengths of the possible reduction sequences, or R is not confluent.

To overcome the above problems, researchers implemented and studied various types of traversal reductions. Program transformation operates on the syntax tree of a program applying rewrite rules along a traversal of the tree: it visits all tree nodes in a certain visiting order and applies a rewrite rule at each node at most once [2, 3, 14, 15]. They distinguish between the standard visiting orders top-down (order: root, subtrees) and bottom-up (order: subtrees, root). Dauchet and De Comité [4], Seynhaeve et al. [12] studied outside-in (OI) and inside-out (IO) one-pass reductions, which are different from the above top-down and bottom-up traversals, respectively, in that all reduction steps can be carried out mainly independently from each other. During a reduction step, the left-hand side of an

[*]The publication is supported by the European Union and co-funded by the European Social Fund. Project title: “Telemedicine-focused research activities on the field of Mathematics, Informatics and Medical sciences” Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073

[†]Department of Foundations of Computer Science, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary E-mail: vagvolgy@inf.u-szeged.hu

DOI: 10.14232/actacyb.22.3.2016.6

applied rule does not overlap with the already rewritten parts of the term, only the values of the substituted subterms depend on the order of the reduction steps. One may proceed in two ways. Along an OI one-pass reduction sequence, we proceed from the root to the leaves. Along an IO one-pass reduction sequence, we proceed from the leaves to the root. Fülöp et al. [5] studied two other very restrictive strategies of term rewriting: and one-pass root-started rewriting and one-pass leaf-started rewriting. They differ from the OI and IO one-pass reductions, respectively, in that the rewriting always concern positions immediately adjacent to the already rewritten parts of the term. Consequently, they establish a much more restricted way of computing.

Reachability is a fundamental problem that appears in several areas of computer science: finite- and infinite-state concurrent systems, computational models like cellular automata and Petri nets, program analysis, discrete and continuous systems, time critical systems, hybrid systems, TRSs, etc. [13]. For TRSs, reachability problem is the following: given a TRS R, and two terms s and t, decide whether s can be rewritten into t with a finite number of rewriting steps of R. Ground reachability problem is the restriction of the reachability problem to ground terms. Ground reachability and unreachability proofs can be used as general purpose verification techniques for the systems modeled by rewriting [9].

Gilleron and Tison [10] introduced and studied the second order reachability problem and the second order sentential form inclusion problem for TRSs. They [10] asked whether the set of sentential forms of the trees of a recognizable tree language overlaps with a given recognizable tree language, and whether the set of sentential forms of the trees of a recognizable tree language is a subset of a given recognizable tree language, respectively. Observe that they [10] defined a second order decidability problem from a first order one by substituting recognizable tree languages for terms. Along this line of research, Fülöp et al. [5] introduced and studied second order decidability problems: the one-pass root-started sentential form inclusion problem, and the one pass leaf-started sentential form inclusion problem. Moreover, Seynhaeve et al. [12] presented and studied the second order IO inclusion problem.

In the light of the above problems, we study the following eight second order decidability problems. Some of them, for instance the second order OI common ancestor problem, are introduced in this paper following the above research line. The terms appearing in an OI (resp. IO) one pass reduction sequence are called the OI (resp. IO) sentential forms of the initial term. For a tree language L, the set of all OI (resp. IO) sentential forms of the elements of L is denoted by $SFOI(L)$ (resp. $SFIO(L)$). First we present the problems concerning OI one pass reducing.

Second order OI inclusion problem.
Instance: A TRS R and recognizable tree languages L and M over Σ.
Question: Is $SFOI(L) \subseteq M$?

Second order OI reachability problem.
Instance: A TRS R and recognizable tree languages L and M over Σ.
Question: Is $SFOI(L) \cap M \neq \emptyset$?

Second order OI joinability problem.
Instance: A TRS R and recognizable tree languages L and M over Σ.
Question: Is $SFOI(L) \cap SFOI(M) \neq \emptyset$?

Second order OI common ancestor problem.
Instance: A TRS R and recognizable tree languages L and M over Σ.
Question: Is there a term $t \in T_\Sigma(X)$ such that $SFOI(t) \cap L \neq \emptyset$ and $SFOI(t) \cap M \neq \emptyset$?

We define the IO counterparts of the above problems replacing OI by IO.

Seynhaeve et al. showed that for left-linear TRSs and right-linear TRSs, the second order IO inclusion problem is decidable, see Proposition 4 in [12]. Fülöp et al. [5] showed that for left-linear TRSs, the one-pass root-started sentential form inclusion problem, the counterpart of the second order IO inclusion problem, and the one pass leaf-started sentential form inclusion problem, the counterpart of the second order OI inclusion problem, are decidable. Seynhaeve et al. showed that for right-linear TRSs, the one-pass root-started sentential form inclusion problem is decidable, see Proposition 5 in [12].

In Section 2, we present our notations and basic definitions. Then we show the following. For left-linear TRSs, the second order OI inclusion problem and the second order OI reachability problem are decidable, see Section 3. For left-linear TRSs, the second order OI joinability problem is undecidable, see Section 4. For right-linear TRSs, the second order common IO ancestor problem is undecidable, see Section 5. In Section 6, we present our concluding remarks and open problems. We sum up the existing results in the literature and our contribution in Table 1, where OI, IO, ll, rl, and ances. abbreviate one-pass OI reduction, one-pass IO reduction, left-linear, right-linear, and ancestor, respectively. Each question mark signifies an open problem.

decidability of	inclusion	reachability	joinability	common ances.
OI	decidable for ll Theorem 1	decidable for ll Theorem 1	undecidable for ll Theorem 2	?
IO	decidable for ll and for rl [12], Prop. 4	?	?	undecidable for rl Theorem 3

Table 1: Summary of results

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 term rewrite systems and of tree language theory [1, 7, 8].

2.1 Terms

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. For a word $\alpha \in N^*$, $length(\alpha)$ stands for the length of α. Consider the words $\alpha, \beta, \gamma \in N^*$ such that $\alpha = \beta\gamma$. Then we say that β is a prefix of α, and that α is an extension of β; and we write $\beta \preceq \alpha$. If $\gamma \neq \lambda$, then β is a proper prefix of α, and we write $\beta \prec \alpha$.

A ranked alphabet is a finite set Σ in which every symbol has a unique rank in N. For $m \geq 0$, Σ_m denotes the set of all elements of Σ which have rank m. The elements of Σ_0 are called constants. Throughout the paper we assume that $\Sigma_0 \neq \emptyset$. That is, we have at least one constant in Σ.

For a set of variables Y and a ranked alphabet Σ, $T_\Sigma(Y)$ denotes the set of Σ-terms (or Σ-trees) over Y. $T_\Sigma(\emptyset)$ is written as T_Σ. A term $t \in T_\Sigma$ is called a ground term. 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 $X = \{ x_1, x_2, \dots \}$ of variables which will be kept fixed in this paper. Moreover, we put $X_m = \{ x_1, \dots, x_m \}$, for $m \geq 0$. Hence $X_0 = \emptyset$.

For a term $t \in T_\Sigma(X)$, the height $height(t)$ and the yield $yd(t)$ and the set of positions $pos(t) \subseteq N^*$ of t are defined by tree induction.

- If $t \in \Sigma_0 \cup X$, then $height(t) = 0$, $yd(t) =$(if $t \in \Sigma_0$ then λ else t), and $pos(t) = \{ \lambda \}$.
- If $t = f(t_1, \dots, t_m)$ with $f \in \Sigma_m,\ m > 0$, then
  $height(t) = 1 + \max\{height(t_i) \mid 1 \leq i \leq m\}$,
  $yd(t) = yd(t_1) \dots yd(t_m)$, and
  $pos(t) = \{ i\alpha \mid 1 \leq i \leq m, \alpha \in pos(t_i) \}$.

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

- for $t \in \Sigma_0 \cup X$, $t/\lambda = t$ and $lab(t, \lambda) = t$;
- for $t = f(t_1, \dots, t_m)$ with $m \geq 1$ and $f \in \Sigma_m$, if $\alpha = \lambda$ then $t/\alpha = t$ and $lab(t, \alpha) = f$, otherwise, if $\alpha = i\beta$ with $1 \leq i \leq m$, then $t/\alpha = t_i/\beta$ and $lab(t, \alpha) = lab(t_i, \beta)$.

Let $t \in T_\Sigma(X)$. We call a position $\alpha \in pos(t)$ of t a variable position if $lab(t, \alpha) \in X$. The set of variable positions of t is denoted by $vpos(t)$. That is, $vpos(t) = \{ \alpha \in pos(t) \mid lab(t, \alpha) \in X \}$. Furthermore, $root(t) = lab(t, \lambda)$.

For trees $t \in T_\Sigma(X_m)$, and $t_1, \dots, t_m \in T_\Sigma(X)$, we denote by $t[t_1, \dots, t_m]$ the tree obtained by substituting t_i for every occurrence of x_i in t, for $1 \leq i \leq m$. A tree language L is a subset of T_Σ.

For any $m \geq 0$, we distinguish a subset $\overline{T}_\Sigma(X_m)$ of $T_\Sigma(X_m)$ as follows: a tree $t \in T_\Sigma(X_m)$ is in $\overline{T}_\Sigma(X_m)$ if and only if $yd(t) = x_1 \dots x_m$.

For each integer $k \geq 0$, we say that a tree $t \in T_\Sigma(X)$ is a k-normal tree over Σ if t satisfies Conditions 1 and 2 [6].

1. $t \in \overline{T}_\Sigma(X_m)$ for some $m \geq 0$.
2. For every $\alpha \in pos(t)$, ($length(\alpha) = k$ and $lab(t, \alpha) \in X_m$) or ($length(\alpha) < k$ and $lab(t, \alpha) \in \Sigma$).

Note that for each k-normal tree t, $height(t) \leq k$ and that the only 0-normal tree is x_1. The set of k-normal trees over Σ is denoted by $NORM_{\Sigma,k}$.

We illustrate our concepts and results via a running example which we present as a sequence of examples throughout Sections 2 and 3. So the ranked alphabet in all examples will be the one introduced below.

Example 1. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$, where $\Sigma_0 = \{\#\}$, $\Sigma_1 = \{f\}$, and $\Sigma_2 = \{g\}$. The following trees are 3-normal trees over Σ: $\#$, $f(\#)$, $g(\#, g(\#, \#))$, $g(g(\#, g(x_1, x_2)), \#)$, $g(\#, g(\#, f(x_1)))$, $g(f(g(x_1, x_2)), g(f(x_3), \#))$.

By the definition of a k-normal tree we have the following.

Remark 1. For each $k \geq 0$, $\{ t \in NORM_{\Sigma,k} \mid height(t) < k \} \subseteq T_\Sigma$.
For all $0 \leq i \leq j \leq k$,
$\{ t \in NORM_{\Sigma,k} \mid height(t) \leq i \} \subseteq \{ t \in NORM_{\Sigma,k} \mid height(t) \leq j \}$.

Let Σ be a ranked alphabet, $u \in NORM_{\Sigma,k} \cap \overline{T}_\Sigma(X_m)$, $k, m \geq 0$, and $v \in T_\Sigma$. We say that u is a k-normal prefix of v if $v = u[u_1, \ldots, u_m]$ for some $u_1, \ldots, u_m \in T_\Sigma$.

Proposition 1. [6] *For each tree $s \in T_\Sigma$ and $k \geq 0$, s has exactly one k-normal prefix.*

Remark 2. Let $k \geq 1$, $s, t \in T_\Sigma$, and assume that for any $\alpha \in pos(s)$, if $length(\alpha) \leq k - 1$ then $\alpha \in pos(t)$ and $lab(s, \alpha) = lab(t, \alpha)$. Then the k-normal prefix of s is equal to the k-normal prefix of t.

Example 2. Let $s = g(f(\#), f(g(\#, \#)))$. Then
$g(x_1, x_2)$ is the 1-normal prefix of s,
$g(f(x_1), f(x_2))$ is the 2-normal prefix of s,
$g(f(\#), f(g(x_1, x_2))$ is the 3-normal prefix of s, and
s is the k-normal prefix of s for $k \geq 4$.

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

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

An alphabet Δ is any finite nonempty set, Δ^* stands for the set of words over Δ, and λ denotes the empty word. For an alphabet Δ, we consider the ranked alphabet $\Delta \cup \{\#\}$, where $\# \notin \Delta$. Here each element of Δ is a unary symbol and $\#$ is a nullary symbol. Then we consider a tree in $T_{\Delta \cup \{\#\}}$ as a word over the alphabet $\Delta \cup \#$. For example, let $\Delta = \{a, b\}$. Then the tree $a(b(b(a(\#))))$ is written as the word $abba\#$. Conversely, for each word $w \in \Delta^*$, the word $w\#$ over the alphabet $\Delta \cup \{\#\}$ can be considered as a tree over the ranked alphabet $\Delta \cup \{\#\}$. For example, the word $aab\#$ can be considered as the tree $a(a(b(\#)))$.

2.2 Term Rewrite Systems

Let $\rightarrow \subseteq A \times A$ be a binary relation on a set A. We denote by $\rightarrow^*$ the reflexive, transitive closure of $\rightarrow$.

Let Σ be a ranked alphabet. Then a term rewrite system (TRS) R over Σ is a finite subset of $(T_\Sigma(X) - X) \times T_\Sigma(X)$ 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 $l \rightarrow r$. We call l the left-hand side and r the right-hand of the rule $l \rightarrow r$. The set of left-hand sides (resp. right-hand sides) of rules in R is denoted by $lhs(R)$ (resp. $rhs(R)$).

A TRS R is left-linear (resp. right-linear) if each element of $lhs(R)$ (resp. $rhs(R)$) is linear. A left-linear and right-linear TRS R is called linear. A TRS R is ground if each element of $lhs(R) \cup rhs(R)$ is a ground term.

Let R be a TRS over Σ. For any terms $s, t \in T_\Sigma(X)$, position $\alpha \in pos(s)$, and rule $l \rightarrow r$ in R with $l, r \in T_\Sigma(X_m)$, $m \geq 0$, we say that s rewrites to t applying the rule $l \rightarrow r$ at α, and denote this by $s \rightarrow_{\alpha, l \rightarrow r} t$ if there are $s_1, \ldots, s_m \in T_\Sigma(X)$ such that $s/\alpha = l[s_1, \ldots, s_m]$ and $t = s[\alpha \leftarrow r[s_1, \ldots, s_m]]$. Here we also say that s rewrites to t and denote this by $s \rightarrow_R t$.

A sequence

$$s_0 \rightarrow_{\beta_1, l_1 \rightarrow r_1} s_1 \rightarrow_{\beta_2, l_2 \rightarrow r_2} s_2 \rightarrow_{\beta_3, l_3 \rightarrow r_3} \cdots \rightarrow_{\beta_n, l_n \rightarrow r_n} s_n,\ n \geq 0 \tag{1}$$

is called a reduction sequence with R.

Dauchet and De Comité [4] introduced the inside-out and outside-in one-pass reductions with a TRS R in an intuitive way and illustrated these concepts by examples. Intuitively, for any terms $s, t \in T_\Sigma(X)$, we say that s is rewritten to t in one pass if we rewrite s into t applying some rules such that the left-hand sides do not overlap. Moreover, in case of an OI pass, we rewrite from the outermost of a bracketed expression of a term to the innermost, i.e., in a top-down order, hence the subtrees are rewritten after duplicating subtrees. In case of an IO pass, we rewrite from the innermost of a bracketed expression of a term to the outermost, i.e., in a bottom-up order, hence the subtrees are rewritten before duplicating subtrees.

We now formally define these concepts. An outside-in one-pass (OI) reduction sequence with R is a sequence

$$s_0 \longrightarrow_{\alpha_1, \beta_1, l_1 \rightarrow r_1} s_1 \longrightarrow_{\alpha_2, \beta_2, l_2 \rightarrow r_2} s_2 \longrightarrow_{\alpha_3, \beta_3, l_3 \rightarrow r_3} \cdots \longrightarrow_{\alpha_n, \beta_n, l_n \rightarrow r_n} s_n, \tag{2}$$

where Conditions 1–4 hold.

1. $n \geq 0$, $s_0, \ldots, s_n \in T_\Sigma(X)$, and $\alpha_i \in pos(s_0)$, $\beta_i \in pos(s_{i-1})$ for $i = 1, \ldots, n$.

2. $s_0 \rightarrow_{\beta_1, l_1 \rightarrow r_1} s_1 \rightarrow_{\beta_2, l_2 \rightarrow r_2} s_2 \rightarrow_{\beta_3, l_3 \rightarrow r_3} \cdots \rightarrow_{\beta_n, l_n \rightarrow r_n} s_n$ is a reduction sequence with R.

3. $\alpha_1 = \beta_1$.

4. For any $1 < j \leq n$, if there is $1 \leq i < j$ such that $\beta_i \preceq \beta_j$, then let k be the largest such i, and then

a) $\alpha_k\gamma\xi = \alpha_j$, for some $\gamma \in vpos(l_k)$ and $\xi \in N^*$,

b) $\beta_k\delta\xi = \beta_j$ for some $\delta \in vpos(r_k)$, and

c) $lab(l_k, \gamma) = lab(r_k, \delta) \in X$.

Informally, along (2), we rewrite in a top-down order, and keep on rewriting the unprocessed subtrees of the initial tree s_0. At the same time we keep track of the positions of the subtrees of the initial tree s_0. For each $i = 1, \dots, n$, the position α_i points to the subtree s_0/α_i of s_0, and the position β_i points to an occurrence of s_0/α_i in s_{i-1}, to be rewritten in the ith step of (2).

Formally, the terms $s_0, \dots, s_n$ are called OI sentential forms of s_0. For each term $s \in T_\Sigma(X)$, $SFOI(s)$ is the set of all OI sentential forms of s. That is, $SFOI(s)$ is the set of all terms t such that there is an OI reduction sequence (2) with $s = s_0$ and $t = s_n$. For a tree language $L \subseteq T_\Sigma$, we put

$$SFOI(L) = \bigcup (SFOI(s) \mid s \in L).$$

We usually write (2) in the form

$$s_0 \underset{R}{\to} s_1 \underset{R}{\to} \cdots \underset{R}{\to} s_n, \tag{3}$$

and note that (3) is an OI reduction. The notation

$$s_0 \underset{R,OI}{\Rightarrow} s_n$$

means that there is an OI reduction (3).

Example 3. Consider the left-linear TRS

$$R = \{\, g(x_1, x_2) \to f(x_2),\ g(x_1, x_2) \to g(x_1, x_1) \,\}$$

over Σ. Then
$g(g(g(\#,\#),\#), g(\#,\#)) \longrightarrow_{\lambda,\lambda,g(x_1,x_2)\to g(x_1,x_1)}$
$g(g(g(\#,\#),\#), g(g(\#,\#),\#)) \longrightarrow_{1,1,g(x_1,x_2)\to f(x_2)}$
$g(f(\#), g(g(\#,\#),\#)) \longrightarrow_{11,21,g(x_1,x_2)\to f(x_2)} g(f(\#), g(f(\#),\#))$
is an OI reduction sequence with R.

Furthermore,
$SFOI(g(g(\#,\#),\#)) =$
$\{\, g(g(\#,\#),\#), f(\#), g(g(\#,\#), g(\#,\#)),$
$g(f(\#),\#), g(f(\#), g(\#,\#)), g(g(\#,\#), f(\#)), g(f(\#), f(\#)) \,\}$.

An inside-out one-pass (IO) reduction sequence with R is a reduction sequence (1) where Conditions 1 and 2 hold.

1. $s_0, \dots, s_n \in T_\Sigma(X)$, and $\beta_i \in pos(s_{i-1}) \cap pos(s_0)$ for $i = 1, \dots, n$,

2. For any $2 \le j \le n$, $\{\, \beta_1, \dots, \beta_{j-1} \}\cap$
$(\{\, \gamma \in N^* \mid \gamma \preceq \beta_j \,\} \cup \{\, \beta_j\xi \mid \xi \in (pos(l_j) - vpos(l_j)) \,\}) = \emptyset$.

Informally, Condition 2 ensures that we rewrite in a bottom-up order and that the left-hand sides do not overlap. It says that for each $2 \leq j \leq n$, the positions $\beta_1, \ldots, \beta_{j-1}$ are no prefixes of β_j nor are positions of any nonvariable symbol in the occurrence of the left-hand side l_j of the rule $l_j \to r_j$ when applying it at β_j in the jth step.

The terms $s_0, \ldots, s_n$ are called IO sentential forms of s_0. For each term $s \in T_\Sigma(X)$, $SFIO(s)$ denotes the set of all IO sentential forms of s. For a tree language $L \subseteq T_\Sigma$, let $SFIO(L) = \bigcup(SFIO(s) \mid s \in L)$. We usually write (1) in the form (3) and note that (3) is an IO reduction. The notation $s_0 \Rightarrow_{R,IO} s_n$ means that there is an IO reduction (1).

Example 4. $g(g(g(\#,\#),\#),g(\#,\#)) \to_{11,g(x_1,x_2)\to f(x_2)}$
$g(g(f(\#),\#),g(\#,\#)) \to_{1,g(x_1,x_2)\to f(x_2)} g(f(\#),g(\#,\#))$
is an IO reduction sequence with R. Furthermore,
$g(\#,g(f(\#),\#)) \to_{2,g(x_1,x_2)\to f(x_2)} g(\#,f(\#)) \to_{\lambda,g(x_1,x_2)\to g(x_1,x_1)} g(\#,\#)$
is another IO reduction sequence with R.

Let R be a TRS over Σ, and $s,t \in T_\Sigma(X)$ be arbitrary. We say that s and t are OI joinable for R if $SFOI(s) \cap SFOI(t) \neq \emptyset$. Furthermore, we say that s is an OI ancestor of t with respect to R if $s \Rightarrow_{R,OI} t$. For tree languages L and M over Σ, we say that L and M are OI joinable for R if $SFOI(L) \cap SFOI(M) \neq \emptyset$. For tree languages L and M over Σ, we say that L and M have a common OI ancestor with respect to R if there is a term $t \in T_\Sigma(X)$ such that $SFOI(t) \cap L \neq \emptyset$ and $SFOI(t) \cap M \neq \emptyset$.

We define the IO counterparts of the above definitions replacing OI by IO.

For the definition of the second order OI (resp. IO) inclusion problem, the second order OI (resp. IO) reachability problem, the second order OI (resp. IO) joinability problem, and second order OI (resp. IO) common ancestor problem, see the Introduction.

2.3 Post Correspondence Problem

A Post Correspondence System (PCS for short) over an alphabet Δ is a pair $\langle \mathbf{w}, \mathbf{z} \rangle = \langle (w_1, \ldots, w_n), (z_1, \ldots, z_n) \rangle$, $n \geq 1$, of lists of nonempty words over the alphabet Δ. We say that the index sequence $k_1, \ldots, k_\ell$ with $\ell \geq 1$, $1 \leq k_1, \ldots, k_\ell \leq n$, is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$, if

$$w_{k_1} \ldots w_{k_\ell} = z_{k_1} \ldots z_{k_\ell} .$$

cf. [11]. The Post Correspondence Problem is the question whether or not a given PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ has a solution.

Proposition 2. [11] *The Post Correspondence Problem is unsolvable. That is, there is no algorithm which takes a PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ as input and determines whether or not there is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$.*

2.4 Recognizable Tree Languages

Let Σ be a ranked alphabet, a bottom-up tree automaton (bta) over Σ is a quadruple $\mathcal{A} = (\Sigma, A, R, A_f)$, where A is a finite set of states of rank 0, $\Sigma \cap A = \emptyset$, $A_f (\subseteq A)$ is the set of final states, R is a finite set of rules of the form

$f(a_1, \ldots, a_m) \to a$ with $m \geq 0$, $f \in \Sigma_m$, $a_1, \ldots, a_m, a \in A$.

We call $f(a_1, \ldots, a_m)$ the left-hand side of the rule $f(a_1, \ldots, a_m) \to a$. We consider R as a ground TRS over $\Sigma \cup A$. The tree language recognized by $\mathcal{A}$ is $L(\mathcal{A}) = \{\, t \in T_\Sigma \mid (\exists a \in A_f)\ t \to_R^* a \}$. We say that a tree language L is recognizable if there exists a bta $\mathcal{A}$ such that $L(\mathcal{A}) = L$ [7]. The bta $\mathcal{A} = (\Sigma, A, R, A_f)$ is total if for all $f \in \Sigma_m$, $m \geq 0$, and $a_1, \ldots, a_m$, R has a rule with the left-hand side $f(a_1, \ldots, a_m)$. The bta $\mathcal{A} = (\Sigma, A, R, A_f)$ is deterministic (dbta) if R has no two rules with the same left-hand side. We give a recognizable tree language L via a total dbta $\mathcal{A}$ recognizing L. Let $\mathcal{A}$ be a total dbta. Then for each tree $t \in T_\Sigma$, there is exactly one state $a \in A$ such that $t \to_{\mathcal{A}}^* a$. We denote this a by $t^{\mathcal{A}}$.

Proposition 3. *Let $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ be an arbitrary total dbta. Let $s, u \in T_\Sigma$ and $\alpha \in pos(s)$. If $(s/\alpha)^{\mathcal{A}} = u^{\mathcal{A}}$, then $s^{\mathcal{A}} = (s[\alpha \leftarrow u])^{\mathcal{A}}$.*

In the rest of the paper we write $s[\alpha \leftarrow u]^{\mathcal{A}}$ for $(s[\alpha \leftarrow u])^{\mathcal{A}}$.

3 OI One Pass Reductions

We show that for left-linear TRSs, the second order OI inclusion problem and the second order OI reachability problem are decidable.

Theorem 1. *For any left-linear TRS R and recognizable tree languages L and M over Σ, it is decidable whether $SFOI(L) \subseteq M$ and whether $SFOI(L) \cap M \neq \emptyset$.*

The proof needs a long preparation, we now start the process of getting ready for it. Let the total dbtas $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$ be such that $L(\mathcal{A}) = L$ and $L(\mathcal{B}) = M$. We introduce the ranked alphabet $\Delta = \Sigma \times A \times \mathcal{P}(B)$. We write the elements of Δ in the form $\langle f, a, C\rangle$ and the rank of a symbol $\langle f, a, C\rangle$ in Δ is the rank of f in Σ. To every $s \in T_\Sigma$, we associate an element of Δ, denoted by $val(s)$ as follows: let $val(s) = \langle f, a, C\rangle$, where $root(s) = f$, $s^{\mathcal{A}} = a$, and

$$C = \{\, b \mid b = t^{\mathcal{B}} \text{ for some } t \in SFOI(s) \,\}.$$

Let $s \in T_\Sigma$ be arbitrary. The evaluated copy of s, denoted by $ec(s)$, is a term in T_Δ defined in the following way.

- $pos(ec(s)) = pos(s)$,
- for each $\alpha \in pos(s)$, $lab(ec(s), \alpha) = val(s/\alpha)$.

For each $k \geq 0$, the evaluated k-prefix of s is the k-normal prefix of $ec(s)$, and is denoted by $ep_k(s)$.

Remark 3. For any $s \in T_\Sigma$ and $k \geq 0$, $ep_k(s) \in NORM_{\Delta,k}$.

Example 5. Let $L(\mathcal{A}) = L$, where $\mathcal{A} = (\Sigma, A, R_\mathcal{A}, A_f)$, $A = \{ 0, 1 \}$, $A_f = \{ 0 \}$, and $R_\mathcal{A}$ consists of the rules

$\# \to 1$,
$f(0) \to 0$, $f(1) \to 1$,
$g(0,0) \to 0$, $g(0,1) \to 1$, $g(1,0) \to 1$, $g(1,1) \to 0$.

Let $L(\mathcal{B}) = M$, where $\mathcal{B} = (\Sigma, B, R_\mathcal{B}, B_f)$, $B = \{ 0, 1, 2 \}$, $B_f = \{ 0 \}$, and $R_\mathcal{B}$ consists of the rules

$\# \to 1$,
$f(0) \to 0$, $f(1) \to 1$, $f(2) \to 2$,
$g(0,0) \to 0$, $g(0,1) \to 1$, $g(0,2) \to 2$,
$g(1,0) \to 1$, $g(1,1) \to 2$, $g(1,2) \to 0$,
$g(2,0) \to 2$, $g(2,1) \to 0$, $g(2,2) \to 1$.

Then

$val(g(\#,\#)) = \langle g, 0, \{ 1, 2 \} \rangle$,
$val(g(g(\#,\#),\#)) = \langle g, 1, \{ 0, 1, 2 \} \rangle$,
$ep_1(g(g(\#,\#),\#)) = \langle g, 1, \{ 0, 1, 2 \} \rangle(x_1, x_2)$,
$ep_2(g(g(\#,\#),\#)) =$
$\langle g, 1, \{ 0, 1, 2 \} \rangle(\langle g, 0, \{ 1, 2 \} \rangle(x_1, x_2), \langle \#, 1, \{ 1 \} \rangle)$,
$ep_2(g(g(g(\#,\#),\#),\#)) =$
$\langle g, 0, \{ 0, 1, 2 \} \rangle(\langle g, 1, \{ 0, 1, 2 \} \rangle(x_1, x_2), \langle \#, 1, \{ 1 \} \rangle)$,
$ep_3(g(g(g(\#,\#),\#),\#)) =$
$\langle g, 0, \{ 0, 1, 2 \} \rangle(\langle g, 1, \{ 0, 1, 2 \} \rangle(\langle g, 0, \{ 1, 2 \} \rangle(x_1, x_2), \langle \#, 1, \{ 1 \} \rangle), \langle \#, 1, \{ 1 \} \rangle)$,
$ep_4(g(g(g(\#,\#),\#),\#)) =$
$\langle g, 0, \{ 0, 1, 2 \} \rangle(\langle g, 1, \{ 0, 1, 2 \} \rangle(\langle g, 0, \{ 1, 2 \} \rangle(\langle \#, 1, \{ 1 \} \rangle, \langle \#, 1, \{ 1 \} \rangle),$
$\langle \#, 1, \{ 1 \} \rangle), \langle \#, 1, \{ 1 \} \rangle)$.

Let

$$\tau = max\{ height(l) \mid l \in lhs(R) \} + 1. \tag{4}$$

Lemma 1. *For any $s, t \in T_\Sigma$ and $\mu \in pos(s)$, if $ep_\tau(s/\mu) = ep_\tau(t)$, then $root(s) = root(s[\mu \leftarrow t])$, $s^\mathcal{A} = s[\mu \leftarrow t]^\mathcal{A}$, and*

$$\{ u^\mathcal{B} \mid u \in SFOI(s) \} \subseteq \{ u^\mathcal{B} \mid u \in SFOI(s[\mu \leftarrow t]) \}. \tag{5}$$

Proof. Let $s, t \in T_\Sigma$, and $\mu \in pos(s)$ such that

$$ep_\tau(s/\mu) = ep_\tau(t).$$

If $\mu = \lambda$, then $s/\mu = s$ and $t = s[\mu \leftarrow t]$. By our assumption, $ep_\tau(s) = ep_\tau(s[\mu \leftarrow t])$. Consequently, the statement of the lemma holds.

From now on, we assume that $\mu \neq \lambda$. Then $root(s) = root(s[\mu \leftarrow t])$ and $s^\mathcal{A} = s[\mu \leftarrow t]^\mathcal{A}$ by Proposition 3. It is left to show (5). To this end, let $b \in \{ u^\mathcal{B} \mid u \in SFOI(s) \}$. Then there is an OI reduction sequence

(a) $s \longrightarrow_{\bar\alpha_1, \bar\beta_1, \bar l_1 \to \bar r_1} \bar s_1 \longrightarrow_{\bar\alpha_2, \bar\beta_2, \bar l_2 \to \bar r_2} \cdots \longrightarrow_{\bar\alpha_n, \bar\beta_n, \bar l_n \to \bar r_n} \bar s_n$

such that

$$b = \bar{s}_n^{\mathcal{B}}.$$

Along (a), R may have produced more than one copies of the subtree s/μ of s. We now consider the rewriting of the ℓth copy of s/μ. R may rewrite at some $\alpha \in \mathit{pos}(s)$ such that

$$\alpha \prec \mu$$

and the left-hand side of the applied rule $l \to r$ overlaps with the ℓth copy of s/μ. Then R rewrites the subtrees $s/\chi_1, \ldots, s/\chi_k$, where $\chi_1, \ldots, \chi_k \in \mathit{pos}(s)$, of the ℓth copy of s/μ. There are

$$\nu_{\ell 1}, \ldots, \nu_{\ell n_\ell} \in \{\, \alpha\xi \mid \xi \in \mathit{vpos}(l) \,\} \tag{6}$$

such that for each χ_j, $j = 1, \ldots, k$, there is a $\nu_{\ell i}$, $i \in \{\, 1, \ldots, n_\ell \,\}$, such that

$$\mu \preceq \nu_{\ell i} \preceq \chi_j. \tag{7}$$

Thus $\nu_{\ell i}$ is an extension of μ and a prefix of χ_j. Then by $\alpha \prec \mu$, (4), (6), and (7),

$$\nu_{\ell i} = \mu\eta_{\ell i} \text{ for some } \eta_{\ell i} \in N^* \text{ with } \mathit{length}(\eta_{\ell i}) \leq \tau - 2 \text{ for each } i = 1, \ldots, n_\ell. \tag{8}$$

Note that we do not necessarily rewrite at the positions $\nu_{\ell 1}, \ldots, \nu_{\ell n_\ell}$, rather we rewrite at the extensions $\chi_1, \ldots, \chi_k$ of the positions $\nu_{\ell 1}, \ldots, \nu_{\ell n_\ell}$. The positions $\nu_{11}, \ldots, \nu_{1n_1}, \nu_{21}, \ldots, \nu_{2n_2}, \ldots$ are simply denoted by $\nu_1, \ldots, \nu_m$. We rearrange the rewrite steps of the OI reduction sequence (a) into the OI reduction sequence (b) below such that the following conditions hold.

• Beginning with step $d_0 + 1$, we carry out the reduction steps that take place at the extensions of μ. Intuitively, the suffix of (b), starting at step $d_0 + 1$, consists of the reduction of the copies of s/μ.

• When rewriting the ℓth copy of s/μ, first we carry out all reduction steps at the extensions of $\nu_{\ell 1}$, second we carry out all reduction steps at the extensions of $\nu_{\ell 2}$, and so on. Thus we have

(b) $s \longrightarrow_{\alpha_1,\beta_1,l_1\to r_1} s_1 \longrightarrow_{\alpha_2,\beta_2,l_2\to r_2} s_2 \longrightarrow_{\alpha_3,\beta_3,l_3\to r_3} \cdots \longrightarrow_{\alpha_{d_0},\beta_{d_0},l_{d_0}\to r_{d_0}}$
$s_{d_0} = v[s/\nu_1, \ldots, s/\nu_m] \longrightarrow_{\alpha_{d_0+1},\beta_{d_0+1},l_{d_0+1}\to r_{d_0+1}} \cdots \longrightarrow_{\alpha_{d_1},\beta_{d_1},l_{d_1}\to r_{d_1}}$
$v[v_1, s/\nu_2, \ldots, s/\nu_m] \longrightarrow_{\alpha_{d_1+1},\beta_{d_1+1},l_{d_1+1}\to r_{d_1+1}} \cdots \longrightarrow_{\alpha_{d_m},\beta_{d_m},l_{d_m}\to r_{d_m}}$
$v[v_1, v_2, \ldots, v_m] = s_n$,
where

- $v \in \overline{T}(X_m)$, $m \geq 0$,
- $\vartheta_i \in \mathit{vpos}(v)$ and $\mathit{lab}(v, \vartheta_i) = x_i$ for $i = 1, \ldots, m$,
- $0 \leq d_0 < d_1 < \cdots < d_m = n$,
- $\bar{s}_n = s_n$.

Moreover, the following four conditions hold.

1. For each $1 \leq i \leq d_0$, μ is not a prefix of α_i, and for each $d_0 + 1 \leq i \leq d_m$, $\mu \preceq \alpha_i$.
2. For each $1 \leq i \leq m$, we have $\nu_i = \mu\zeta_i$ for some $\zeta_i \in N^*$ with $length(\zeta_i) \leq \tau - 2$.
3. For each $1 \leq i \leq m$ and $d_{i-1} + 1 \leq j \leq d_i$, we have $\nu_i \preceq \alpha_j$ and $\vartheta_i \preceq \beta_j$.
4. For each $1 \leq i \leq m$, there is an OI reduction sequence $s/\nu_i \Rightarrow_{R,OI} v_i$ for some subtree v_i of s_n.

We obtained Condition 2 by (8). Since $b = \bar{s}_n^{\mathcal{B}}$ and $\bar{s}_n = s_n$, we also have

$$b = s_n^{\mathcal{B}}.$$

Note that along the reduction in Condition 4 we do not necessarily rewrite s/ν_i at the position λ. By Condition 2 and the assumption $ep_\tau(s/\mu) = ep_\tau(t)$ of the lemma, $val(s/\nu_i) = val(s[\mu \leftarrow t]/\nu_i)$. Hence by Condition 4, for each $1 \leq i \leq m$, there is a tree $z_i \in T_\Sigma$ such that

$$s[\mu \leftarrow t]/\nu_i \underset{R,OI}{\Rightarrow} z_i \text{ and } z_i^{\mathcal{B}} = v_i^{\mathcal{B}}.$$

Hence, as R is left-linear, we have the OI reduction sequence
$s[\mu \leftarrow t] \longrightarrow_{\alpha_1,\beta_1,l_1\to r_1} s'_1 \longrightarrow_{\alpha_2,\beta_2,l_2\to r_2} s'_2 \longrightarrow_{\alpha_3,\beta_3,l_3\to r_3} \cdots \longrightarrow_{\alpha_{d_0},\beta_{d_0},l_{d_0}\to r_{d_0}}$
$s'_{d_0} = v[s[\mu \leftarrow t]/\nu_1, \ldots, s[\mu \leftarrow t]/\nu_m] \longrightarrow_{\gamma_{d_0+1},\delta_{d_0+1},l_{d_0+1}\to r_{d_0+1}} \cdots$
$\longrightarrow_{\gamma_{e_1},\delta_{e_1},l_{e_1}\to r_{e_1}} v[z_1, s[\mu \leftarrow t]/\nu_2, \ldots, s[\mu \leftarrow t]/\nu_m] \longrightarrow_{\gamma_{e_1+1},\delta_{e_1+1},l_{e_1+1}\to r_{e_1+1}} \cdots$
$\longrightarrow_{\gamma_{e_m},\delta_{e_m},l_{e_m}\to r_{e_m}} v[z_1, z_2, \ldots, z_m]$.
Here

$$d_0 < e_1 < \cdots < e_m,$$

and the following conditions hold.

- For each $1 \leq i \leq d_0$, $s'_i \in T_\Sigma$.
- For each $1 \leq i \leq d_0$, μ is not a prefix of α_i, and for each $d_0 + 1 \leq i \leq e_m$, $\mu \preceq \gamma_i$.
- For each $d_0 + 1 \leq j \leq e_1$, we have $\nu_1 \preceq \gamma_j$ and $\vartheta_1 \preceq \delta_j$. Furthermore, for each $2 \leq i \leq m$ and $e_{i-1} + 1 \leq j \leq e_i$, we have $\nu_i \preceq \gamma_j$ and $\vartheta_i \preceq \delta_j$.
- For each $1 \leq i \leq m$, there is an OI reduction sequence $s[\mu \leftarrow t]/\nu_i \Rightarrow_{R,OI} z_i$.

By the definition of z_i, $i = 1, \ldots, m$, and Proposition 3,

$$s_n^{\mathcal{B}} = v[v_1, \ldots, v_m]^{\mathcal{B}} = v[z_1, \ldots, z_m]^{\mathcal{B}}.$$

Thus, by $b = s_n^{\mathcal{B}}$, we have $b \in \{\, u^{\mathcal{B}} \mid u \in SFOI(s[\mu \leftarrow t]) \,\}$. This proves (5), and with it also finishes the proof of the lemma. □

Lemma 2. *For any $s, t \in T_\Sigma$ and $\mu \in pos(s)$, if $ep_\tau(s/\mu) = ep_\tau(t)$, then $\{\, u^{\mathcal{B}} \mid u \in SFOI(s) \,\} = \{\, u^{\mathcal{B}} \mid u \in SFOI(s[\mu \leftarrow t]) \,\}$.*

Proof. The inclusion from left to right is proven in Lemma 1. The other inclusion can be verified by applying Lemma 1 on $s' = s[\mu \leftarrow t]$ and $t' = s/\mu$. □

By Lemma 1 and Lemma 2 we have the following result.

Lemma 3. *For any $s, t \in T_\Sigma$ and $\mu \in pos(s)$, if $ep_\tau(s/\mu) = ep_\tau(t)$, then $val(s) = val(s[\mu \leftarrow t])$.*

Lemma 4. *For any $s, t \in T_\Sigma$ and $\mu \in pos(s)$, if $ep_\tau(s/\mu) = ep_\tau(t)$, then $ep_\tau(s) = ep_\tau(s[\mu \leftarrow t])$.*

Proof. Let $s, t \in T_\Sigma$ and $\mu \in pos(s)$ such that

$$ep_\tau(s/\mu) = ep_\tau(t). \tag{9}$$

Let $\alpha \in pos(s)$ such that $length(\alpha) \leq \tau - 1$. We now show that

$$lab(ec(s), \alpha) = lab(ec(s[\mu \leftarrow t]), \alpha). \tag{10}$$

For this we distinguish the following three cases.

Case 1: α is not a prefix of μ and μ is not a prefix of α. In this case, $s/\alpha = s[\mu \leftarrow t]/\alpha$, and hence we have (10).

Case 2: $\alpha \preceq \mu$. Then by (9) and Lemma 3, (10) holds.

Case 3: $\mu \prec \alpha$. Then $\alpha = \mu\beta$ for some $\beta \in pos(s/\mu)$, and
$lab(ec(s), \alpha) = lab(ec(s/\mu), \beta) =$
$lab(ec(t), \beta) = lab(ec(s[\mu \leftarrow t]), \alpha)$ (by (9)).
Hence (10) holds.

In the aggregate, we conclude that for each $\alpha \in pos(s)$ with $length(\alpha) \leq \tau - 1$, (10) holds. Consequently, by Remark 2, the τ-normal prefix of $ec(s)$ is equal to the τ-normal prefix of $ec(s[\mu \leftarrow t])$, that is, $ep_\tau(s) = ep_\tau(s[\mu \leftarrow t])$. □

For each $i \geq 0$, let $P_i = \{\, ep_\tau(s) \mid s \in T_\Sigma \text{ and } height(s) \leq i \,\}$. It should be clear that P_i can be computed effectively for every $i \geq 0$.

Example 6. Observe that $\tau = 2$. The trees in T_Σ with height at most two are the following:
$\#,\ \ f(\#),\ \ g(\#,\#),$
$f(f(\#)),\ f(g(\#,\#)),\ g(\#,f(\#)),\ g(f(\#),\#),\ g(f(\#),f(\#)),\ g(\#,g(\#,\#)),$
$g(f(\#),g(\#,\#)),\ g(g(\#,\#),\#),\ \ g(g(\#,\#),f(\#)),\ \ g(g(\#,\#),g(\#,\#)).$
Moreover, $P_0 = \{\, \langle \#, 1, \{\,1\,\} \rangle \,\}$,
P_1 consists of the trees:
$\langle \#, 1, \{\,1\,\} \rangle,\ \langle f, 1, \{\,1\,\} \rangle(\langle \#, 1, \{\,1\,\} \rangle),\ \langle g, 0, \{\,1, 2\,\} \rangle(\langle \#, 1, \{\,1\,\} \rangle, \langle \#, 1, \{\,1\,\} \rangle),$
and P_2 consists of the trees:
$\langle \#, 1, \{\,1\,\} \rangle,\ \langle f, 1, \{\,1\,\} \rangle(\langle \#, 1, \{\,1\,\} \rangle),\ \langle g, 0, \{\,1, 2\,\} \rangle(\langle \#, 1, \{\,1\,\} \rangle, \langle \#, 1, \{\,1\,\} \rangle),$
$\langle f, 1, \{\,1\,\} \rangle(\langle f, 1, \{\,1\,\} \rangle(x_1)),\ \langle f, 0, \{\,1, 2\,\} \rangle(\langle g, 0, \{\,1, 2\,\} \rangle(x_1, x_2)),$

$\langle g, 0, \{1,2\}\rangle(\langle \#, 1, \{1\}\rangle, \langle f, 1, \{1\}\rangle(x_1)),$
$\langle g, 0, \{1,2\}\rangle(\langle f, 1, \{1\}\rangle(x_1), \langle \#, 1, \{1\}\rangle),$
$\langle g, 0, \{1,2\}\rangle(\langle f, 1, \{1\}\rangle(x_1), \langle f, 1, \{1\}\rangle(x_2)),$
$\langle g, 1, \{0,1,2\}\rangle(\langle \#, 1, \{1\}\rangle, \langle g, 0, \{1,2\}\rangle(x_1, x_2)),$
$\langle g, 1, \{0,1,2\}\rangle(\langle f, 1, \{1\}\rangle(x_1), \langle g, 0, \{1,2\}\rangle(x_2, x_3)),$
$\langle g, 1, \{0,1,2\}\rangle(\langle g, 0, \{1,2\}\rangle(x_1, x_2), \langle \#, 1, \{1\}\rangle),$
$\langle g, 1, \{0,1,2\}\rangle(\langle g, 0, \{1,2\}\rangle(x_1, x_2), \langle f, 1, \{1\}\rangle(x_3)),$
$\langle g, 0, \{0,1,2\}\rangle(\langle g, 0, \{1,2\}\rangle(x_1, x_2), \langle g, 0, \{1,2\}\rangle(x_3, x_4)).$

By Remark 3 and the definition of P_i, $i \geq 0$,

$$P_0 \subseteq P_1 \subseteq \cdots \subseteq NORM_{\Delta,\tau}. \tag{11}$$

Since $NORM_{\Delta,\tau}$ is a finite set, there is a smallest index n such that $P_n = P_{n+1}$. We call n the evaluation index. From now on, throughout this section, n stands for the evaluation index.

Lemma 5. *$P_n = P_\ell$ for $\ell > n$.*

Proof. We proceed by induction on ℓ.

Base Step: $\ell = n + 1$. Then $P_n = P_\ell$ by the definition of the evaluation index n.

Induction step: Let $\ell \geq n + 2$, and assume that $P_n = P_j$ for all j such that $n \leq j \leq \ell - 1$.

Let $s \in T_\Sigma$ with $height(s) = \ell$. Then $s = f(s_1, \ldots s_m)$ for some $f \in \Sigma_m$, $m \geq 1$, and $s_1, \ldots s_m \in T_\Sigma$. For each $i = 1, \ldots, m$, we define the tree $v_i \in T_\Sigma$ as follows. If $height(s_i) \leq \ell - 2$, then let $v_i = s_i$. Assume that $height(s_i) = \ell - 1$. By the induction hypothesis, there is a tree $v_i \in T_\Sigma$ such that $height(v_i) \leq \ell - 2$ and $ep_\tau(s_i) = ep_\tau(v_i)$. Let

$$v = f(v_1, \ldots, v_m).$$

Then $height(v) \leq \ell - 1$. Hence, by (11) $ep_\tau(v) \in P_{\ell-1}$. By the induction hypothesis,

$$ep_\tau(v) \in P_n. \tag{12}$$

By Lemma 4,
$ep_\tau(f(s_1, s_2, \ldots, s_m)) = ep_\tau(f(v_1, s_2 \ldots s_m)) =$
$ep_\tau(f(v_1, v_2, s_3, \ldots, s_m)) = \cdots = ep_\tau(f(v_1, v_2, \ldots, v_m)).$
Thus $ep_\tau(s) = ep_\tau(v)$. By (12), $ep_\tau(s) \in P_n$. Since $s \in T_\Sigma$ with $height(s) = \ell$ is arbitrary, we have $P_\ell \subseteq P_n$. By (11) $P_\ell = P_n$. □

Lemma 6. *The evaluation index n can be computed effectively.*

Proof. Let us compute P_i for $i = 0, 1, \ldots$ until we find that $P_i = P_{i+1}$. As mentioned above, the procedure stops at some i. By definition, n equals to this i. □

By the definition of $val(p)$ for $p \in T_\Sigma$, and the definition of P_i, $i \geq 0$, we have the following.

Lemma 7. *$SFOI(L) \subseteq M$ if and only if for each $p \in P_n$ with $p = \langle f, a, C\rangle(p_1, \ldots, p_m)$, $m \geq 0$, if $a \in A_f$ then $C \subseteq B_f$.*
$SFOI(L) \cap M \neq \emptyset$ if and only if there is $p \in P_n$ such that $p = \langle f, a, C\rangle(p_1, \ldots, p_m)$, $m \geq 0$, $a \in A_f$, and $C \cap B_f \neq \emptyset$.

Example 7. We now show that $SFOI(L) \not\subseteq M$ and $SFOI(L) \cap M \neq \emptyset$. To this end we need not compute n and P_n. Observe that
$\langle g, 0, \{1, 2\}\rangle(\langle \#, 1, \{1\}\rangle, \langle \#, 1, \{1\}\rangle) \in P_2 \subseteq P_n$, $0 \in A_f$, and $\{1, 2\} \not\subseteq B_f$.
Hence by Lemma 7, $SFOI(L) \not\subseteq M$. Furthermore,
$\langle g, 0, \{0, 1, 2\}\rangle(\langle g, 0, \{1, 2\}\rangle(x_1, x_2), \langle g, 0, \{1, 2\}\rangle(x_3, x_4)) \in P_2 \subseteq P_n$, $0 \in A_f$, and $\{0, 1, 2\} \cap B_f \neq \emptyset$.
Hence by Lemma 7, $SFOI(L) \cap M \neq \emptyset$.

Proof. (of Theorem 1) Recall that $\mathcal{A} = (\Sigma, A, R_{\mathcal{A}}, A_f)$ and $\mathcal{B} = (\Sigma, B, R_{\mathcal{B}}, B_f)$ are total dbtas such that $L(\mathcal{A}) = L$ and $L(\mathcal{B}) = M$. Let us compute the evaluation index n and the set P_n (cf. Lemma 6). For each $p \in P_n$ with $p = \langle f, a, C\rangle(p_1, \ldots, p_m)$, $m \geq 0$, by direct inspection we decide whether $a \in A_f$ implies $C \subseteq B_f$. If for each $p \in P_n$, the answer is yes, then $SFOI(L) \subseteq M$. Otherwise, $SFOI(L) \not\subseteq M$, see Lemma 7.

For each $p \in P_n$ with $p = \langle f, a, C\rangle(p_1, \ldots, p_m)$, $m \geq 0$, by direct inspection we decide whether $a \in A_f$ implies $C \cap B_f \neq \emptyset$. If the answer is yes for some $p \in P_n$, then $SFOI(L) \cap M \neq \emptyset$. Otherwise, $SFOI(L) \cap M = \emptyset$, see Lemma 7. □

4 Second Order OI Joinability Problem

We show that the second order OI joinability problem is undecidable for left-linear TRSs.

For an alphabet Δ, we also consider the alphabets $\overline{\Delta} = \{\overline{a} \mid a \in \Delta\}$ and $\widehat{\Delta} = \{\widehat{a} \mid a \in \Delta\}$. The alphabets Δ, $\overline{\Delta}$, and $\widehat{\Delta}$ are pairwise disjoint. For each word $w \in \Delta^*$, the word $\overline{w} \in \overline{\Delta}^*$ is defined as follows.

- If $w = \lambda$, then $\overline{w} = \lambda$.
- If $w = az$ for some $a \in \Sigma$ and $z \in \Delta^*$, then $\overline{w} = \overline{a}\,\overline{z}$.

For each word $w \in \Delta^*$, we define the word $\widehat{w} \in \widehat{\Delta}^*$ in a similar way to $\overline{w}$.

Let $\langle \mathbf{w}, \mathbf{z}\rangle = \langle (w_1, \ldots, w_n),\ (z_1, \ldots, z_n)\rangle$ be a PCS over the alphabet Δ. We associate the ranked alphabet Σ, the recognizable tree languages L and M over Σ, and the TRS R over Σ with the PCS $\langle \mathbf{w}, \mathbf{z}\rangle$. To this end, we consider the sets Δ, $\overline{\Delta}$, $\widehat{\Delta}$, and $\Gamma = \{1, \ldots, n\}$ of unary symbols. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_4$, $\Sigma_0 = \{\#\}$, $\Sigma_1 = \Delta \cup \overline{\Delta} \cup \widehat{\Delta} \cup \Gamma$, $\Sigma_2 = \{f, g\}$, $\Sigma_4 = \{h\}$. Let

$$L = \{ f(s, t) \mid s \in (T_{\Gamma \cup \{\#\}} - \{\#\}), t \in T_{\widehat{\Delta} \cup \{\#\}} \}$$

and

$$M = \{ g(s, t) \mid s \in T_{\Delta \cup \{\#\}}, t \in T_{\overline{\Delta} \cup \{\#\}} \}.$$

The TRS R consists of the following rules:

- $f(x_1, x_2) \to h(x_1, x_1, x_2, x_2)$, $g(x_1, x_2) \to h(x_1, x_2, x_1, x_2)$,
- $k(x_1) \to w_k(x_1)$, $k(x_1) \to \overline{z_k}(x_1)$ for $k = 1, \dots, n$,
- $\widehat{a}(x_1) \to a(x_1)$, $\widehat{a}(x_1) \to \overline{a}(x_1)$ for $a \in \Delta$.

Example 8. Let $\Delta = \{\, a, b \,\}$. Let PCS $\langle \mathbf{w}, \mathbf{z} \rangle = \langle (a, ab),\ (aa, b) \rangle$. Note that 12 is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. The ranked alphabet Σ consists of

- the nullary symbol $\#$,
- the unary symbols $1, 2, a, b, \overline{a}, \overline{b}, \widehat{a}, \widehat{b}$,
- the binary symbols f, g, and
- the symbol h of rank 4.

The TRS R consists of the following rules:

- $f(x_1, x_2) \to h(x_1, x_1, x_2, x_2)$, $g(x_1, x_2) \to h(x_1, x_2, x_1, x_2)$,
- $1x_1 \to ax_1$, $2x_1 \to abx_1$, $1x_1 \to \overline{a}\overline{a}x_1$, $2x_1 \to \overline{b}x_1$,
- $\widehat{a}x_1 \to ax_1$, $\widehat{a}x_1 \to \overline{a}x_1$, $\widehat{b}x_1 \to bx_1$, $\widehat{b}x_1 \to \overline{b}x_1$.

Let

$$L = \{\, f(s, t) \mid s \in (T_{\{\, 1,2,\# \,\}} - \{\, \# \,\}), t \in T_{\{\, \widehat{a},\widehat{b},\# \,\}} \,\}$$

and

$$M = \{\, g(s, t) \mid s \in T_{\{\, a,b,\# \,\}}, t \in T_{\{\, \overline{a},\overline{b},\# \,\}} \,\}.$$

Since 12 is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$, we have the following OI reduction sequence.

$f(12\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R h(12\#, 12\#, \widehat{a}\widehat{a}\widehat{b}\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R h(a2\#, 12\#, \widehat{a}\widehat{a}\widehat{b}\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R$
$h(aab\#, 12\#, \widehat{a}\widehat{a}\widehat{b}\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R h(aab\#, \overline{a}\overline{a}2\#, \widehat{a}\widehat{a}\widehat{b}\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R$
$h(aab\#, \overline{a}\overline{a}\overline{b}\#, \widehat{a}\widehat{a}\widehat{b}\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R h(aab\#, \overline{a}\overline{a}\overline{b}\#, a\widehat{a}\widehat{b}\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R$
$h(aab\#, \overline{a}\overline{a}\overline{b}\#, aa\widehat{b}\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R h(aab\#, \overline{a}\overline{a}\overline{b}\#, aab\#, \widehat{a}\widehat{a}\widehat{b}\#) \to_R$
$h(aab\#, \overline{a}\overline{a}\overline{b}\#, aab\#, \overline{a}\widehat{a}\widehat{b}\#) \to_R h(aab\#, \overline{a}\overline{a}\overline{b}\#, aab\#, \overline{a}\overline{a}\widehat{b}\#) \to_R$
$h(aab\#, \overline{a}\overline{a}\overline{b}\#, aab\#, \overline{a}\overline{a}\overline{b}\#)$.

Consider the OI-reduction sequence
$g(aab\#, \overline{a}\overline{a}\overline{b}\#) \to_R h(aab\#, \overline{a}\overline{a}\overline{b}\#, aab\#, \overline{a}\overline{a}\overline{b}\#)$.

Thus, the terms $f(12\#, \widehat{a}\widehat{a}\widehat{b}\#)$ and $g(aab\#, \overline{aab}\#)$ are OI joinable. Consequently L and M are OI joinable.

Lemma 8. *The PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ has a solution if and only if L and M are OI joinable.*

Proof. Assume that the index sequence $k_1, \ldots, k_\ell$ is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. Then

$$w_{k_1} \ldots w_{k_\ell} = z_{k_1} \ldots z_{k_\ell}. \tag{13}$$

We have the OI-reduction sequence

(a) $f(k_1 \ldots k_\ell \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R$
$h(k_1 \ldots k_\ell \#, k_1 \ldots k_\ell \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R$
$h(w_{k_1} k_2 \ldots k_\ell \#, k_1 \ldots k_\ell \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R^*$
$h(w_{k_1} \ldots w_{k_\ell} \#, k_1 \ldots k_\ell \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{z_{k_1}} k_2 \ldots k_\ell \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#, \widehat{w_{k_1}} \ldots w_{\widehat{k_\ell}} \#) \rightarrow_R^*$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}} \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R^*$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}} \#, w_{k_1} \widehat{w_{k_2}} \ldots \widehat{w_{k_\ell}} \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R^*$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}} \#, w_{k_1} \ldots w_{k_\ell} \#, \widehat{w_{k_1}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R^*$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}} \#, w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \widehat{w_{k_2}} \ldots \widehat{w_{k_\ell}} \#) \rightarrow_R^*$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}} \#, w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}} \#)$.

We also have the OI-reduction sequence

(b) $g(w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}} \#) \rightarrow_R$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}} \#, w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}} \#)$.

By (13),
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}} \#, w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}} \#) =$
$h(w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}} \#, w_{k_1} \ldots w_{k_\ell} \#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}} \#)$.

Hence $SFOI(L) \cap SFOI(M) \neq \emptyset$. That is, L and M are OI joinable.

Conversely, assume that L and M are OI joinable, i.e., $SFOI(L) \cap SFOI(M) \neq \emptyset$. Consequently, there are $u \in SFOI(L) \cap SFOI(M)$, $1 \leq k_1, \ldots, k_\ell \leq n$, $\ell \geq 1$,

$$s \in T_{\widehat{\Delta} \cup \{\#\}}, t_1 \in \Delta^*, \text{ and } t_2 \in \overline{\Delta}^* \tag{14}$$

such that

$$f(k_1 \ldots k_\ell \#, s) \underset{R,OI}{\Rightarrow} u \tag{15}$$

and

$$g(t_1 \#, t_2 \#) \underset{R,OI}{\Rightarrow} u. \tag{16}$$

We write (15) in the form

(c) $f(k_1 \ldots k_\ell \#, s) \rightarrow_{R,OI} h(k_1 \ldots k_\ell \#, k_1 \ldots k_\ell \#, s, s) \Rightarrow_{R,OI} u =$
$h(u_1 \#, u_2 \#, u_3 \#, u_4 \#)$ for some $u_1, u_2, u_3, u_4 \in \Sigma^*$. Here

(d) $k_1 \ldots k_\ell \# \Rightarrow_{R,OI} u_1 \#$ and $k_1 \ldots k_\ell \# \Rightarrow_{R,OI} u_2 \#$, and

(e) $s \Rightarrow_{R,OI} u_3 \#$ and $s \Rightarrow_{R,OI} u_4 \#$.

We write (16) in the form

(f) $g(t_1 \#, t_2 \#) \rightarrow_{R,OI} h(t_1 \#, t_2 \#, t_1 \#, t_2 \#) = u$. Hence

$$t_1 \# = u_1 \# = u_3 \# \text{ and } t_2 \# = u_2 \# = u_4 \#. \tag{17}$$

Consequently, by (14) and (e), we get that $s = \widehat{u_3} \#$ and $\overline{u_3} \# = u_4 \#$. Hence, by (17), we have

$$\overline{t_1} = t_2. \tag{18}$$

By (d) and (17),

$$k_1 \dots k_\ell \# \underset{R,OI}{\Rightarrow} t_1 \# \text{ and } k_1 \dots k_\ell \# \underset{R,OI}{\Rightarrow} t_2 \#.$$

Since $t_1 \in \Delta^*$ and $t_2 \in \overline{\Delta}^*$,

$$w_{k_1} \dots w_{k_\ell} = t_1 \text{ and } \overline{z}_{k_1} \dots \overline{z_{k_\ell}} = t_2. \tag{19}$$

By (18) and (19), we have

$$\overline{w_{k_1}} \dots \overline{w_{k_\ell}} = \overline{z_{k_1}} \dots \overline{z_{k_\ell}}.$$

Consequently (13) holds. Hence the index sequence $k_1, \dots, k_\ell$ is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. □

Theorem 2. *For left-linear TRSs, the second order OI joinability problem is undecidable.*

Proof. Let $\langle \mathbf{w}, \mathbf{z} \rangle$ be a Post Correspondence System over the alphabet Δ. We associated the ranked alphabet Σ, the recognizable tree languages L and M over Σ and the TRS R over Σ with the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. By Lemma 8, the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ has a solution if and only if L and M are OI joinable. By Proposition 2, there is no algorithm which takes a PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ as input and determines whether or not there is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. □

5 Second Order IO Common Ancestor Problem

We show that the second order common IO ancestor problem is undecidable for right-linear TRSs.

Let $\langle \mathbf{w}, \mathbf{z} \rangle = \langle (w_1, \dots, w_n), (z_1, \dots, z_n) \rangle$ be a Post Correspondence System over the alphabet Δ. We associate the ranked alphabet Σ, the recognizable tree languages L and M over Σ, and the TRS R over Σ with the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. To this end, we consider the sets Δ, $\overline{\Delta} = \{ \overline{d} \mid d \in \Delta \}$, $\Gamma = \{ 1, \dots, n \}$, and $\overline{\Gamma} = \{ \overline{1}, \dots, \overline{n} \}$ of unary symbols. Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_4$, $\Sigma_0 = \{ \# \}$, $\Sigma_1 = \Delta \cup \overline{\Delta} \cup \Gamma \cup \overline{\Gamma}$, $\Sigma_2 = \{ g, h \}$, $\Sigma_4 = \{ f \}$.
The TRS R consists of the following rules:

- $f(x_1, x_1, x_2, x_2) \to g(x_1, x_2)$, $f(x_1, x_2, x_1, x_2) \to h(x_1, x_2)$,
- $kx_1 \to \overline{k}x_1$, $kx_1 \to w_k x_1$, $\overline{k}x_1 \to \overline{z_k}x_1$ for $k = 1, \dots, n$,
- $dx_1 \to \overline{d}x_1$ for $d \in \Delta$.

Note that R is right-linear.
Let $L = \{ g(s,t) \mid s \in (T_{\overline{\Gamma} \cup \{ \# \}} - \{ \# \}), t \in T_{\overline{\Delta} \cup \{ \# \}} \}$ and $M = \{ h(s,t) \mid s \in T_{\Delta \cup \{ \# \}}, t \in T_{\overline{\Delta} \cup \{ \# \}} \}$.

Example 9. Let $\Delta = \{\, a, b \,\}$. Let PCS $\langle \mathbf{w}, \mathbf{z} \rangle = \langle (a, ab), (aa, b) \rangle$. Note that 12 is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. The ranked alphabet Σ consists of

- the nullary symbol $\#$,
- the unary symbols $1, 2, a, b, \overline{1}, \overline{2}, \overline{a}, \overline{b}$,
- the symbol f of rank 4, and the binary symbols g and h.

The TRS R consists of the following rules:

- $f(x_1, x_1, x_2, x_2) \to g(x_1, x_2)$, $f(x_1, x_2, x_1, x_2) \to h(x_1, x_2)$,
- $1x_1 \to ax_1$, $2x_1 \to abx_1$,
- $\overline{1}x_1 \to \overline{aa}x_1$, $\overline{2}x_1 \to \overline{b}x_1$,
- $1x_1 \to \overline{1}x_1$, $2x_1 \to \overline{2}x_1$,
- $ax_1 \to \overline{a}x_1$, $bx_1 \to \overline{b}x_1$.

Furthermore

$$L = \{\, g(s, t) \mid s \in (T_{\{\, \overline{1}, \overline{2}, \# \,\}} - \{\, \# \,\}), t \in T_{\{\, \overline{a}, \overline{b}, \# \,\}} \,\}$$

and

$$M = \{\, h(s, t) \mid s \in T_{\{\, a, b, \# \,\}}, t \in T_{\{\, \overline{a}, \overline{b}, \# \,\}} \,\}.$$

Since 12 is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$, we have the following two IO reduction sequences.

$f(12\#, \overline{12}\#, aab\#, \overline{aa}\overline{b}\#) \to_R f(12\#, \overline{12}\#, aa\overline{b}\#, \overline{aa}\overline{b}\#) \to_R$
$f(12\#, \overline{12}\#, a\overline{a}\overline{b}\#, \overline{aa}\overline{b}\#) \to_R f(12\#, \overline{12}\#, \overline{aa}\overline{b}\#, \overline{aa}\overline{b}\#) \to_R$
$f(1\overline{2}\#, \overline{12}\#, \overline{aa}\overline{b}\#, \overline{aa}\overline{b}\#) \to_R f(\overline{12}\#, \overline{12}\#, \overline{aa}\overline{b}\#, \overline{aa}\overline{b}\#) \to_R g(\overline{12}\#, \overline{aa}\overline{b}\#)$

and

$f(12\#, \overline{12}\#, aab\#, \overline{aa}\overline{b}\#) \to_R f(1ab\#, \overline{12}\#, aab\#, \overline{aa}\overline{b}\#) \to_R$
$f(aab\#, \overline{12}\#, aab\#, \overline{aa}\overline{b}\#) \to_R f(aab\#, \overline{1}\overline{b}\#, aab\#, \overline{aa}\overline{b}\#) \to_R$
$f(aab\#, \overline{aa}\overline{b}\#, aab\#, \overline{aa}\overline{b}\#) \to_R h(aab\#, \overline{aa}\overline{b}\#)$.

Thus $f(12\#, \overline{12}\#, aab\#, \overline{aa}\overline{b}\#)$ is a common IO ancestor of the terms $g(\overline{12}\#, \overline{aa}\overline{b}\#)$ and $h(aab\#, \overline{aa}\overline{b}\#)$. Consequently, the tree languages L and M have a common IO one-pass ancestor.

Lemma 9. *The PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ has a solution if and only if L and M have a common IO ancestor.*

Proof. Assume that the index sequence $k_1, \dots, k_\ell$ is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. Then

$$w_{k_1} \dots w_{k_\ell} = z_{k_1} \dots z_{k_\ell}. \tag{20}$$

Furthermore, we have the IO-reduction sequences

$f(k_1 \dots k_\ell\#, \overline{k_1} \dots \overline{k_\ell}\#, w_{k_1} \dots w_{k_\ell}\#, \overline{z_{k_1}} \dots \overline{z_{k_\ell}}\#) \to_R$

$f(k_1 \ldots k_{\ell-1}\overline{k_\ell}\#, \overline{k_1} \ldots \overline{k_\ell}\#, w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R^*$
$f(\overline{k_1} \ldots \overline{k_\ell}\#, \overline{k_1} \ldots \overline{k_\ell}\#, w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R^*$
$f(\overline{k_1} \ldots \overline{k_\ell}\#, \overline{k_1} \ldots \overline{k_\ell}\#, w_{k_1} \ldots w_{k_{\ell-1}}\overline{w_{k_\ell}}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R^*$
$f(\overline{k_1} \ldots \overline{k_\ell}\#, \overline{k_1} \ldots \overline{k_\ell}\#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R$
$g(\overline{k_1} \ldots \overline{k_\ell}\#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}}\#)$

and

$f(k_1 \ldots k_\ell\#, \overline{k_1} \ldots \overline{k_\ell}\#, w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R$
$f(k_1 \ldots k_{\ell-1}w_{k_\ell}\#, \overline{k_1} \ldots \overline{k_\ell}\#, w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R^*$
$f(w_{k_1} \ldots w_{k_\ell}\#, \overline{k_1} \ldots \overline{k_\ell}\#, w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R^*$
$f(w_{k_1} \ldots w_{k_\ell}\#, \overline{k_1} \ldots \overline{k_{\ell-1}}\overline{z_{k_\ell}}\#, w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R^*$
$f(w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}}, \ldots, \overline{z_{k_\ell}}\#, w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#) \rightarrow_R$
$h(w_{k_1} \ldots w_{k_\ell}\#, \overline{z_{k_1}}, \ldots, \overline{z_{k_\ell}}\#) = h(w_{k_1} \ldots w_{k_\ell}\#, \overline{w}_{k_1} \ldots \overline{w}_{k_\ell}\#)$.

Hence $g(\overline{k_1} \ldots \overline{k_\ell}\#, \overline{w_{k_1}} \ldots \overline{w_{k_\ell}}\#) \in L$ and $h(w_{k_1} \ldots w_{k_\ell}\#, \overline{w}_{k_1} \ldots \overline{w}_{k_\ell}\#) \in M$ have a common ancestor.

Conversely, assume that $p \in T_\Sigma(X)$ is a common ancestor of $g(\overline{k_1} \ldots \overline{k_\ell}\#, q) \in L$ with $1 \leq k_1, \ldots, k_\ell \leq n$, $\ell \geq 1$,

$$q \in T_{\overline{\Delta} \cup \{\#\}} \tag{21}$$

and $h(s, t) \in M$ with

$$s \in T_{\Delta \cup \{\#\}} \tag{22}$$

and

$$t \in T_{\overline{\Delta} \cup \{\#\}}. \tag{23}$$

Then p is a ground term. Hence

$$p = f(p_1, p_2, p_3, p_4) \text{ for some } p_1, p_2, p_3, p_4 \in T_\Sigma.$$

Furthermore, there are IO one-pass reduction sequences
$f(p_1, p_2, p_3, p_4) \rightarrow_R^* f(\overline{k_1} \ldots \overline{k_\ell}\#, \overline{k_1} \ldots \overline{k_\ell}\#, q, q) \rightarrow_R g(\overline{k_1} \ldots \overline{k_\ell}\#, q)$,
and
$f(p_1, p_2, p_3, p_4) \rightarrow_R^* f(s, t, s, t) \rightarrow_R h(s, t)$.
Here
(a) $p_1 \Rightarrow_{R,IO} \overline{k_1} \ldots \overline{k_\ell}\#$, (b) $p_1 \Rightarrow_{R,IO} s$,
(c) $p_2 \Rightarrow_{R,IO} \overline{k_1} \ldots \overline{k_\ell}\#$, (d) $p_2 \Rightarrow_{R,IO} t$,
(e) $p_3 \Rightarrow_{R,IO} q$ and $p_3 \Rightarrow_{R,IO} s$, and (f) $p_4 \Rightarrow_{R,IO} q$ and $p_4 \Rightarrow_{R,IO} t$.
Consequently, (g) $p_1 \in T_{\Gamma \cup \{\#\}}$, $p_2 \in T_{\overline{\Gamma} \cup \{\#\}}$, $p_3 \in T_{\Delta \cup \{\#\}}$.
We proceed as follows.
(h) $p_1 = k_1 \ldots k_\ell\#$ by (a) and (g).
(i) $p_2 = \overline{k_1} \ldots \overline{k_\ell}\#$ by (c) and (g).
(j) $s = w_{k_1} \ldots w_{k_\ell}\#$ by (22), (b), and (h).
(k) $t = \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#$ by (23), (d), and (i).
(l) $q = \overline{s}$ by (21), (22), (e), and (g).
(m) $q = t$ by (21), (23), and (f).
(n) $\overline{s} = t$ by (l) and (m).
By (j), (k), and (n),

$$\overline{w_{k_1} \ldots w_{k_\ell}}\# = \overline{z_{k_1}} \ldots \overline{z_{k_\ell}}\#.$$

Thus (20) holds. Hence the index sequence $k_1, \ldots, k_\ell$ is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. □

Theorem 3. *For right-linear TRSs, the second order common IO ancestor problem is undecidable.*

Proof. Let $\langle \mathbf{w}, \mathbf{z} \rangle$ be a Post Correspondence System over the alphabet Δ. We associated the ranked alphabet Σ, the recognizable tree languages L and M over Σ, and the TRS R over Σ with the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. By Lemma 9, the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ has a solution if and only if L and M have a common IO one-pass reduction ancestor. By Proposition 2, there is no algorithm which takes a PCS $\langle \mathbf{w}, \mathbf{z} \rangle$ as input and determines whether or not there is a solution of the PCS $\langle \mathbf{w}, \mathbf{z} \rangle$. □

6 Conclusion

We summed up the existing results in the literature and our contribution in Table 1.

We conjecture that for right-linear TRSs, the second order IO reachability problem and the second order common OI ancestor problem are decidable. We raise the following open problems.

Problem 1. *Given a TRS R and a recognizable tree language L, is it decidable whether $SFIO(L)$ is recognizable and whether $SFOI(L)$ is recognizable?*

For any linear TRS R, $\Rightarrow_{R,IO}$ is equal to $\Rightarrow_{R,OI}$. Hence for any linear TRS R and recognizable tree language L, $SFIO(L) = SFOI(L)$. Therefore, we raise the following question.

Problem 2. *Given a TRS R and a recognizable tree language L, is it decidable whether $SFIO(L) \subseteq SFOI(L)$, whether $SFOI(L) \subseteq SFIO(L)$, and whether $SFIO(L) = SFOI(L)$?*

References

[1] Baader, F. and Nipkow, T. Term Rewriting and All That, Cambridge University Press, Cambridge, United Kingdom, 1998.

[2] Van Den Brand, M., Klint, P., and Vinju, J. J. Term rewriting with traversal functions. ACM Transactions on Software Engineering and Methodology 12(2): 152-190 (2003).

[3] Bravenboer, M., Kalleberg, K. T., Vermaas R., and Visser E. Stratego/XT 0.17. A language and toolset for program transformation. Sci. Comput. Program. 72(1-2): 52-70 (2008).

[4] Dauchet, M., De Comité, F. A Gap Between Linear and Non Linear Term-Rewriting Systems. In Lescanne, P editor, Rewriting Techniques and Applications, RTA-87, Proceedings. Lecture Notes in Computer Science 256 Springer 1987, pages 95-104.

[5] Fülöp, Z., Jurvanen, E., Steinby, M., and Vágvölgyi, S. On one-pass term rewriting. In Brim, L., Gruska, J., Zlatuska, J. editors, Mathematical Foundations of Computer Science, 1998 Lecture Notes in Computer Science, **1450**, Springer Publishing Company, Berlin, 1998, pages 248-256; see also in Acta Cybernetica, **14** (1999) 83-98.

[6] Fülöp, Z. and Vágvölgyi, S. A Characterization of Irreducible Sets Modulo Left-Linear Term Rewriting Systems by Tree Automata, Fundamenta Informaticae, XIII (1990) 211-226.

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

[8] Gécseg, F. and Steinby, M. Tree languages. In G. Rozenberg and A. Salomaa, editors, Beyond Words, volume 3 of Handbook of Formal Languages, chapter 1, pages 1-68. Springer-Verlag, Berlin, 1997.

[9] Genet, T., Reachability analysis of rewriting for software verification, Habilitation document, 2009.
http://www.irisa.fr/celtique/genet/Publications/habilitation.pdf.

[10] Gilleron, R., Tison, S., Regular Tree Languages and Rewrite Systems. Fundamenta Informaticae **24** (1995) 157-174.

[11] Harrison, M. Introduction to Formal Language Theory, Addison-Wesley Publishing Company, Boston, MA, USA 1978.

[12] Seynhaeve, F., Tison, S., Tommasi, M., Homomorphisms and Concurrent Term Rewriting. In Ciobanu, G., Paun, G. editors Fundamentals of Computation Theory, 12th International Symposium, FCT '99, Iasi, Romania, 1999, Proceedings. Springer 1999 Lecture Notes in Computer Science **1684** (1999) pages 475-487.

[13] J. Ouaknine, J., Potapov I., Worrell, J., Reachability Problems - 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings. Lecture Notes in Computer Science 8762, Springer 2014, ISBN 978-3-319-11438-5.

[14] Stratego/XT Manual

http://releases.strategoxt.org/strategoxt-manual/unstable/manual/one-page/index.html#tutorial-introduction.

[15] Visser, E. Program Transformation with Stratego/XT: Rules, Strategies, Tools, and Systems in Stratego/XT 0.9. In C. Lengauer, D. S. Batory, C. Consel, Odersky, M. editor, Domain-Specific Program Generation, International Seminar, Dagstuhl Castle, Germany, March 23-28, 2003, Revised Papers, Lecture Notes in Computer Science 3016 pages 216-238 Springer 2004.

Received 5th March 2014