PAPER *Special Issue on Selected Papers from LA Symposium*

Layered Transducing Term Rewriting System and Its Recognizability Preserving Property

Toshinori TAKAI[†], **Hiroyuki SEKI**[††], ***Regular Members***, **Youhei FUJINAKA**[††], ***Nonmember***, ***and*** **Yuichi KAJI**[††], ***Regular Member***

SUMMARY A term rewriting system which effectively preserves recognizability (EPR-TRS) has good mathematical properties. In this paper, a new subclass of TRSs, layered transducing TRSs (LT-TRSs) is defined and its recognizability preserving property is discussed. The class of LT-TRSs contains some EPR-TRSs, e.g., $\{f(x) \to f(g(x))\}$ which do not belong to any of the known decidable subclasses of EPR-TRSs. Bottom-up linear tree transducer, which is a well-known computation model in the tree language theory, is a special case of LT-TRS. We present a sufficient condition for an LT-TRS to be an EPR-TRS. Also reachability and joinability are shown to be decidable for LT-TRSs.

key words: *term rewriting system, tree automaton, recognizability, recognizability preserving property, layered transducing TRS*

1. Introduction

Tree automaton is a natural extension of finite-state automaton on strings. A set of ground terms (tree language) T is *recognizable* if there exists a tree automaton which accepts T. Tree automaton inherits good mathematical properties from finite-state automaton. For example, the class of recognizable sets is closed under boolean operations (union, intersection and complementation), and decision problems such as emptiness and membership are decidable for a recognizable set. Let $\mathcal{L}(\mathcal{A})$ denote the language accepted by a tree automaton $\mathcal{A}$. For a TRS $\mathcal{R}$ and a tree language T, define $(\to^*_{\mathcal{R}})(T) = \{t \mid \exists s \in T \text{ s.t. } s \to^*_{\mathcal{R}} t\}$. A TRS $\mathcal{R}$ *effectively preserves recognizability* (abbreviated as EPR) if for any tree automaton $\mathcal{A}$, $(\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ is also recognizable and a tree automaton $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ can be effectively constructed. Due to the above mentioned properties of recognizable sets, some important problems, e.g., reachability, joinability and local confluence are decidable for EPR-TRSs [8], [9]. Furthermore, with additional conditions, strong normalization property, neededness and unifiability become decidable for EPR-TRSs [4], [11], [14].

The problem to decide whether a given TRS is EPR is undecidable [7], and decidable subclasses of EPR-TRSs have been proposed in a serie of works [3], [9]–[11], [13], [14]. These subclasses put a rather strong constraint on the syntax of the right-hand side of a rewrite rule. For example, the right-hand side of a rewrite rule in a linear semi-monadic TRS (L-SM-TRS) [3] is either a variable or $f(t_1, t_2, \ldots, t_n)$ where each t_i $(1 \le i \le n)$ is either a variable or a ground term. Linear generalized semi-monadic TRS (L-GSM-TRS) [9] and right-linear finite path-overlapping TRS (RL-FPO-TRS) [14] weaken this constraint, but some simple EPR-TRSs such as $\{f(x) \to f(g(x))\}$ still do not belong to any of the known decidable subclasses of EPR-TRSs. To show that a given TRS $\mathcal{R}$ is EPR, for a given tree automaton $\mathcal{A}$, a tree automaton $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ should be constructed. The above mentioned restrictions on the right-hand side of a rewrite rule are sufficient conditions for a procedure of automata construction to halt.

In this paper, a new subclass of TRSs, *layered transducing TRSs* (LT-TRSs) is defined and its recognizability preserving property is discussed. Intuitively, an LT-TRS is a TRS such that certain unary function symbols are specified as *markers* and a marker moves from leaf to root in each rewrite step. Bottom-up linear tree transducer [6], which is a well-known computation model in the tree language theory, can be considered as a special case of LT-TRS. We propose a procedure which, for a given tree automaton $\mathcal{A}$ and an LT-TRS $\mathcal{R}$, constructs a tree automaton $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$. The procedure introduces a state $[z, q]$ which is the product of a state z already belonging to $\mathcal{A}_*$ and a marker q and constructs a transition rule which is the product of a transition rule already in $\mathcal{A}_*$ and a rewrite rule in $\mathcal{R}$.

However, an LT-TRS is not always EPR and the above procedure does not always halt. We present a sufficient condition for the procedure to halt. The subclass of LT-TRSs which satisfy the sufficient condition is still incomparable with any of the known decidable subclasses of EPR-TRSs. Especially, the class contains some EPR-TRSs, such as $\{f(x) \to f(g(x))\}$ mentioned above. Finally, reachability and joinability are shown to be decidable for LT-TRSs.

The rest of the paper is organized as follows. After providing preliminary definitions in Sect. 2, LT-TRS is defined in Sect. 3. A procedure for automata construc-

Manuscript received March 22, 2002.
Manuscript revised August 20, 2002.

[†]The author is with National Institute of Advanced Industrial Science and Technology, Amagasaki-shi, 661–0974 Japan.
[††]The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.

tion is presented and the partial correctness of the procedure is proved in Sect. 4. Sufficient conditions for the construction procedure to halt are presented in Sect. 5. Also reachability and joinability are shown to be decidable for LT-TRS in Sect. 5.

2. Preliminaries

2.1 Term Rewriting Systems

We use the usual notions for terms, substitutions, etc (see [1] for details). Let Σ be a *signature* and $\mathcal{V}$ be an enumerable set of *variables*. An element in Σ is called a *function symbol* and the *arity* of $f \in \Sigma$ is denoted by $a(f)$. A function symbol c with $a(c) = 0$ is called a *constant*. The set of *terms*, defined in the usual way, is denoted by $\mathcal{T}(\Sigma, \mathcal{V})$. The set of variables occurring in t is denoted by $\mathcal{V}ar(t)$. A term t is *ground* if $\mathcal{V}ar(t) = \emptyset$. The set of ground terms is denoted by $\mathcal{T}(\Sigma)$. A ground term in $\mathcal{T}(\Sigma)$ is also called a Σ*-term*. A term is *linear* if no variable occurs more than once in the term. A *substitution* σ is a mapping from $\mathcal{V}$ to $\mathcal{T}(\Sigma, \mathcal{V})$, and written as $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ where t_i with $1 \leq i \leq n$ is a term which substitutes for the variable x_i. The term obtained by applying a substitution σ to a term t is written as $t\sigma$. A *position* in a term t is defined as a sequence of positive integers as usual, and the set of all positions in a term t is denoted by $\mathcal{P}os(t)$. An empty sequence λ is called the root position. A subterm of t at a position o is denoted by t/o. If t/o is a variable then o is called a *variable position*. If a term t is obtained from a term t' by replacing the subterms of t' at positions $o_1, \ldots, o_m$ ($o_i \in \mathcal{P}os(t'), o_i$ and o_j are disjoint if $i \neq j$) with terms $t_1, \ldots, t_m$, respectively, then we write $t = t'[o_i \leftarrow t_i \mid 1 \leq i \leq m]$.

A *rewrite rule* over a signature Σ is an ordered pair of terms in $\mathcal{T}(\Sigma, \mathcal{V})$, written as $l \to r$. The variable restriction ($\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ and $l \notin \mathcal{V}$) is not assumed unless stated otherwise. A *term rewriting system* (*TRS*) over Σ is a finite set of rewrite rules over Σ. For terms t, t' and a TRS $\mathcal{R}$, we write $t \to_{\mathcal{R}} t'$ if there exists a position $o \in \mathcal{P}os(t)$, a substitution σ and a rewrite rule $l \to r \in \mathcal{R}$ such that $t/o = l\sigma$ and $t' = t[o \leftarrow r\sigma]$. Define $\to^*_{\mathcal{R}}$ to be the reflexive and transitive closure of $\to_{\mathcal{R}}$. Also the transitive closure of $\to_{\mathcal{R}}$ is denoted by $\to^+_{\mathcal{R}}$. The subscript $\mathcal{R}$ of $\to_{\mathcal{R}}$ is omitted if $\mathcal{R}$ is clear from the context. A *redex* (*in* $\mathcal{R}$) is an instance of l for some $l \to r \in \mathcal{R}$. A *normal form* (in $\mathcal{R}$) is a term which has no redex as its subterm. Let $\mathrm{NF}_{\mathcal{R}}$ denote the set of all ground normal forms in $\mathcal{R}$. A rewrite rule $l \to r$ is *left-linear* (resp. *right-linear*) if l is linear (resp. r is linear). A rewrite rule is *linear* if it is left-linear and right-linear. A TRS $\mathcal{R}$ is *left-linear* (resp. *right-linear, linear*) if every rule in $\mathcal{R}$ is left-linear (resp. right-linear, linear).

Notions such as *reachability*, *joinability*, *confluence* and *local confluence* are defined in the usual way.

2.2 Tree Automata

A *tree automaton* (abbreviated as *TA*) [6] is defined by a 4-tuple $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{final})$ where Σ is a signature, $\mathcal{P}$ is a finite set of states, $\mathcal{P}_{final} \subseteq \mathcal{P}$ is a set of final states, and Δ is a finite set of transition rules of the form $f(p_1, \ldots, p_n) \to p$ where $f \in \Sigma$, $a(f) = n$, and $p_1, \ldots, p_n, p \in \mathcal{P}$ or of the form $p' \to p$ where $p', p \in \mathcal{P}$. A rule with the former form is called a *non-ε-rule* and a rule with the latter form is called an *ε-rule*. In this paper, we use $p, p', p_1, p_2, \ldots$ to denote a state. Consider the set of ground terms $\mathcal{T}(\Sigma \cup \mathcal{P})$ where we define $a(p) = 0$ for $p \in \mathcal{P}$. A *transition* of a TA can be regarded as a rewrite relation on $\mathcal{T}(\Sigma \cup \mathcal{P})$ by regarding transition rules in Δ as rewrite rules on $\mathcal{T}(\Sigma \cup \mathcal{P})$. For terms t and t' in $\mathcal{T}(\Sigma \cup \mathcal{P})$, we write $t \vdash_{\mathcal{A}} t'$ if and only if $t \to_{\Delta} t'$. If $t \vdash_{\mathcal{A}} t'$ is caused by an ε-rule then $t \vdash_{\mathcal{A}} t'$ is called an ε-transition. The reflexive and transitive closure and the transitive closure of $\vdash_{\mathcal{A}}$ is denoted by $\vdash^*_{\mathcal{A}}$ and $\vdash^+_{\mathcal{A}}$ respectively. For a TA $\mathcal{A}$ and $t \in \mathcal{T}(\Sigma)$, if $t \vdash^*_{\mathcal{A}} p_f$ for a final state $p_f \in \mathcal{P}_{final}$, then we say t is *accepted* by $\mathcal{A}$. The set of ground terms accepted by $\mathcal{A}$ is denoted by $\mathcal{L}(\mathcal{A})$. Also let $\mathcal{L}_p(\mathcal{A}) = \{t \mid t \vdash^*_{\mathcal{A}} p\}$ for a state p. A set T of ground terms is *recognizable* if there is a TA $\mathcal{A}$ such that $T = \mathcal{L}(\mathcal{A})$. A state $p \in \mathcal{P}$ is *reachable* in $\mathcal{A}$ if there exists a Σ-term t such that $t \vdash^*_{\mathcal{A}} p$. A state $p \in \mathcal{P}$ is *useful* in $\mathcal{A}$ if there exists a Σ-term t, a position $o \in \mathcal{P}os(t)$ and a final state $p_f \in \mathcal{P}_{final}$ such that $t \vdash^*_{\mathcal{A}} t[o \leftarrow p] \vdash^*_{\mathcal{A}} p_f$. It is not difficult to show that for a given TA $\mathcal{A}$, we can construct a TA $\mathcal{A}'$ which satisfies $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$ and has only useful states. Recognizable sets inherit some useful properties of regular (string) languages.

Lemma 1 [6]**:** The class of recognizable sets is effectively closed under union, intersection and complementation. For a recognizable set T, the following problems are decidable. (1) Does a given ground term t belong to T? (2) Is T empty? □

2.3 TRS which Preserves Recognizability

For a TRS $\mathcal{R}$ and a set T of ground terms, define $(\to^*_{\mathcal{R}})(T) = \{t \mid \exists s \in T \text{ s.t. } s \to^*_{\mathcal{R}} t\}$. A TRS $\mathcal{R}$ is said to *effectively preserve recognizability* if, for any tree automaton $\mathcal{A}$, the set $(\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ is also recognizable and we can effectively construct a tree automaton which accepts $(\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$. In this paper, the class of TRSs which effectively preserve recognizability is written as EPR-TRS.

Theorem 1: If a TRS $\mathcal{R}$ belongs to EPR-TRS, then the reachability relation and the joinability relation for $\mathcal{R}$ are decidable [8]. It is also decidable whether $\mathcal{R}$ is locally confluent or not [9]. □

Unfortunately it is undecidable whether a given

TRS belongs to EPR-TRS or not [7]. Therefore decidable subclasses of EPR-TRS have been proposed, for example, ground TRS by Brainerd [2], right-linear monadic TRS (RL-M-TRS) by Salomaa [13], linear semi-monadic TRS (L-SM-TRS) by Coquidé et al. [3], right-linear semi-monadic TRS (RL-SM-TRS), which is equivalent to the inverse of left-linear growing TRS by Nagaya and Toyama [11], linear generalized semi-monadic TRS (L-GSM-TRS) by Gyenizse and Vágvölgyi [9], and right-linear finite path overlapping TRS (RL-FPO-TRS) by Takai et al. [14].

Theorem 2: RL-M-TRS $\subset$ RL-SM-TRS $\subset$ RL-FPO-TRS $\subset$ EPR-TRS and ground TRS $\subset$ L-SM-TRS $\subset$ L-GSM-TRS $\subset$ RL-FPO-TRS. All inclusions are proper. $\square$

Réty [12] defined a subclass of TRSs and showed that the class effectively preserves recognizability for the subclass $\mathcal{C}$ of tree languages of which member is a set $\{t\sigma \mid t$ is a linear term and σ is a substitution such that $x\sigma$ is a constructor term for each $x \in \mathcal{V}ar(t)\}$ (abbreviated as $\mathcal{C}$-EPR). $\mathcal{R}_3$ of Example 5 in Sect. 4 is not an EPR-TRS but it is $\mathcal{C}$-EPR.

3. Layered Transducing TRS

A new class of TRS named *layered transducing TRS* (*LT-TRS*) is proposed in this section.

Definition 1: Let $\Sigma = \mathcal{F} \cup \mathcal{Q}$ be a signature where $\mathcal{F} \cap \mathcal{Q} = \emptyset$. A function symbol q in $\mathcal{Q}$ is called a *marker* and $a(q) = 1$. A *layered transducing TRS* (*LT-TRS*) is a linear TRS over Σ in which each rewrite rule has one of the following forms:

(i) $f(t_1, \cdots, t_n) \to r$, or
(ii) $t_1 \to r$

where

1. $f \in \mathcal{F}$,
2. t_i ($1 \le i \le n$ in Case (i) and $i = 1$ in Case (ii)) is either a ground term or a term of the form $q_i(l_i)$ where $q_i \in \mathcal{Q}$ and l_i is either a variable or a ground term and
3. r is either a variable or a term of the form $q(r_1)$ where $q \in \mathcal{Q}$ and $r_1 \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. $\square$

Example 1: Let $g \in \mathcal{F}$ with $a(g) = 1$ and let $q \in \mathcal{Q}$. $\mathcal{R}_1 = \{q(x) \to q(g(x))\}$ is an LT-TRS. Note that $\mathcal{R}_1$ is an EPR-TRS but is not an FPO-TRS [14]. $\square$

Example 2: Let $f, g, h \in \mathcal{F}$, $q_1, q_2, q \in \mathcal{Q}$. $\mathcal{R}_2 = \{f(q_1(x_1), q_2(x_2)) \to q(g(h(x_2), x_1)),\ q_1(x_1) \to q(h(x_1))\}$ is an LT-TRS. $\square$

In this paper, we use a, b, c to denote a constant, f, g, h to denote a non-marker symbol, $q, q', q_1, q_2, \ldots$ to denote a marker and $s, t, t_1, t_2, \ldots$ to denote a term in $\mathcal{T}(\Sigma, \mathcal{V})$.

4. Construction of Tree Automata

In this section, we will present a procedure which takes an LT-TRS $\mathcal{R}$ and a tree automaton $\mathcal{A}$ as an input and constructs a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ if the procedure halts. Let $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{final})$ be a TA. By the definition of $(\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$,

$$\text{if } t \vdash^*_{\mathcal{A}} p \text{ and } t \to^*_{\mathcal{R}} s \text{ then } s \vdash^*_{\mathcal{A}_*} p \text{ also holds.}$$

To satisfy this property, the proposed procedure starts with $\mathcal{A}_0 = \mathcal{A}$ and constructs a series of TAs $\mathcal{A}_0, \mathcal{A}_1, \ldots$. We define $\mathcal{A}_*$ as the limit of this chain of TAs. For example, let $f(p_1, p_2) \to p \in \Delta$ and $f(q_1(x_1), q_2(x_2)) \to q(g(h(x_2), x_1)) \in \mathcal{R}$ and assume that

$$t = f(q_1(t_1), q_2(t_2)) \tag{1}$$
$$\vdash^*_{\mathcal{A}} f(q_1(p'_1), q_2(p'_2)) \vdash^*_{\mathcal{A}} f(p_1, p_2) \vdash_{\mathcal{A}} p. \tag{2}$$

Note that $f(q_1(t_1), q_2(t_2)) \to_{\mathcal{R}} q(g(h(t_2), t_1))(= t')$ and hence $\mathcal{A}_*$ is required to satisfy $q(g(h(t_2), t_1)) \vdash^*_{\mathcal{A}_*} p$. The procedure constructs a 'product' rule of $f(p_1, p_2) \to p$ and $f(q_1(x_1), q_2(x_2)) \to q(g(h(x_2), x_1))$ and some auxiliary rules so that $\mathcal{A}_*$ can simulate the transition sequence (2) when $\mathcal{A}_*$ reads $q(g(h(t_2), t_1))$. More precisely, new states $[p, q]$, $\langle h(p'_2)\rangle$ and $\langle g(h(p'_2), p'_1)\rangle$ are introduced and rules

$$\begin{array}{lcl} h(p'_2) & \to & \langle h(p'_2)\rangle, \\ g(\langle h(p'_2)\rangle, p'_1) & \to & \langle g(h(p'_2), p'_1)\rangle, \\ \langle g(h(p'_2), p'_1)\rangle & \to & [p, q] \end{array} \tag{3}$$

are constructed. The following transition rule is also added so that $s \vdash^*_{\mathcal{A}_*} [p, q]$ if and only if $q(s) \vdash^*_{\mathcal{A}_*} p$.

$$q([p, q]) \to p. \tag{4}$$

When $\mathcal{A}_*$ reads $q(g(h(t_2), t_1))$, we can see by (2) that

$$t' = q(g(h(t_2), t_1)) \vdash^*_{\mathcal{A}} q(g(h(p'_2), p'_1)). \tag{5}$$

$\mathcal{A}_*$ guesses that in a term t such that $t \to_{\mathcal{R}} t'$, the markers q_1 and q_2 were placed above the subterms t_1 and t_2, respectively, as in (1) and $\mathcal{A}_*$ behaves as if it reads q_1 and q_2 at p'_1 and p'_2. That is, $\mathcal{A}_*$ simulates the transition $f(p_1, p_2) \vdash_{\mathcal{A}} p$ by rules (3). Also see Fig. 1.

$$\begin{aligned} t' &\vdash^*_{\mathcal{A}} q(g(h(p'_2), p'_1)) \vdash_{\mathcal{A}} q(g(\langle h(p'_2)\rangle, p'_1)) \\ &\vdash_{\mathcal{A}} q(\langle g(h(p'_2), p'_1)\rangle) \vdash_{\mathcal{A}} q([p, q]) \vdash_{\mathcal{A}} p \end{aligned}$$

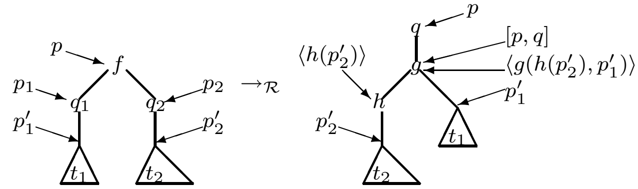


Fig. 1 An idea of automata construction.

The last transition is by (4); $\mathcal{A}_*$ encounters the marker q at the state $[p, q]$, which means that the guess was correct, and $\mathcal{A}_*$ changes its state to p by forgetting the guess q. The construction of new rules and states is repeated until $\mathcal{A}_i$ saturates. Hence, states with more than one nesting such as $[\langle f([\langle f([p, q_1])\rangle, q_2])\rangle, q_3]$ may be defined in general. For a state $z' \in \mathcal{Z}_i$, we identify $\langle z' \rangle$ with z'. then we implicitly assume that $\mathcal{F} \cap \mathcal{Q} = \emptyset$ and $\mathcal{Q}$ is a set of markers.

As mentioned above, the TA construction procedure introduces a state of the form $[z, q]$ or $\langle t \rangle$ where $z \in \mathcal{Z}$, q is a marker, $t \in \mathcal{T}(\Sigma \cup \mathcal{Z}) \backslash \mathcal{Z}$ and $\mathcal{Z}$ is the set of states of the TA being constructed. To slightly abuse the notation, for a state z, let $\langle z \rangle$ denote z itself. For example, if we write $\langle t_1 \rangle$ where $t_1 = [p, q]$ then $\langle t_1 \rangle$ denotes $[p, q]$. Similarly, if we write $\langle t_2 \rangle$ where $t_2 = \langle f(p) \rangle$ then $\langle t_2 \rangle$ denotes $\langle f(p) \rangle$ since $\langle f(p) \rangle$ itself is a state.

Procedure 1: The set difference is denoted by $A \setminus B (= \{x \mid x \in A \text{ and } x \notin B\})$. Suppose $\Sigma = \mathcal{F} \cup \mathcal{Q}$ and $\mathcal{F} \cap \mathcal{Q} = \emptyset$.

Input: a tree automaton $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{final})$ and an LT-TRS $\mathcal{R}$ over Σ. Without loss of generality, assume that (i) $\mathcal{A}$ has no ε-rule, (ii) every state in $\mathcal{P}$ is reachable and (iii) there exists a state p_{any} such that $\mathcal{L}_{p_{any}}(\mathcal{A}) = \mathcal{T}(\Sigma)$.

Output: a tree automaton $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\rightarrow^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$.

Step 1 Let $i := 0$ and $\mathcal{A}_0 = (\Sigma, \mathcal{Z}_0, \Delta_0, \mathcal{P}_{final}) := \mathcal{A}$. In **Step 2**–**Step 4**, this procedure constructs $\mathcal{A}_1, \mathcal{A}_2, \cdots$ by adding new states and transition rules to $\mathcal{A}_0$.

Step 2 Let $i := i + 1$ and $\mathcal{A}_i = (\Sigma, \mathcal{Z}_i, \Delta_i, \mathcal{P}_{final}) := \mathcal{A}_{i-1}$.

Step 3 For each rewrite rule $l \rightarrow r \in \mathcal{R}$, state $z \in \mathcal{Z}_{i-1}$ and substitution $\rho\colon \mathcal{V} \rightarrow \mathcal{Z}_{i-1}$ such that:

1. $x\rho = p_{any}$ for $x \in \mathcal{V}ar(r) \setminus \mathcal{V}ar(l)$,
2. $l\rho \vdash^*_{\mathcal{A}_{i-1}} z$ (6)

holds and no ε-transition occurs at the root position or at the variable positions o_j $(1 \leq j \leq h)$ where l has h distinct variables, $\mathcal{V}ar(l) = \{x_1, \ldots, x_h\}$, and x_j $(1 \leq j \leq h)$ occurs at a position o_j in l,

do the following:

(a) `if` $r \in \mathcal{V}$ `and` $r\rho \neq z$, `then add` $r\rho \rightarrow z$ `to` Δ_i;
(b) `if` $r \notin \mathcal{V}$, `then let` $r = q(r_1)$

`add` $\langle r_1\rho \rangle$ `and` $[z, q]$ `to` $\mathcal{Z}_i$;
`add` $\langle r_1\rho \rangle \rightarrow [z, q]$ `and` $q([z, q]) \rightarrow z$ `to` Δ_i;
`do` **ADDREC**(r_1, i, ρ).

Step 4 If $\mathcal{A}_{i-1} = \mathcal{A}_i$, then let $\mathcal{A}_* := \mathcal{A}_i$ and output $\mathcal{A}_*$, else go to **Step 2**. □

Procedure 2: (**ADDREC**) This procedure takes a term $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, an integer $i \geq 1$ and a substitution $\rho\colon \mathcal{V} \rightarrow \mathcal{Z}_{i-1}$ as an input, and adds new states and transition rules to $\mathcal{A}_i$ so that $t\sigma \vdash^*_{\mathcal{A}_i} \langle t\rho \rangle$ holds for every substitution $\sigma\colon \mathcal{V} \rightarrow \mathcal{T}(\Sigma)$ such that $\sigma = \{x_j \mapsto s_j \mid s_j \vdash^*_{\mathcal{A}_i} x_j\rho,\ 1 \leq j \leq h\}$.

ADDREC(t, i, ρ)=
  `if` $t = x$ `then` return;
  `else let` $t = h(t_1, \cdots, t_n)$
    `add` $\langle t_1\rho \rangle, \cdots, \langle t_n\rho \rangle$ `to` $\mathcal{Z}_i$;
    `add` $h(\langle t_1\rho \rangle, \cdots, \langle t_n\rho \rangle) \rightarrow \langle t\rho \rangle$ `to` Δ_i;
    `do` **ADDREC**(t_j, i, ρ) $(1 \leq j \leq n)$. □

Example 3: Let $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{final})$ be a TA where $\Sigma = \mathcal{F} \cup \mathcal{Q}$, $\mathcal{F} = \{f, g, h, c\}$, $\mathcal{Q} = \{q_1, q_2, q\}$, $\mathcal{P} = \{p_1, p'_1, p_2, p_c, p_f\}$, $\mathcal{P}_{final} = \{p_f\}$ and $\Delta = \{c \rightarrow p_c,\ q_1(p_c) \rightarrow p'_1,\ q_1(p'_1) \rightarrow p_1,\ q_2(p_c) \rightarrow p_2,\ f(p_1, p_2) \rightarrow p_f\}$. It can be easily verified that $\mathcal{L}(\mathcal{A}) = \{f(q_1(q_1(c)), q_2(c))\}$. We apply Procedure 1 to $\mathcal{A}$ and LT-TRS $\mathcal{R}_2$ of Example 2. For $i = 1$, $f(q_1(x_1), q_2(x_2)) \rightarrow q(g(h(x_2), x_1)) \in \mathcal{R}_2$ is considered. Let $\rho = \{x_1 \mapsto p'_1,\ x_2 \mapsto p_c\}$. Since $f(q_1(x_1), q_2(x_2))\rho = f(q_1(p'_1), q_2(p_c)) \vdash^*_{\mathcal{A}_0} f(p_1, p_2) \vdash_{\mathcal{A}_0} p_f$, condition (6) is satisfied and rules $\langle g(h(p_c), p'_1) \rangle \rightarrow [p_f, q]$ and $q([p_f, q]) \rightarrow p_f$ are added to Δ_1. Also $g(\langle h(p_c) \rangle, p'_1) \rightarrow \langle g(h(p_c), p'_1) \rangle$ and $h(p_c) \rightarrow \langle h(p_c) \rangle$ are constructed by **ADDREC**$(g(h(x_2), x_1), 1, \rho)$. Consider $q_1(x_1) \rightarrow q(h(x_1)) \in \mathcal{R}_2$. Since $q_1(p'_1) \vdash_{\mathcal{A}_0} p_1$, condition (6) is satisfied and rules $\langle h(p'_1) \rangle \rightarrow [p_1, q], q([p_1, q]) \rightarrow p_1$ and $h(p'_1) \rightarrow \langle h(p'_1) \rangle$ are constructed. For $\rho' = \{x_1 \mapsto p_c\}, \langle h(p_c) \rangle \rightarrow [p'_1, q]$ and $q([p'_1, q]) \rightarrow p'_1$ are constructed. The transition rules constructed in Procedure 1 are listed in Table 1. Since no rule is added to $\mathcal{A}_2$, the procedure halts and we obtain $\mathcal{A}_* = \mathcal{A}_2$ as the output. We can verify that $\mathcal{L}(\mathcal{A}_*) = (\rightarrow^*_{\mathcal{R}_2})(\mathcal{L}(\mathcal{A}))$. □

Example 4: Let $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{final})$, $\Sigma = \mathcal{F} \cup \mathcal{Q}$, $\mathcal{F} = \{c, g\}$, $\mathcal{Q} = \{q\}$, $\mathcal{P} = \{p_c, p_f\}$, $\mathcal{P}_{final} = \{p_f\}$ and $\Delta = \{c \rightarrow p_c, q(p_c) \rightarrow p_f\}$. Clearly, $\mathcal{L}(\mathcal{A}) = \{q(c)\}$. If we apply Procedure 1 to $\mathcal{A}$ and $\mathcal{R}_1$ of Example 1, then for $i = 1$ of the procedure, $q(x) \rightarrow q(g(x)) \in \mathcal{R}_1$ and $\rho_1 = \{x \mapsto p_c\}$ are considered and $g(p_c) \rightarrow \langle g(p_c) \rangle$, $\langle g(p_c) \rangle \rightarrow [p_f, q]$ and $q([p_f, q]) \rightarrow p_f$ are added. For $i = 2$, $q(x) \rightarrow q(g(x)) \in \mathcal{R}_1$ and $\rho_2 = \{x \mapsto [p_f, q]\}$ are considered and $g([p_f, q]) \rightarrow \langle g([p_f, q]) \rangle$ and $\langle g([p_f, q]) \rangle \rightarrow [p_f, q]$ are added. Since no rule is added

Table 1 The transition rules constructed by Procedure 1 (Example 3).

	Step 3	**ADDREC**
$\mathcal{A}_1$	$\langle g(h(p_c), p'_1) \rangle \rightarrow [p_f, q]$ $q([p_f, q]) \rightarrow p_f$ $\langle h(p'_1) \rangle \rightarrow [p_1, q]$ $q([p_1, q]) \rightarrow p_1$ $\langle h(p_c) \rangle \rightarrow [p'_1, q]$ $q([p'_1, q]) \rightarrow p'_1$	$g(\langle h(p_c) \rangle, p'_1) \rightarrow \langle g(h(p_c), p'_1) \rangle$ $h(p_c) \rightarrow \langle h(p_c) \rangle$ $h(p'_1) \rightarrow \langle h(p'_1) \rangle$

when $i = 3$, the procedure halts with $\mathcal{A}_* = \mathcal{A}_2$. Clearly, $\mathcal{L}(\mathcal{A}_*) = \{q(g^n(c)) \mid n \geq 0\}$. Note that the transition $g([p_f, q]) \vdash_{\mathcal{A}_*} \langle g([p_f, q])\rangle \vdash_{\mathcal{A}_*} [p_f, q]$ simulates infinitely many rewrite steps caused by the rule $q(x) \to q(g(x))$. In the methods proposed in [14] and [5] (without approximation), infinitely many states such as $\langle g^n(p_c)\rangle$ and $\langle q(g^n(p_c))\rangle$ $(n \geq 1)$ are introduced to simulate each rewrite step $q(g^{n-1}(c)) \to_{\mathcal{R}_1} q(g^n(c))$ by a different transition $g(\langle g^{n-1}(p_c)\rangle) \vdash \langle g^n(p_c)\rangle$, and thus the construction does not halt in their methods. □

Example 5: Let $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{\mathit{final}})$, $\Sigma = \mathcal{F} \cup \mathcal{Q}$, $\mathcal{F} = \{c, f\}$, $\mathcal{Q} = \{q\}$, $\mathcal{P} = \mathcal{P}_{\mathit{final}} = \{p\}$, $\Delta = \{c \to p, f(p) \to p, q(p) \to p\}$ and $\mathcal{R}_3 = \{f(q(x)) \to q(f(x))\}$. $\mathcal{R}_3$ is an LT-TRS. Assume that Procedure 1 is executed for $\mathcal{A}$ and $\mathcal{R}_3$. For $i = 1$, consider a substitution $\rho_1 = \{x \mapsto p\}$, then $f(q(x))\rho_1 = f(q(p)) \vdash^*_{\mathcal{A}_0} p$. Hence, $f(p) \to \langle f(p)\rangle$, $\langle f(p)\rangle \to [p, q]$ and $q([p, q]) \to p$ are added to Δ_1. Next, for a substitution $\rho_2 = \{x \mapsto [p, q]\}$, $f(g(x))\rho_2 = f(q([p, q])) \vdash_{\mathcal{A}_1} f(p) \vdash_{\mathcal{A}_0} p$ holds and $f([p, q]) \to \langle f([p, q])\rangle$ and $\langle f([p, q])\rangle \to [p, q]$ are added to Δ_2. For the same ρ_2, $f(q(x))\rho_2 \vdash_{\mathcal{A}_1} f(p) \vdash_{\mathcal{A}_1} \langle f(p)\rangle$ also holds and $\langle f([p, q])\rangle \to [\langle f(p)\rangle, q]$ and $q([\langle f(p)\rangle, q]) \to \langle f(p)\rangle$ are added to Δ_2. The procedure repeats a similar construction and does not halt. □

Note that $\mathcal{R}_3$ is not an EPR-TRS since for a recognizable set $T_1 = \{(fq)^n(c) \mid n \geq 0\}$, $(\to^*_{\mathcal{R}_3})(T_1) \cap \mathrm{NF}_{\mathcal{R}_3} = \{q^n(f^n(c)) \mid n \geq 0\}$ is not recognizable.

We first show a few technical lemmas which will be used for the proof of soundness (Lemma 6) and completeness (Lemma 7) of Procedure 1.

Lemma 2: (i) If $f(z_1, \ldots, z_n) \to z \in \Delta_i \setminus \Delta_0$ $(i \geq 1, f \in \mathcal{F})$, then $z = \langle \tau\rho\rangle$ for some $\tau \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and $\rho: \mathcal{V} \to \mathcal{Z}_{i-1}$. (ii) If $q(z_1) \to z \in \Delta_i \setminus \Delta_0$ $(i \geq 1, q \in \mathcal{Q})$, then $z \in \mathcal{P}$ or $z = \langle \tau\rho\rangle$ for some τ and ρ, and $z_1 = [z, q]$.

Proof. (i) Obvious from **ADDREC**. (ii) Obvious from Step 3 (b) of Procedure 1. □

Lemma 3: If Step 3 is executed for a rule $l \to q(r_1)$, state $z \in \mathcal{Z}_{i-1}$ and substitution $\rho: \mathcal{V} \to \mathcal{Z}_{i-1}$ and a state of the form $\langle t\rho\rangle$ is constructed, then t is a subterm of r_1.

Proof. A state $\langle t\rho\rangle$ mentioned in the lemma is constructed either Step 3 (b) or **ADDREC**. If $\langle t\rho\rangle$ is constructed in Step 3 (b), then $t = r_1$ and the lemma holds. Assume $\langle t\rho\rangle$ is constructed in **ADDREC**. Note that when **ADDREC** is called from Step 3 (b), its first argument is r_1 and after that **ADDREC** is recursively called based on the term structure of its first argument. Hence, t is a subterm of r_1 in this case. □

Lemma 4: Let $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{\mathit{final}})$ be a TA. Assume that every state $p \in \mathcal{P}$ is reachable (resp. useful) in $\mathcal{A}$. If Procedure 1 is executed for $\mathcal{A}$ and an LT-TRS $\mathcal{R}$, then every state $z \in \mathcal{Z}_i$ constructed during the execution of Procedure 1 is reachable (resp. useful) in $\mathcal{A}_i$.

Proof. See Appendix. □

Note that an ε-rule is constructed only in Step 3 (a) or (b) of Procedure 1.

Lemma 5: (i) An ε-rule in Δ_i constructed in Step 3 (a) of Procedure 1 is one of the following forms:

$$p' \to p, \qquad [z', q'] \to p,$$
$$p' \to \langle \tau\rho\rangle, \quad [z', q'] \to \langle \tau\rho\rangle$$

where $p, p' \in \mathcal{P}, z' \in \mathcal{Z}_{i-1}, q' \in \mathcal{Q}, \tau \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and $\rho: \mathcal{V} \to \mathcal{Z}_{i-1}$. (ii) An ε-rule in Δ_i constructed in Step 3 (b) of Procedure 1 is one of the following forms:

$$p' \to [z, q], \quad [z', q'] \to [z, q], \quad \langle \tau\rho\rangle \to [z, q]$$

where $p' \in \mathcal{P}, z, z' \in \mathcal{Z}_{i-1}, q, q' \in \mathcal{Q}, \tau \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and $\rho: \mathcal{V} \to \mathcal{Z}_{i-1}$.

Proof. See Appendix. □

Lemma 6: (Soundness) Let $i \geq 1$, $t \in \mathcal{T}(\Sigma)$ and $\tau \in \mathcal{T}(\Sigma, \mathcal{V})$. Let $t \vdash^*_{\mathcal{A}_i} z''$.

(A) If $z'' = p \in \mathcal{P}$ then there exists a Σ-term s such that $s \vdash^*_{\mathcal{A}_{i-1}} p$ and $s \to^*_{\mathcal{R}} t$.
(B) If $z'' = [z, q]$ then there exists a Σ-term s such that $s \vdash^*_{\mathcal{A}_{i-1}} z$ and $s \to^*_{\mathcal{R}} q(t)$.
(C) If $z'' = \langle \tau\rho\rangle$ then there exists a substitution $\sigma: \mathcal{V} \to \mathcal{T}(\Sigma)$ such that $\tau\sigma \to^*_{\mathcal{R}} t$ and $x\sigma \vdash^*_{\mathcal{A}_{i-1}} x\rho$ for $x \in \mathcal{V}ar(\tau)$.

Proof. We will prove the lemma by double induction on i and the length of the transition sequence $t \vdash^*_{\mathcal{A}_i} z''$. In the rest of the proof, for a rule $l \to r$ in $\mathcal{R}$, we assume $\mathcal{V}ar(l) = \{x_1, \ldots, x_h\}$ and x_j $(1 \leq j \leq h)$.
(A) Assume $t \vdash^*_{\mathcal{A}_i} p$. The following three cases (i)–(iii) should be considered according to the rule applied in the last transition in $t \vdash^*_{\mathcal{A}_i} p$.

(i) If $t \vdash^*_{\mathcal{A}_i} z' \vdash_{\mathcal{A}_i} p$ $(z' \in \mathcal{Z}_i)$, then by Lemma 5 $z' \to p$ is constructed in Step 3 (a). Hence, there exists a rewrite rule $l \to r \in \mathcal{R}$ and a substitution $\rho: \mathcal{V} \to \mathcal{Z}_{i-1}$ satisfying the conditions 1 and 2 stated in Step 3. Since Step 3 (a) is applied, $r \in \mathcal{V}$. By Lemma 4, there exists a Σ-term s_j such that $s_j \vdash^*_{\mathcal{A}_{i-1}} x_j\rho$ $(1 \leq j \leq h)$. By Lemma 5 (i), we further consider two subcases, (i-a) $z' = p' \in \mathcal{P}$ and (i-b) $z' = [z, q]$ $(z \in \mathcal{Z}_{i-1}, q \in \mathcal{Q})$.
(i-a) If $t \vdash^*_{\mathcal{A}_i} p' \vdash_{\mathcal{A}_i} p$, then by the inductive hypothesis (A), there exists a Σ-term s' such that $s' \vdash^*_{\mathcal{A}_{i-1}} p'$ and $s' \to^*_{\mathcal{R}} t$. If $r = x_k$ for some k $(1 \leq k \leq h)$ then $p' = r\rho = x_k\rho$. Let $s = l[o_j \leftarrow s_j \mid 1 \leq j \leq h, j \neq k][o_k \leftarrow s']$. If $r \notin \mathcal{V}ar(l)$ then $p' = r\rho = p_{any}$. Let $s = l[o_j \leftarrow s_j \mid 1 \leq j \leq h]$. In either case, $s \to_{\mathcal{R}} s' \to^*_{\mathcal{R}} t$ and $s \vdash^*_{\mathcal{A}_{i-1}} l\rho \vdash^*_{\mathcal{A}_{i-1}} p$.
(i-b) If $t \vdash^*_{\mathcal{A}_i} [z, q] \vdash_{\mathcal{A}_i} p$, then $r \in \mathcal{V}ar(l)$ and $r\rho = x_k\rho = [z, q]$ for some k $(1 \leq k \leq h)$. By the inductive hypothesis (B), there exists a Σ-term s' such that $s' \vdash^*_{\mathcal{A}_{i-1}} z$ and $s' \to^*_{\mathcal{R}} q(t)$. Since $\mathcal{R}$ is an LT-TRS, the occurrence o_k of x_k in l can be

written as $o_k = o' \cdot 1$ for some o' and $l/o' = q(x_k)$. Let $s = l[o_j \leftarrow s_j \mid 1 \leq j \leq h, j \neq k][o' \leftarrow s']$. We can see that $s \rightarrow^*_{\mathcal{R}} l[o_j \leftarrow s_j \mid 1 \leq j \leq h, j \neq k][o' \leftarrow q(t)] \rightarrow_{\mathcal{R}} t$ and $s \vdash^*_{\mathcal{A}_{i-1}} l[o_j \leftarrow x_j\rho \mid 1 \leq j \leq h, j \neq k][o' \leftarrow z] \vdash^*_{\mathcal{A}_{i-1}} p$.

(ii) If $t = f(t_1, \ldots, t_n) \vdash^*_{\mathcal{A}_i} f(z_1, \ldots, z_n) \vdash_{\mathcal{A}_i} p$ ($f \in \mathcal{F}, z_j \in \mathcal{Z}_i$ $(1 \leq j \leq n)$) then $f(z_1, \ldots, z_n) \rightarrow p \in \Delta_0$ and thus $z_j \in \mathcal{P}$ by Lemma 2 (i). By the inductive hypothesis (A), there exists a Σ-term s_j such that $s_j \rightarrow^*_{\mathcal{R}} t_j$ and $s_j \vdash^*_{\mathcal{A}_{i-1}} z_j$ for $1 \leq j \leq n$. For $s = f(s_1, \ldots, s_n)$, $s \rightarrow^*_{\mathcal{R}} t$ and $s \vdash^*_{\mathcal{A}_{i-1}} p$ and the lemma holds.

(iii) If $t = q(t_1) \vdash^*_{\mathcal{A}_i} q(z_1) \vdash_{\mathcal{A}_i} p$ ($q \in \mathcal{Q}, z_1 \in \mathcal{Z}_i$) then $z_1 \in \mathcal{P}$ or $z_1 = [p, q]$. We can prove the lemma in a similar way to the case (ii), using the inductive hypothesis (A) when $z_1 \in \mathcal{P}$ and using the inductive hypothesis (B) when $z_1 = [p, q]$.

(B) Assume $t \vdash^*_{\mathcal{A}_i} [z, q]$. By Lemma 2, the right-hand side of a non-ε-transition rule constructed in Procedure 1 does not have the form of $[z, q]$. Also, by Lemma 5 (i), an ε-rule constructed in Step 3 (a) does not have the form of $[z, q]$. Hence the last rule applied in $t \vdash^*_{\mathcal{A}_i} [z, q]$ is an ε-rule constructed in Step 3 (b), and thus there exists a rule $l \rightarrow r \in \mathcal{R}$ and a substitution $\rho\colon \mathcal{V} \rightarrow \mathcal{Z}_{i-1}$ which satisfies the conditions 1 and 2 in Step 3. By Lemma 4, there exists a Σ-term s_j such that $s_j \vdash^*_{\mathcal{A}_{i-1}} x_j\rho$ for $1 \leq j \leq h$. Note that $r = q(r_1)$ for some $r_1 \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. There are three cases: (i) $t \vdash^*_{\mathcal{A}_i} p \vdash_{\mathcal{A}_i} [z, q]$ ($p \in \mathcal{P}$), (ii) $t \vdash^*_{\mathcal{A}_{i-1}} [z', q'] \vdash_{\mathcal{A}_{i-1}} [z, q]$ ($z' \in \mathcal{Z}_{i-1}, q' \in \mathcal{Q}$) and (iii) $t \vdash^*_{\mathcal{A}_i} \langle\tau\rho\rangle \vdash_{\mathcal{A}_i} [z, q]$ ($\tau \in \mathcal{T}(\mathcal{F}, \mathcal{V}), \rho\colon \mathcal{V} \rightarrow \mathcal{Z}_{i-1}$). In cases (i) and (ii), either $r_1 = x_k$ for some k $(1 \leq k \leq h)$ or $r_1 \in \mathcal{V}ar(r) \setminus \mathcal{V}ar(l)$.

(i) If $t \vdash^*_{\mathcal{A}_i} p \vdash_{\mathcal{A}_i} [z, q]$ ($p \in \mathcal{P}$), then by the inductive hypothesis (A), there exists a Σ-term s' such that $s' \vdash^*_{\mathcal{A}_{i-1}} p$ and $s' \rightarrow^*_{\mathcal{R}} t$. If $r_1 = x_k$ then $p = x_k\rho$. Let $s = l[o_j \leftarrow s_j \mid 1 \leq j \leq h, j \neq k][o_k \leftarrow s']$. If $r_1 \notin \mathcal{V}ar(l)$ then $p = p_{any}$. Let $s = l[o_j \leftarrow s_j \mid 1 \leq j \leq h]$. In either case, $s \rightarrow_{\mathcal{R}} q(s') \rightarrow^*_{\mathcal{R}} q(t)$ and $s \vdash^*_{\mathcal{A}_{i-1}} l\rho \vdash^*_{\mathcal{A}_{i-1}} z$.

(ii) If $t \vdash^*_{\mathcal{A}_{i-1}} [z', q'] \vdash_{\mathcal{A}_{i-1}} [z, q]$ ($z' \in \mathcal{Z}_{i-1}, q' \in \mathcal{Q}$), then by the inductive hypothesis (B), there exists a Σ-term s' such that $s' \vdash^*_{\mathcal{A}_{i-1}} z'$ and $s' \rightarrow^*_{\mathcal{R}} q'(t)$. The rest of the proof is similar to the proof of (A) (i-b).

(iii) If $t \vdash^*_{\mathcal{A}_i} \langle\tau\rho\rangle \vdash_{\mathcal{A}_i} [z, q]$, then by the inductive hypothesis (C), there exists a substitution $\sigma\colon \mathcal{V} \rightarrow \mathcal{T}(\Sigma)$ such that $\tau\sigma \rightarrow^*_{\mathcal{R}} t$ and $x\sigma \vdash^*_{\mathcal{A}_{i-1}} x\rho$ for $x \in \mathcal{V}ar(\tau)$. Note that $r = q(r_1) = q(\tau)$ in this case. Let $s = l\sigma$ then $s \rightarrow_{\mathcal{R}} r\sigma = q(\tau\sigma) \rightarrow^*_{\mathcal{R}} q(t)$ and $s \vdash^*_{\mathcal{A}_{i-1}} l\rho \vdash^*_{\mathcal{A}_{i-1}} z$ and the lemma holds.

(C) Assume $t \vdash^*_{\mathcal{A}_i} \langle\tau\rho\rangle$ for some $\tau \in \mathcal{T}(\Sigma, \mathcal{V})$ and substitution $\rho\colon \mathcal{V} \rightarrow \mathcal{Z}_{i-1}$. There are three cases to consider.

(i) Assume $t \vdash^*_{\mathcal{A}_i} z' \vdash_{\mathcal{A}_i} \langle\tau\rho\rangle$ ($z' \in \mathcal{Z}_i$). There are two subcases by Lemma 5 (i).

(i-a) If $t \vdash^*_{\mathcal{A}_i} p' \vdash_{\mathcal{A}_i} \langle\tau\rho\rangle$ ($p \in \mathcal{P}$), then by the inductive hypothesis (A), there exists a Σ-term s' such that $s' \rightarrow^*_{\mathcal{R}} t$ and $s' \vdash^*_{\mathcal{A}_{i-1}} p'$. The rule $p' \rightarrow \langle\tau\rho\rangle$ is introduced in Step 3 (a). Hence, there exists a rewrite rule $l \rightarrow r \in \mathcal{R}$, a state $z \in \mathcal{Z}_{i-1}$ and a substitution $\rho\colon \mathcal{V} \rightarrow \mathcal{Z}_{i-1}$ which satisfies the conditions in Step 3, $l\rho \vdash^*_{\mathcal{A}_{i-1}} \langle\tau\rho\rangle$ and $r \in \mathcal{V}$. By Lemma 4, there exists s_j such that $s_j \vdash^*_{\mathcal{A}_{i-1}} x_j\rho$ $(1 \leq j \leq h)$. If $r = x_k$ then $p' = r\rho = x_k\rho$. Let $s = l[o_j \leftarrow s_j \mid 1 \leq j \leq h, j \neq k][o_k \leftarrow s']$. If $r \notin \mathcal{V}ar(l)$ then $p' = p_{any}$. Let $s = l[o_j \leftarrow s_j \mid 1 \leq j \leq h]$. In either case, $s \rightarrow_{\mathcal{R}} s' \rightarrow^*_{\mathcal{R}} t$ and $s \vdash^*_{\mathcal{A}_{i-1}} l\rho \vdash^*_{\mathcal{A}_{i-1}} \langle\tau\rho\rangle$. By the inductive hypothesis (C), there exists a substitution $\sigma\colon \mathcal{V} \rightarrow \mathcal{T}(\Sigma)$ such that $\tau\sigma \rightarrow^*_{\mathcal{R}} s$ and $x\sigma \vdash^*_{\mathcal{A}_{i-1}} x\rho$ for $x \in \mathcal{V}ar(\tau)$. Hence, $\tau\sigma \rightarrow^*_{\mathcal{R}} s \rightarrow^*_{\mathcal{R}} t$ and the lemma holds.

(i-b) The case $t \vdash^*_{\mathcal{A}_i} [z, q] \vdash_{\mathcal{A}_i} \langle\tau\rho\rangle$ can be treated in a similar way to (i-a).

(ii) Assume $t = q(t') \vdash^*_{\mathcal{A}_i} q$ $([\langle\tau\rho\rangle, q]) \vdash_{\mathcal{A}_i} \langle\tau\rho\rangle$. Applying the inductive hypothesis (B) to $t' \vdash^*_{\mathcal{A}_i} [\langle\tau\rho\rangle, q]$, there exists a Σ-term s' such that $s' \rightarrow^*_{\mathcal{R}} q(t') = t$ and $s' \vdash^*_{\mathcal{A}_{i-1}} \langle\tau\rho\rangle$. By the inductive hypothesis (C), there exists a substitution $\sigma\colon \mathcal{V} \rightarrow \mathcal{T}(\Sigma)$ such that $\tau\sigma \rightarrow^*_{\mathcal{R}} s'$ and $x\sigma \vdash^*_{\mathcal{A}_{i-1}} x\rho$ for $x \in \mathcal{V}ar(\tau)$. Hence, $\tau\sigma \rightarrow^*_{\mathcal{R}} s' \rightarrow^*_{\mathcal{R}} t$ and the lemma holds.

(iii) Assume $t = f(t_1, \ldots, t_n) \vdash^*_{\mathcal{A}_i} f(\langle\tau_1\rho\rangle, \ldots, \langle\tau_n\rho\rangle) \vdash_{\mathcal{A}_i} \langle\tau\rho\rangle$ where $\tau = f(\tau_1, \ldots, \tau_n)$. By the inductive hypothesis (C), there exists a substitution $\sigma_j\colon \mathcal{V} \rightarrow \mathcal{T}(\Sigma)$ such that $\tau_j\sigma_j \rightarrow^*_{\mathcal{R}} t_j$ $(1 \leq j \leq n)$. Let $\sigma = \sigma_1\sigma_2 \cdots \sigma_n$. Note that $\mathrm{dom}(\sigma_j)$ is mutually disjoint since τ is linear. Clearly, $\tau\sigma = f(\tau_1, \ldots, \tau_n)\sigma = f(\tau_1\sigma_1, \ldots, \tau_n\sigma_n) \rightarrow^*_{\mathcal{R}} f(t_1, \ldots, t_n) = t$ and the lemma holds.

□

Lemma 7: (Completeness) If $s \rightarrow^*_{\mathcal{R}} t$ and $s \vdash^*_{\mathcal{A}_0} p \in \mathcal{P}$, then there exists an integer $i \geq 0$ such that $t \vdash^*_{\mathcal{A}_i} p$.

Proof. Assume that $s \rightarrow^*_{\mathcal{R}} t$ and $s \vdash^*_{\mathcal{A}_0} p$. The lemma is shown by induction on the number of rewrite steps in $s \rightarrow^*_{\mathcal{R}} t$. If $s = t$ then the lemma holds clearly. Assume $s \rightarrow^*_{\mathcal{R}} t' \rightarrow_{\mathcal{R}} t$. By the inductive hypothesis, there exists $i' \geq 0$ such that $t' \vdash^*_{\mathcal{A}_{i'}} p$. Consider a rewrite step $t' \rightarrow_{\mathcal{R}} t$. There exists a rewrite rule $l \rightarrow r \in \mathcal{R}$, a position o in t' and a substitution σ such that $t' = t'[o \leftarrow l\sigma]$ and $t = t'[o \leftarrow r\sigma]$. Assume $r = q(r_1)$ ($q \in \mathcal{Q}, r_1 \in \mathcal{T}(\Sigma, \mathcal{V})$). (The case when r is a variable is easier and omitted.) Since $t' = t'[o \leftarrow l\sigma]$, the transition sequence $t' \vdash^*_{\mathcal{A}_{i'}} p$ can be written as $t' = t'[o \leftarrow l\sigma] \vdash^*_{\mathcal{A}_{i'}} t'[o \leftarrow l\rho] \vdash^*_{\mathcal{A}_{i'}} t'[o \leftarrow z] \vdash^*_{\mathcal{A}_{i'}} p$ for some $z \in \mathcal{Z}_{i'}$ and $\rho\colon \mathcal{V} \rightarrow \mathcal{Z}_{i'}$. Since $l\rho \vdash^*_{\mathcal{A}_{i'}} z$, transition rules which enable $r_1\rho \vdash^*_{\mathcal{A}_{i'+1}} \langle r_1\rho\rangle \vdash^*_{\mathcal{A}_{i'+1}} [z, q]$ are added to $\Delta_{i'+1}$ in Step 3 of Procedure 1. Also $q([z, q]) \rightarrow z$ is added to $\Delta_{i'+1}$ and hence $t = t'[o \leftarrow r\sigma] = t'[o \leftarrow q(r_1\sigma)] \vdash^*_{\mathcal{A}_{i'+1}} t'[o \leftarrow q(r_1\rho)] \vdash^*_{\mathcal{A}_{i'+1}} t'[o \leftarrow$

$q([z,q])] \vdash_{\mathcal{A}_{i'+1}} t'[o \leftarrow z] \vdash^*_{\mathcal{A}_{i'+1}} p$ and the lemma holds. □

Lemma 8: (Partial Correctness) Let $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{final})$ be a TA without ε-rule and $\mathcal{R}$ be an LT-TRS. Assume that for input $\mathcal{A}$ and $\mathcal{R}$, Procedure 1 constructs a series of tree automata $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \cdots$. For any term $t \in \mathcal{T}(\Sigma)$ and state $p \in \mathcal{P}$,

> there exists a term $s \in \mathcal{T}(\Sigma)$ such that $s \vdash^*_{\mathcal{A}} p$ and $s \rightarrow^*_{\mathcal{R}} t$ if and only if there exists $i \geq 0$ such that $t \vdash^*_{\mathcal{A}_i} p$.

Proof. ($\Rightarrow$) By Lemma 7. ($\Leftarrow$) By induction on i and Lemma 6 (A). □

Lemma 9: If Procedure 1 halts for a TA $\mathcal{A}$ having no ε-rule and an LT-TRS $\mathcal{R}$, then $\mathcal{L}(\mathcal{A}_*) = (\rightarrow^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ holds for the output $\mathcal{A}_*$ of the procedure.

Proof. ($\subseteq$) Assume $t \in \mathcal{L}(\mathcal{A}_*)$. Since $\mathcal{A}_* = \mathcal{A}_i$ for some $i \geq 0$, there exists a final state p_f such that $t \vdash^*_{\mathcal{A}_i} p_f$. By Lemma 8, there exists a Σ-term s such that $s \vdash^*_{\mathcal{A}} p_f$ and $s \rightarrow^*_{\mathcal{R}} t$. Therefore, $t \in (\rightarrow^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$. $\mathcal{L}(\mathcal{A}_*) \supseteq (\rightarrow^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ can be shown in a similar way. □

5. Recognizability Preserving Property

In this section, two sufficient conditions for Procedure 1 to halt are proposed. One condition is that the sets of non-marker function symbols occurring in the left-hand sides and the right-hand sides of rewrite rules are disjoint. The other condition is the one which in effect restricts the class of recognizable sets. An LT-TRS $\mathcal{R}$ which satisfies the former condition effectively preserves recognizability. Although the latter condition does not directly give a subclass of LT-TRSs which are EPR, we can show that some properties of LT-TRSs are decidable by using results derived from the latter condition.

5.1 I/O-Separated LT-TRS

An LT-TRS $\mathcal{R}$ is *I/O-separated* if $\mathcal{R}$ satisfies the following condition.

Condition 1: For a signature $\Sigma = \mathcal{F} \cup \mathcal{Q}$, $\mathcal{F}$ is further divided as $\mathcal{F} = \mathcal{F}_I \cup \mathcal{F}_O$, $\mathcal{F}_I \cap \mathcal{F}_O = \emptyset$. A function symbol in $\mathcal{F}_I$ (respectively, $\mathcal{F}_O$) is called an *input symbol* (respectively, *output symbol*). Consider a rewrite rule

(i) $f(t_1, \cdots, t_n) \rightarrow r$, or
(ii) $t_1 \rightarrow r$

in $\mathcal{R}$ where $f, t_1, \ldots, t_n$ and r satisfy the conditions stated in Definition 1. Then $f \in \mathcal{F}_I$ and no input symbol appears in r. □

$\mathcal{R}_1$ in Example 1 and $\mathcal{R}_2$ in Example 2 are both I/O-separated LT-TRSs.

Lemma 10: If $q(z') \rightarrow z \in \Delta_i$ ($q \in \mathcal{Q}$) then either of $z \in \mathcal{P}$ or $z = \langle l \rangle$ for some rule $l \rightarrow r$ in $\mathcal{R}$ such that l is a ground term.

Proof. See Appendix. □

Lemma 11: Let $\mathcal{R}$ be an I/O-separated LT-TRS over $\Sigma = \mathcal{F}_I \cup \mathcal{F}_O \cup \mathcal{Q}$. If Procedure 1 is executed for a TA $\mathcal{A}$ and $\mathcal{R}$, then it always halts.

Proof. Assume a TA $\mathcal{A}$ and an I/O separated LT-TRS $\mathcal{R}$ are given to Procedure 1 as an input. A new state is introduced in Step 3 (b) or **ADDREC** and it is of the form $\langle t\rho \rangle$ or $[z,q]$ where $l \rightarrow r_1$, ρ and q satisfy condition (6) in Step 3, $r = q(r_1)$ and t is a subterm of r_1 (by Lemma 3). Hence, it is sufficient to show that the number of ρ and z which satisfy (6) is finite.

First, we show that the number of substitution ρ which satisfy (6) is finite. In a similar way to the proof of Lemma 5 (i), we can easily prove that for any substitution $\rho: \mathcal{V} \rightarrow \mathcal{Z}_{i-1}$ which satisfies (6), $x\rho$ ($x \in \mathcal{V}ar(l)$) is either in $\mathcal{P}$ or of the form $[z,q]$. If $x\rho = [z,q]$ then $x = x_k \in \mathcal{V}ar(l)$ and for the position o_k of x_k in l, $l/o' = q(x_k)$ where $o_k = o' \cdot 1$. Thus, $q([z,q]) \rightarrow z \in \Delta_i$ and by Lemma 10, the number of such substitutions ρ is finite.

We can also show there are only finite number of states z which satisfy (6) since such a state z is either in $\mathcal{P}$ or of the form $\langle \tau \rangle$ where τ is the left-hand side of a rule in $\mathcal{R}$. □

Theorem 3: Every I/O-separated LT-TRS effectively preserves recognizability. □

A bottom-up tree transducer [6] is a well-known computation model in the theory of tree languages. For a linear bottom-up tree transducer $\mathcal{M}$, if we consider the set of states of $\mathcal{M}$ as the set of markers, $\mathcal{M}$ corresponds to an I/O-separated LT-TRS. Hence, the following known property of tree transducer is obtained as a corollary.

Corollary 1: [6] Every linear bottom-up tree transducer effectively preserves recognizability. □

5.2 Marker-Bounded Sets

Let $\Sigma' \subseteq \Sigma$ be a subset of function symbols. Consider a tree representation of a term t. Let $\mathrm{depth}_{\Sigma'}(t)$ denote the maximum number of occurrences of function symbols in Σ' which occur in a single path from the root to a leaf in t. That is, $\mathrm{depth}_{\Sigma'}(t)$ is defined as:

$$\mathrm{depth}_{\Sigma'}(g(t_1, \cdots, t_n)) = \begin{cases} \max\{\mathrm{depth}_{\Sigma'}(t_i) \mid 1 \leq i \leq n\} + 1 & g \in \Sigma', \\ \max\{\mathrm{depth}_{\Sigma'}(t_i) \mid 1 \leq i \leq n\} & g \notin \Sigma'. \end{cases}$$

For example, for $\Sigma = \{f, g, h, c\}, \Sigma' = \{f, g\}$, $\mathrm{depth}_{\Sigma'}(f(g(c), g(h(g(c))))) = 3$.

For a signature $\Sigma = \mathcal{F} \cup \mathcal{Q}$, a set $T \subseteq \mathcal{T}(\Sigma)$ is *marker-bounded* if the following condition holds:

Condition 2: There exists $k \geq 0$ such that $|t|_{\mathcal{Q}} \leq k$ for each $t \in T$. □

An LT-TRS $\mathcal{R}$ is *simple* if every rule $l \to r$ in $\mathcal{R}$ satisfies the following conditions:

(1) l is not a ground term,
(2) r is not a variable, and
(3) $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$.

Lemma 12: Let $\mathcal{R}$ be an LT-TRS over $\Sigma = \mathcal{F} \cup \mathcal{Q}$ which satisfy conditions (1) and (3) in the above definition. If $t \to_{\mathcal{R}}^* t'$ then $\mathrm{depth}_{\mathcal{Q}}(t) \geq \mathrm{depth}_{\mathcal{Q}}(t')$.

Proof. The lemma can be easily proved by the form of a rewrite rule of an LT-TRS. □

Definition 2: (deg) For each state $z \in \mathcal{Z}_i$, let $\deg(z)$ denote the number of nestings in z, which is defined as follows:

$$\begin{cases} \deg(p) = 0 \ (p \in \mathcal{P}), \\ \deg([z,q]) = \deg(z) + 1 \ (z \in \mathcal{Z}_i, q \in \mathcal{Q}), \\ \deg(\langle f(t_1, \cdots, t_n)\rangle) \\ = \max\{\deg(\langle t_j \rangle) \mid 1 \leq j \leq n\} \end{cases}$$

□

By definition, $\deg(\langle f(t_1,\ldots,t_n)\rangle) = \max(\{0\} \cup \{\deg([z,q]) \mid [z,q] \text{ occurs in } f(t_1,\ldots,t_n)\})$

Lemma 13: (i) For $f(z_1,\ldots,z_n) \to z \in \Delta_i$ ($f \in \mathcal{F}, i \geq 0$), $\deg(z_j) \leq \deg(z)$ $(1 \leq j \leq n)$.

(ii) For $\langle r_1\rho\rangle \to [z,q] \in \Delta_i$ ($r_1 \in \mathcal{T}(\mathcal{F},\mathcal{V}), \rho{:}\, \mathcal{V} \to \mathcal{Z}_{i-1}, z \in \mathcal{Z}_{i-1}, q \in \mathcal{Q}$), $\deg(\langle r_1\rho\rangle) \leq \deg([z,q])$.

Proof. (i) If $i = 0$ then $\deg(z_j) = \deg(z) = 0$ $(1 \leq j \leq n)$ by definition. If $f(z_1,\ldots,z_n) \to z$ is added to Δ_i by **ADDREC**, then $z_j = \langle t_j\rho\rangle$ $(1 \leq j \leq n)$ and $z = \langle h(t_1,\ldots,t_n)\rho\rangle$ for some $h \in \mathcal{F}$ and $t_j \in \mathcal{T}(\Sigma,\mathcal{V})$ $(1 \leq j \leq n)$. Hence $\deg(z_j) \leq \deg(z)$ $(1 \leq j \leq n)$ by the definition of $\deg(\cdot)$.

(ii) Suppose that $\langle r_1\rho\rangle \to [z,q]$ is added to Δ_i in Step 3 (b). Then there exists a rewrite rule $l \to q(r_1)$ and a state $z \in \mathcal{Z}_{i-1}$ such that $l\rho \vdash_{\mathcal{A}_{i-1}}^* z$ where no ε-transition occurs at the root position or at any variable position of l. Assume that $l \to q(r_1)$ has the form of (i) in Definition 1, namely, $l = f(t_1,\ldots,t_n)$ where $f \in \mathcal{F}$ and each t_j is either a ground term or $t_j = q_j(l_j)$. (The case that $l \to q(r_1)$ has the form of (ii) in Definition 1 is easier and omitted.) If $\deg(\langle r_1\rho\rangle) = 0$ then the lemma holds clearly. If $\deg(\langle r_1\rho\rangle) \geq 1$, then a state $[z',q']$ with $\deg(z') = \deg(\langle r_1\rho\rangle) - 1$ occurs in $r_1\rho$. The transition sequence $l\rho \vdash_{\mathcal{A}_{i-1}}^* z$ can be written as $l\rho \vdash_{\mathcal{A}_{i-1}}^* f(z_1,\ldots,z_n) \vdash_{\mathcal{A}_{i-1}}^* z$. Since $[z',q']$ is not a subterm of r_1, there exists a variable $x \in \mathcal{V}ar(r_1)$ such that $[z',q']$ occurs in $x\rho$. Note that $x \in \mathcal{V}ar(l)$ since otherwise $x\rho = p_{any}$ by Step 3 of Procedure 1, which is a contradiction. Let $l = f(t_1,\ldots,g_m(x),\ldots,t_n)$. Remember that no ε-transition occurs at any variable position of l in $l\rho \vdash_{\mathcal{A}_{i-1}}^* z$. By Lemma 2(ii), $x\rho = [z_m,q_m]$ for some $z_m \in \mathcal{Z}_{i-1}$. On the other hand, $[z',q']$ is a subterm of $x\rho$ and hence $x\rho = [z_m,q_m] = [z',q']$ since $\deg([z_m,q_m]) = \deg(x\rho) \leq \deg(\langle r\rho\rangle) = \deg([z',q'])$. Since $f(z_1,\ldots,z_n) \to z \in \Delta_{i-1}$, $\deg(z_m) \leq \deg(z)$ by (i) of this lemma. Summarizing, $\deg(\langle r_1\rho\rangle) = \deg(z_m) + 1 \leq \deg(z) + 1 \leq \deg([z,q])$ and the lemma holds. □

Lemma 14: Procedure 1 always halts for a TA $\mathcal{A}$ having no ε-rule and an LT-TRS $\mathcal{R}$ which satisfy the following conditions.

(1) $\mathcal{L}(\mathcal{A})$ is marker-bounded.
(2) $\mathcal{R}$ is simple.

Proof. Let $\mathcal{A}$ and $\mathcal{R}$ be a TA and an LT-TRS which satisfy the conditions of the lemma. Without loss of generality, assume that $\mathcal{A} = (\Sigma, \mathcal{P}, \Delta, \mathcal{P}_{final})$ has only useful states. The proof is by contradiction. Assume that Procedure 1 does not halt for $\mathcal{A}$ and $\mathcal{R}$. We will show that there exists a term $t \in \mathcal{L}(\mathcal{A})$ which does not satisfy condition 2, which is a contradiction. By Lemma 3, a state constructed in Procedure 1 is of the form $\langle t\rho\rangle$ or $[z,q]$ where t is a subterm of the right-hand side of a rule in $\mathcal{R}$, $\rho{:}\, \mathcal{V} \to \mathcal{Z}_{i-1}$ is a substitution and $q \in \mathcal{Q}$. This implies that for an arbitrary integer v, the number of states z_0 with $\deg(z_0) < v$ constructed in the procedure and the number of rules which contain only such states are both finite. Since Procedure 1 does not halt, there exists an integer $i \geq 0$ and a state $z_0 \in \mathcal{Z}_i$ with $\deg(z_0) = k' \geq k+1$ where k is a constant in Condition 2. Note that $k' \leq i$ by the definition of $\deg(z_0)$ and the construction in Procedure 1. By Lemma 4, there exists a Σ-term s_0 such that $s_0 \vdash_{\mathcal{A}_i}^* z_0$. Since $\deg(z_0) \geq 1$, z_0 can be written as $z_0 = \langle \cdots [z_1', q_1'] \cdots \rangle$ ($= \langle \xi \rangle$, including the case that $z_0 = [z_1', q_1']$) and $\deg(z_0) = \deg([z_1', q_1'])$. The state $z_0 = \langle \xi \rangle$ is introduced in Step 3 (b) of Procedure 1 or **ADDREC** when the loop counter of the procedure is $i' \leq i$. Hence there exists a rewrite rule $l \to r \in \mathcal{R}$, a state $z_1' \in \mathcal{Z}_{i'-1}$ and a substitution $\rho{:}\, \mathcal{V} \to \mathcal{Z}_{i'-1}$ such that $l\rho \vdash_{\mathcal{A}_{i'-1}}^* z_1'$ and $r = q_1(r_1)$. By construction, $\langle r_1\rho\rangle \to [z_1', q_1]$ is added to $\Delta_{i'}$. By Lemma 13, $\deg(\langle r_1\rho\rangle) \leq \deg([z_1', q_1])$. By Lemma 3, ξ is a subterm of $r_1\rho$, and hence $k' = \deg(z_0)$ $= \deg([z_1', q_1']) \leq \deg(\langle r_1\rho\rangle)$. Hence, $\deg(z_1') \geq k' - 1$. Since $s_0 \vdash_{\mathcal{A}_i}^* z_0 = \langle \xi \rangle$ and ξ is a subterm of $r_1\rho$, there exists a Σ-term t_0 such that $t_0 \vdash_{\mathcal{A}_i}^* \langle r_1\rho\rangle \vdash_{\mathcal{A}_i} [z_1, q_1]$ and s_0 is a subterm of t_0. By Lemma 6 (B), there exists a Σ-term s_1 such that $s_1 \to_{\mathcal{R}}^* q_1(t_0)$ and $s_1 \vdash_{\mathcal{A}_{i-1}}^* z_1$. Summarizing, $s_1 \vdash_{\mathcal{A}_{i-1}}^* z_1$, $s_1 \to_{\mathcal{R}}^* q_1(t_0)$, $\deg(z_1) \geq k' - 1$, and s_0 is a subterm of t_0. Repeating the above argument, we can see that there exist states $z_j \in \mathcal{Z}_{i-j}$ $(1 \leq j \leq i)$, markers $q_j \in \mathcal{Q}$ $(1 \leq j \leq i)$, Σ-terms s_j $(0 \leq j \leq i)$, t_j $(0 \leq j \leq i-1)$ such that:

$$\begin{array}{l} s_j \vdash_{\mathcal{A}_{i-j}}^* z_j, s_j \to_{\mathcal{R}}^* q_j(t_{j-1}), \deg(z_j) \geq k' - j \\ \text{and } s_{j-1} \text{ is a subterm of } t_{j-1} \quad (1 \leq j \leq i). \end{array} \tag{7}$$

Since $s_j \to_{\mathcal{R}}^* q_j(t_{j-1})$ $(1 \leq j \leq i)$, $\mathrm{depth}_{\mathcal{Q}}(s_j) \geq \mathrm{depth}_{\mathcal{Q}}(t_{j-1}) + 1$ by Lemma 12. Since s_j is a subterm of t_j $(0 \leq j \leq i-1)$, $\text{depth}_{\mathcal{Q}}(t_j) \geq \text{depth}_{\mathcal{Q}}(s_j)$. Hence, $\text{depth}_{\mathcal{Q}}(s_i) \geq \text{depth}_{\mathcal{Q}}(s_0) + i \geq i \geq k'$. Since z_i is useful by Lemma 4, there exists a Σ-term t', a position $o \in \mathcal{P}os(t')$ and a final state $p_f \in \mathcal{P}_{\mathit{final}}$ such that

$$t' \vdash_{\mathcal{A}}^{*} t'[o \leftarrow z_i] \vdash_{\mathcal{A}}^{*} p_f. \tag{8}$$

Let $t = t'[o \leftarrow s_i]$, then $t \vdash_{\mathcal{A}}^{*} t'[o \leftarrow z_i] \vdash_{\mathcal{A}}^{*} p_f \in \mathcal{P}_{\mathit{final}}$ by (7) and (8). Thus, $t \in \mathcal{L}(\mathcal{A})$ holds. Furthermore, $\text{depth}_{\mathcal{Q}}(t) \geq \text{depth}_{\mathcal{Q}}(s_i) \geq k' \geq k+1$. This conflicts with Condition 2. Therefore, Procedure 1 halts. □

Theorem 4: For any TA $\mathcal{A}$ and an LT-TRS $\mathcal{R}$ which satisfy conditions (1) and (2) of Lemma 14, $(\rightarrow_{\mathcal{R}}^{*})(\mathcal{L}(\mathcal{A}))$ is recognizable.

Proof. By Lemmas 9 and 14. □

Example 6: Let $\mathcal{F} = \{add, s', 0\}$ and $\mathcal{Q} = \{s\}$. The following TRS $\mathcal{R}_4$ is a simple LT-TRS.

$$\mathcal{R}_4 = \begin{cases} add(s(x), s(y)) \rightarrow s(s'(add(x,y))) \\ add(s(x), 0) \rightarrow s(x) \\ add(0, s(y)) \rightarrow s(y) \end{cases}$$

Let $\mathcal{R}_5 = \{s'(x) \rightarrow s(x), add(0,0) \rightarrow 0\}$, then the relation $\rightarrow_{\mathcal{R}_4}^{*} \cdot \rightarrow_{\mathcal{R}_5}^{*}$ defines addition on natural numbers. Since we can easily see that $\mathcal{R}_5$ is an EPR-TRS [13], for any TA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ is marker-bounded, $(\rightarrow_{\mathcal{R}_5}^{*})((\rightarrow_{\mathcal{R}_4}^{*})(\mathcal{L}(\mathcal{A})))$ is always recognizable. □

$\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ in the examples 1 through 4 are also simple LT-TRSs. As mentioned in Sect. 2.3, a TRS in Réty's subclass [12] is $\mathcal{C}$-EPR. The subclass of TRSs defined in [12] and the subclass of simple LT-TRSs are incomparable. In fact, any non-left-linear TRS in the former class is not an LT-TRS. On the other hand, $\{f(q(x)) \rightarrow q(f(f(x)))\}$ belongs to the latter class but does not belong to the former class. Also the class of marker-bounded sets and $\mathcal{C}$ are incomparable. For example, consider $\mathcal{R}_3$ of Example 5. $\mathcal{L}(\mathcal{A}) = \{f(g^n(c)) \mid n \geq 0\}$ is not marker-bounded but $\mathcal{R}_3$ belongs to Réty's subclass and $\mathcal{L}(\mathcal{A})$ belongs to $\mathcal{C}$.

Corollary 2: For a finite set T of ground terms and a simple LT-TRS $\mathcal{R}$, $(\rightarrow_{\mathcal{R}}^{*})(T)$ is recognizable. □

Corollary 3: For a simple LT-TRS $\mathcal{R}$, reachability and joinability are decidable.

Proof. The reachability problem is to decide whether for a given TRS $\mathcal{R}$ and Σ-terms s and t, $s \rightarrow_{\mathcal{R}}^{*} t$ holds or not. It is obvious that $s \rightarrow_{\mathcal{R}}^{*} t$ if and only if $t \in (\rightarrow_{\mathcal{R}}^{*})(\{s\})$. The latter condition is decidable by Lemma 1 and Corollary 2.

Decidability of joinability can easily be verified by noting that $\exists w\colon s \rightarrow_{\mathcal{R}}^{*} w$ and $t \rightarrow_{\mathcal{R}}^{*} w$ if and only if $(\rightarrow_{\mathcal{R}}^{*})(\{s\}) \cap (\rightarrow_{\mathcal{R}}^{*})(\{t\}) \neq \emptyset$. □

6. Conclusion

In this paper, a new subclass of TRSs called LT-TRSs is defined and a sufficient condition for an LT-TRS to effectively preserve recognizability is provided. The subclass of LT-TRSs satisfying the condition contains simple EPR-TRSs which do not belong to any of the known decidable subclasses of EPR-TRSs.

Extending the proposed class is a future study. For example, Procedure 1 could be extended by packed state technique used in [14] so that Procedure 1 is sound even if the left-linearity condition is dropped for an input LT-TRS. Finding a more general sufficient condition on a TA $\mathcal{A}$ to satisfy that $(\rightarrow_{\mathcal{R}}^{*})(\mathcal{L}(\mathcal{A}))$ is recognizable for any LT-TRS $\mathcal{R}$ is another interesting question.

Acknowledgments

The authors would like to thank Dr. Yoshiki Kinoshita and Dr. Hitoshi Ohsaki of National Institute of Advanced Industrial Science and Technology for their valuable discussions and insightful comments.

References

[1] F. Baader and T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.
[2] W.S. Brainerd, "Tree generating regular systems," Inform. and Control, vol.14, pp.217–231, 1969.
[3] J.L. Coquidé, M. Dauchet, R. Gilleron, and S. Vágvölgyi, "Bottom-up tree pushdown automata: Classification and connection with rewrite systems," Theoretical Computer Science, vol.127, pp.69–98, 1994.
[4] I. Durand and A. Middeldorp, "Decidable call by need computations in term rewriting (extended abstract)," Proc. of CADE-14, LNAI 1249, pp.4–18, North Queensland, Australia, 1997.
[5] T. Genet, "Decidable approximations of sets of descendants and sets of normal forms," Proc. RTS98, LNCS 1379, pp.151–165, Tsukuba, Japan, 1998.
[6] F. Gécseg and M. Steinby, Tree Automata, Académiai Kiadó, 1984.
[7] R. Gilleron, "Decision problems for term rewriting systems and recognizable tree languages," Proc. STACS'91, LNCS 480, pp.148–159, Hamburg, Germany, 1991.
[8] R. Gilleron and S. Tison, "Regular tree languages and rewrite systems," Fundamenta Informaticae, vol.24, pp.157–175, 1995.
[9] P. Gyenizse and S. Vágvölgyi, "Linear generalized semi-monadic rewrite systems effectively preserve recognizability," Theoretical Computer Science, vol.194, pp.87–122, 1998.
[10] F. Jacquemard, "Decidable approximations of term rewriting systems," Proc. RTA96, LNCS 1103, pp.362–376, New Brunswick, NJ, 1996.
[11] T. Nagaya and Y. Toyama, "Decidability for left-linear growing term rewriting systems," Proc. RTA99, LNCS 1631, pp.256–270, Trento, Italy, 1999.
[12] P. Réty, "Regular sets of descendants for constructor-based rewrite systems," Proc. LPAR'99, LNCS 1705, pp.148–160, Tbilisi, Georgia, 1999.

[13] K. Salomaa, "Deterministic tree pushdown automata and monadic tree rewriting systems," J. Comput. System Sci., vol.37, pp.367–394, 1988.

[14] T. Takai, Y. Kaji, and H. Seki, "Right-linear finite path overlapping term rewriting systems effectively preserve recognizability," Proc. RTA2000, LNCS, vol.1833, pp.246–260, Norwich, U.K., 2000.

Appendix: Proof of the Lemmas

A.1 Proof of Lemma 4

Assume that every state $p \in \mathcal{P}$ is useful and show that every state $z \in \mathcal{Z}_i$ is useful in $\mathcal{A}_i$ by induction on i (A proof for reachable states is easier and omitted). The basis case is obvious. Assume that Step 3 is executed for a rule $l \to q(r_1)$, state $z \in \mathcal{Z}_{i-1}$ and substitution $\rho: \mathcal{V} \to \mathcal{Z}_{i-1}$ and states $\langle r_1\rho \rangle$, $[z, q]$ and some states of the form $\langle t\rho \rangle$ are constructed. By Lemma 3, t is a subterm of r_1. We show all these new states are useful. By the inductive hypothesis, z is useful and hence there exists a Σ-term t, a position $o \in \mathcal{P}os(t)$ and a final state $p_f \in \mathcal{P}_{\mathit{final}}$ such that

$$t \vdash^*_{\mathcal{A}_{i-1}} t[o \leftarrow z] \vdash^*_{\mathcal{A}_{i-1}} p_f. \qquad (\text{A}\cdot 1)$$

Let $\mathcal{V}ar(l) = \{x_1, \ldots, x_n\}$. By the inductive hypothesis, $x_j\rho$ is useful and thus reachable in $\mathcal{A}_{i-1}$, and hence there exists a Σ-term s_j such that $s_j \vdash^*_{\mathcal{A}_{i-1}} x_j\rho$ for $1 \le j \le h$. Let $\sigma = \{x_j \mapsto s_j \mid 1 \le j \le h\}$, then

$$q(r_1\sigma) \vdash^*_{\mathcal{A}_i} q(\langle r_1\rho \rangle) \vdash_{\mathcal{A}_i} q([z, q]) \vdash z. \qquad (\text{A}\cdot 2)$$

By (A·1) and (A·2),

$$t[o \leftarrow q(r_1\sigma)] \vdash^*_{\mathcal{A}_i} t[o \leftarrow z] \vdash^*_{\mathcal{A}_{i-1}} p_f. \qquad (\text{A}\cdot 3)$$

All the new states appear in (A·3) and thus they are useful. □

A.2 Proof of Lemma 5

(i) Consider the condition (6) $l\rho \vdash^*_{\mathcal{A}_{i-1}} z$ in Step 3. An ε-rule added in Step 3 (a) is of the form $r\rho \to z$ where $r \in \mathcal{V}$. Note that since $\mathcal{R}$ is an LT-TRS, for any variable position o_j in l, o_j is written as $o_j = o'_j \cdot 1$ and $l/o'_j = q_j(x_j)$ where $q_j \in \mathcal{Q}$. Since no ε-transition occurs at any variable position o_j in (6), each $x_j\rho$ (especially, $r\rho$) is either in $\mathcal{P}$ or of the form $[z', q']$ by Lemma 2 (ii). Similarly, since no ε-transition occurs at the root position in (6), z is either in $\mathcal{P}$ or of the form $\langle \tau\rho \rangle$ by Lemma 2 (i) and (ii).

(ii) Obvious from Step 3 (b) of Procedure 1. □

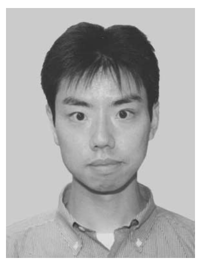
A.3 Proof of Lemma 10

We prove the lemma by induction on i. If $i = 0$ then the lemma holds clearly. Suppose that $q([z, q]) \to z$ is added to Δ_i in Step 3 (b). Then there exists a rewrite rule $l \to r$, a state $z \in \mathcal{Z}_{i-1}$ and a substitution $\rho: \mathcal{V} \to \mathcal{Z}_{i-1}$ satisfying $l\rho \vdash^*_{\mathcal{A}_{i-1}} z$. Since $\mathcal{R}$ is an I/O-separated LT-TRS, (i) $l = f(t_1, \ldots, t_n)$ ($f \in \mathcal{F}_I$) or (ii) $l = t_1$ where each t_j is a ground term or $t_j = q_j(l_j)$ such that $q_j \in \mathcal{Q}$ and l_j is a variable or a ground term.

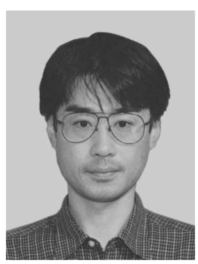
(i) If $l = f(t_1, \ldots, t_n)$ then the transition sequence $l\rho \vdash^*_{\mathcal{A}_{i-1}} z$ can be written as $l\rho \vdash^*_{\mathcal{A}_{i-1}} f(z_1, \ldots, z_n) \vdash_{\mathcal{A}_{i-1}} z$. Since $f \in \mathcal{F}_I$, by the discussion before the above claim, $f(z_1, \ldots, z_n) \to z \in \Delta_0$ and thus $z \in \mathcal{P}$.

(ii) If l is a ground term then for the sequence $l\rho (= l) \vdash^*_{\mathcal{A}_{i-1}} z$, we can see $z \in \mathcal{P}$ or $z = \langle l \rangle$. If $l = q_1(l_1)$ ($q_1 \in \mathcal{Q}$) then $l\rho \vdash^*_{\mathcal{A}_{i-1}} z$ can be written as $l\rho \vdash^*_{\mathcal{A}_0} q_1(p') \vdash_{\mathcal{A}_{i-1}} z$ ($p' \in \mathcal{P}$) or $l\rho \vdash^*_{\mathcal{A}_{i-1}} q_1([z, q_1]) \vdash_{\mathcal{A}_{i-1}} z$. In the former case, $z \in \mathcal{P}$. In the latter case, by the induction hypothesis, $z \in \mathcal{Z}$ or $z = \langle \tau \rangle$ where τ is the left-hand side of a rule in $\mathcal{R}$.

Thus the lemma holds in every case. □



Toshinori Takai received his B.E. degree in information science from Kyushu Institute of Technology, Fukuoka, Japan, in 1996, and received M.E. and D.E. degrees in information science from Nara Institute of Science and Technology, in 1998 and 2002, respectively. Now he is a research associate of National Institute of Advanced Industrial Science and Technology.



Hiroyuki Seki received B.E. and Ph. D. degrees in information and computer sciences from Osaka University, Japan, in 1982 and 1987, respectively. He was with Osaka University as an Assistant Professor in 1990–1992 and an Assoicate Professor in 1992–1994. In 1994, he joined the faculty of Nara Institute of Science and Technology, where he has been a Professor since 1996.

Youhei Fujinaka received his B.E. degree in information science from Shimane University, Matsue, Japan, in 2000, and received M.E. degree in information science from Nara Institute of Science and Technology, in 2002. Now he is a teacher of Izumo high school and teaching mathematics.

Yuichi Kaji received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1991, 1992 and 1994, respectively. In 1994, he joined Nara Institute of Science and Technology, Nara, Japan. His current research interests include the theory of error correcting codes, information security, and the theory of automata and rewriting systems.