Decidability of Reachability for Term Rewriting with Reduction Strategies

Yoshiharu Kojima

Abstract

The reachability problem for a given set of initial term, a goal term, and a term rewriting system (TRS) is to decide whether the goal one is reachable from the initial one by the reduction of the TRS or not. The innermost reduction is the strategy that rewrites innermost redexes, the outermost reduction is the one that rewrites outermost redexes, and the context-sensitive reduction is the one in which rewritable positions are indicated by specifying arguments of function symbols. In this paper, we show some classes such that the reachability problem is decidable or undecidable with respect to the innermost reduction, the outermost reduction, the context-sensitive reduction, or the context-sensitive innermost reduction. We used the common approach based on the tree automata technique for proving decidability of reachability. We show algorithms to construct a tree automaton recognizing the set of terms reachable from a given term by reductions of TRSs in some classes with respect to the given strategy. We prove undecidability of reachability by reducing the Post correspondence problem, which is well-known as an undecidable problem, to reachability problems.

Moreover, we formalize controlled term rewriting systems (CntTRS), which are combinations of term rewriting systems with constraints to select possible rewrite positions, and we show some properties for them. The constraints are specified, for each rewrite rule, by a selection automaton which defines a set of positions in a term based on tree automata computations. We show complexity results on reachability and the regular model checking problem, which is the intersection emptiness problem of the rewrite closure of a regular tree language and another regular tree language.

Acknowledgement

I am deeply grateful to Professor Masahiko Sakai whose comments and suggestions were very valuable throughout the course of my study. I would also like to thank Frolent Jaqcuemard at INRIA, Professor Toshiki Sakabe, Associate Professor Keiichirou Kusakari and Assistant Professor Naoki Nishida whose comments made great contribution to my work. I also thank Mutsunori Yagiura for his helpful advises.

Contents

Chapter 1

Introduction

A term rewriting system (TRS) is a set of rewrite rules of terms, and mainly used as a model of a functional program. The reachability problem for two given terms s, t and a reduction of a term rewrite system (TRS) R with respect to a strategy is to decide whether t is reachable from s by the reduction of R with respect to the strategy. Procedures to solve reachability problems of term rewriting systems are applicable to security protocol verification [11] and solving other problems of TRSs (e.g. joinability, termination, etc.). Since it is known that this problem is undecidable in general, efforts have been made to find subclasses of TRSs in which the reachability is decidable or undecidable [7, 8, 18, 17, 24, 26, 27, 32, 33, 34, 35] as shown in Figure 1.1.

Reduction strategies, which specify rewritable position of reductions of TRSs, are used to express computation of programs more precisely. The innermost reduction is the strategy that rewrites innermost redexes, and is a model of call-by-value computation. The outermost reduction is the strategy that rewrites outermost redexes, and is a model of call-by-name. The context-sensitive reduction [23] is the strategy that specifies rewritable arguments of each function symbol by a mapping μ: if $i \notin \mu(f)$ then reductions at ith argument of f are forbidden. The context-sensitive reduction is used in evaluating `if` $\cdots$ `then` $\cdots$ `else` $\cdots$ or `case` structures. The context-sensitive innermost reduction is the combination of innermost and context-sensitive.

Since reachability problems for each strategy are different as the following Example 1, algorithms to solve reachability for the ordinary reduction may not

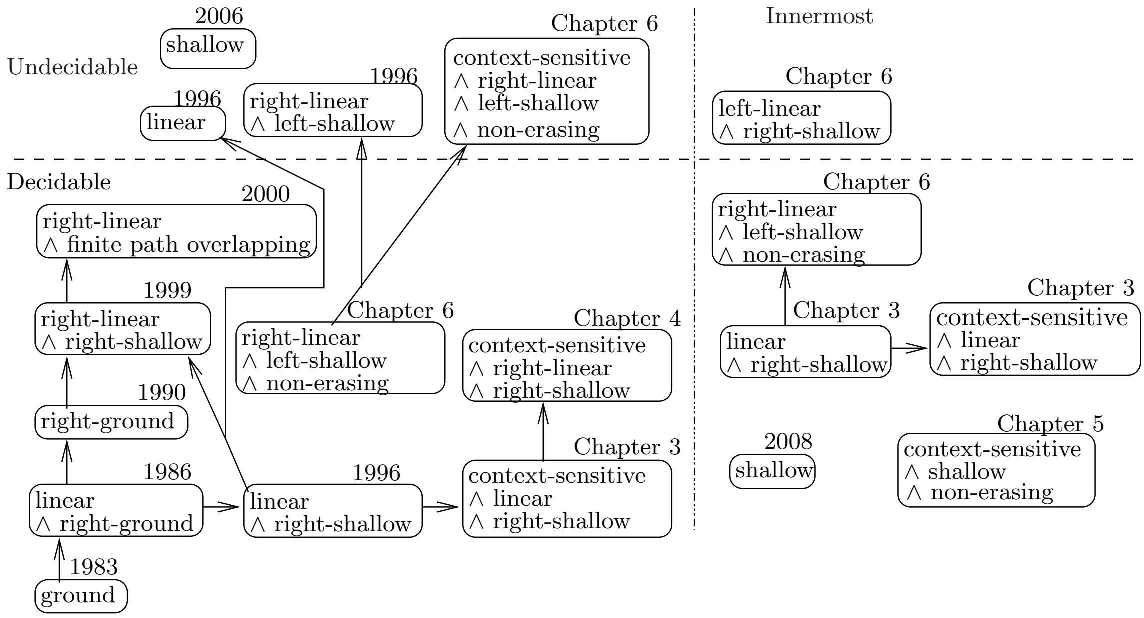

Figure 1.1: Major subclasses of TRSs in which the reachability is decidable or undecidable with respect to the ordinary reduction, the innermost reduction, the context-sensitive reduction, and the context-sensitive innermost reduction.

be applied to problems with respect to other strategies.

Example 1 Let the TRS R be $R = \{a \to b, f(x) \to g(x), h(x) \to i(x)\}$ and the replacement map μ be $\mu(f) = \emptyset$ and $\mu(g) = \mu(h) = \mu(i) = \{1\}$. The set of reachable terms from $f(h(a))$ for each strategy is as follows:

$$
\begin{array}{rcl}
\text{Ordinary reduction} & : & \left\{ \begin{array}{c} f(h(a)), f(h(b)), f(i(a)), f(i(b)), \\ g(h(a)), g(h(b)), g(i(a)), g(i(b)) \end{array} \right\}, \\
\text{Innermost reduction} & : & \{f(h(a)), f(h(b)), f(i(b)), g(i(b))\}, \\
\text{Outermost reduction} & : & \{f(h(a)), g(h(a)), g(i(a)), g(i(b))\}, \\
\text{Context-sensitive reduction} & : & \{f(h(a)), g(h(a)), g(h(b)), g(i(a)), g(i(b))\}, \\
\begin{array}{l}\text{Context-sensitive} \\ \text{innermost reduction}\end{array} & : & \{f(h(a)), g(h(a)), g(h(b)), g(i(b))\}.
\end{array}
$$

$\square$

It is known that reachability problems with respect to the innermost reduction, the outermost reduction, and the context-sensitive reduction are decidable for several classes [10, 13, 30]. However, there are few classes for which it is known that reachability is decidable or undecidable with respect to the strategies.

In this paper, we show decidability or undecidability of reachability for linear and right-shallow TRSs, right-linear and right-shallow TRSs, shallow and non-erasing TRSs, right-linear, left-shallow, and non-erasing TRSs, and left-linear and right-shallow TRSs with respect to the innermost reduction, the outermost reduction, the context-sensitive reduction, and the context-sensitive innermost reduction.

We show the results on decidability in this paper by presenting algorithms that construct a tree automaton recognizing the set of terms reachable to or reachable from a given term by the reduction of a given TRS. Results on undecidability are proved by reducing the Post correspondence problems (PCP), which is well-known as an undecidable problem, to reachability problems.

Moreover, in this paper, we study the so called *controlled term rewriting systems* (CntTRS). Such a system is defined by a combination of term rewriting rules with some constraints (called control) specifying possible rewrite positions.

In order to define the constraints, we have chosen a similar model to the selection automata (SA) of [9], which, intuitively, select positions in a term based on the computations of a tree automaton. These constraints give powerful selection mechanisms, with the same expressive power as the monadic second order logic of the tree, or monadic Datalog [14]. We also consider a restriction of the SA where a position p in a term t is selected if the sequence of symbols on the path from p to the root of t belongs to a given regular language. The corresponding restricted rewriting model is called prefix CntTRS (pCntTRS).

We show complexity results on the reachability problem and the regular model checking problem, which is the intersection emptiness problem of a rewrite closure of a regular tree language and another regular tree language, for some classes of CntTRSs and pCntTRSs.

Overview In Chapter 3, we show that the reachability problems are decidable for linear and right-shallow TRSs with respect to the context-sensitive reduction or the context-sensitive innermost reduction. We show the results by modifying the algorithm in [18] that constructs a tree automaton recognizing the set of reachable terms from the set of terms recognized by an input tree automaton for linear and right-shallow TRSs with respect to the ordinary reduction. In the case of the context-sensitive reduction, we need information on whether a term is at a rewritable position or not. Therefore, we augment a parameter to each state of input automata that shows accepted terms are at a rewritable position or not. In the case of the context-sensitive innermost reduction, we also need information on whether each argument of a term is a normal form or not. Therefore, we augment a state of the automata that recognizes normal forms to each state of input automata.

In Chapter 4, we show that the reachability is decidable for right-linear and right-shallow TRSs with respect to the context-sensitive reduction. Since the algorithm in Chapter 3 is insufficient for left-non-linear TRSs, we employ the idea used in [24, 32], which shows the decidability of reachability for left-non-linear TRSs.

In Chapter 5, we show that the reachability is decidable for shallow and non-erasing TRSs with respect to the context-sensitive innermost reduction. We mod-

Table 1.1: Decidability and undecidability of the reachability problem.

Type of reduction	Right-linear $\wedge$ left-shallow	Right-linear $\wedge$ left-shallow $\wedge$ non-erasing	Left-linear $\wedge$ right-shallow
Ordinary	Undecidable [18, 35]	Decidable	Decidable
Innermost	Undecidable	Decidable	Undecidable
Outermost	Undecidable	Undecidable	Still open
Context-sensitive	Undecidable	Undecidable	Undecidable

ify the algorithm in Chapter 4 similarly to the modification from the algorithm for linear right-shallow TRSs with respect to the context-sensitive reduction to the one with respect to the context-sensitive innermost reduction.

In Chapter 6, we show decidability/undecidability results in Table 1. The proofs for decidability with respect to the ordinary reduction are by showing a similar algorithm to the one in [18] rather than the input of the algorithm. The proof for decidability with respect to the innermost reduction is by modifying the one for the ordinary reduction. The proofs for undecidability are reduction from PCP to reachability problems.

In Chapter 7, we formalize CntTRSs and pCntTRSs, and show complexity results on reachability problems and regular model-checking problems for CntTRSs and pCntTRSs. In the first section, we give a definition of selection automata used as constraints for each rewrite rule, and a definition of rewrite relations of CntTRSs. In the second section, we introduce tree grammars in context-free or context-sensitive setting. These are useful to prove decidability or undecidability in the third section. In the third section, we prove complexity for reachability and model-checking problems for some classes. In fact, model-checking problems for all classes introduced in the third section are undecidable. Therefore, in the fourth section, we introduce recursive pCntTRSs, which are in restricted form of CntTRSs, and show that model-checking problems for recursive pCntTRSs are decidable in EXPTIME.

Chapter 2

Preliminaries

We use usual notations of term rewriting system [1], context-sensitivity [23] and tree automaton [5]. Let F be a set of function symbols with a fixed arity and X be an enumerable set of variables. The arity of function symbol f is denoted by $\mathrm{ar}(f)$. Function symbols with $\mathrm{ar}(f) = 0$ are *constants*. The set of *terms*, defined in the usual way, is denoted by $\mathcal{T}(F, X)$. A term is *linear* if no variable occurs more than once in the term. The set of variables occurring in t is denoted by $\mathrm{Var}(t)$. A term t is *ground* if $\mathrm{Var}(t) = \emptyset$. The set of ground terms is denoted by $\mathcal{T}(F)$.

A *position* in a term t is defined, as usual, as a sequence of positive integers, and the set of all positions in a term t is denoted by $\mathrm{Pos}(t)$, where the empty sequence ε is used to denote the root position. If a position p is represented as $p'p''$, then p' is a *prefix* of p. The *depth* of a position p is defined as $|p|$. The *height* $|t|$ of a term t is defined as $\max(\{|p| \mid p \in \mathrm{Pos}(t)\})$. A term t is *shallow* if depths of variable occurrences in t are all 0 or 1. The *subterm* of t at position p is denoted by $t|_p$, and $t[t']_p$ represents the term obtained from t by replacing the subterm $t|_p$ by t'. If a term s is a subterm of t, it is denoted by $s \trianglelefteq t$, and if $s \trianglelefteq t$ and $s \neq t$, s is a *proper* subterm of t. A *context* C is a term that contains the symbol $\Box$, and $C[t]_p$ represents the term obtained by replacing $\Box$ in the position p of C by t.

A *substitution* σ is a mapping from X to $\mathcal{T}(F, X)$ whose domain $\mathrm{Dom}(\sigma) = \{x \in X \mid x \neq \sigma(x)\}$ is finite. The term obtained by applying a substitution σ to a term t is written as $t\sigma$. The term $t\sigma$ is an *instance* of t.

A *rewrite rule* is an ordered pair of terms in $\mathcal{T}(F, X)$, written as $l \to r$, such that $l \notin X$ and $\mathrm{Var}(l) \supseteq \mathrm{Var}(r)$. A *term rewriting system (over F)* (TRS) is a finite set of rewrite rules. *The ordinary reduction* $\xrightarrow[R]{}$ of a TRS R is as follows: $s \xrightarrow[R]{} t$ if and only if $s = s[l\sigma]_p$, and $t = s[r\sigma]_p$ for some rule $l \to r \in R$, with a substitution σ and a position $p \in \mathrm{Pos}(s)$. We call $l\sigma$ a *redex*. We sometimes write $\xrightarrow[R]{}^p$ by presenting the position p explicitly.

A rewrite rule $l \to r$ is *left-linear* (resp. *right-linear*, *linear*, *right-shallow*) if l is linear (resp. r is linear, l and r are linear, r is shallow). A TRS R is *left-linear* (resp. *right-linear*, *linear*, *right-shallow*) if every rule in R is left-linear (resp. right-linear, linear, right-shallow). A rewrite rule $l \to r$ is *collapsing* if r is a variable.

Let $\to$ be a binary relation on a set $\mathcal{T}(F)$. We say $s \in \mathcal{T}(F)$ is a *normal form* (with respect to $\to$) if there exists no term $t \in \mathcal{T}(F)$ such that $s \to t$. We use $\circ$ to denote the composition of two relations. We write $\leftarrow$ for symmetric closure, $\xrightarrow{*}$ for the reflexive and transitive closure of $\to$, respectively. We also write $\xrightarrow{n}$ for the relation $\to \circ \cdots \circ \to$ composed of n $\to$'s. The set of *reachable terms* from a term in T is defined by $\to[T] = \{t \mid s \in T,\ s \xrightarrow{*} t\}$. The *reachability problem* (resp. *joinability problem*) with respect to $\to$ is the problem that decides whether $s \xrightarrow{*} s'$ (resp. $s \xrightarrow{*} \circ \xleftarrow{*} s'$) or not, for given terms s and s'.

A context-sensitive reduction is a subrelation of the ordinary reduction in which rewritable positions are indicated by specifying arguments of function symbols. A mapping $\mu :\ F \to \mathcal{P}(\mathbb{N})$ is said to be a *replacement map* if $\mu(f) \subseteq \{1, \ldots, \mathrm{ar}(f)\}$ for all $f \in F$. The set of *μ-replacing positions* $\mathrm{Pos}^\mu(t)$ ($\subseteq \mathrm{Pos}(t)$) is recursively defined: $\mathrm{Pos}^\mu(t) = \{\varepsilon\}$ if t is a constant or a variable, otherwise $\mathrm{Pos}^\mu(f(t_1, \ldots, t_n)) = \{\varepsilon\} \cup \{ip \mid i \in \mu(f), p \in \mathrm{Pos}^\mu(t_i)\}$. The context-sensitive reduction of a TRS R and a replacement map μ is formally defined: $s \xhookrightarrow[R]{}^{\mu} t$ if and only if $s \xrightarrow[R]{}^p t$ for some $p \in \mathrm{Pos}^\mu(s)$.

A reduction $s \xrightarrow[R]{}^p t$ is *innermost* if all proper subterms of $s|_p$ are normal forms and is *outermost* if there is no subterm $s|_{p'}$ such that $s|_{p'}$ is a redex and p' is a prefix of p. If all proper subterms are normal forms with respect to the context-sensitive reduction or not in $\mathrm{Pos}^\mu(s)$, we say that the step is *context-sensitive innermost*. We denote innermost reductions of the TRS R by $\xrightarrow[R]{}_{\mathrm{in}}$, outermost reductions by $\xrightarrow[R]{}_{\mathrm{out}}$, and context-sensitive innermost reductions by

$\xhookrightarrow[R]{\mu}_{\text{in}}$. We denote the set of normal forms of R as NF_R and normal forms of R and μ with respect to the context-sensitive reduction as NF_R^μ.

A *tree automaton* (TA) is a quadruple $\mathcal{A} = \langle F, Q, Q^f, \Delta \rangle$ where Q is a finite set of states, $Q^f (\subseteq Q)$ is a set of final states, and Δ is a finite set of transition rules of the forms $f(q_1, \ldots, q_n) \to q$ or $q_1 \to q$ where $f \in F$ with $\mathrm{ar}(f) = n$, and $q_1, \ldots, q_n, q \in Q$. We sometimes omit F if it is not necessary to specify it. We can regard Δ as a (ground) TRS over $F \cup Q$. The rewrite relation induced by Δ is called a *transition relation* denoted by $\xrightarrow[\Delta]{}$. We denote $|\alpha|$ as the length of a transition sequence α (if α is $s \xrightarrow{n} t$, then $|\alpha| = n$). We say that a term s $(\in \mathcal{T}(F))$ is *accepted* by $\mathcal{A}$ if $s \xrightarrow[\Delta]{*} q \in Q^f$. The set of all terms accepted by $\mathcal{A}$ is denoted by $\mathcal{L}(\mathcal{A})$. We say $\mathcal{A}$ *recognizes* $\mathcal{L}(\mathcal{A})$. A set of terms T is *regular* if there exists a TA that recognizes T. We use a notation $\mathcal{L}(\mathcal{A}, q)$ or $\mathcal{L}(\Delta, q)$ to represent the set $\{s \mid s \xrightarrow[\Delta]{*} q\}$. A TA $\mathcal{A}$ is *reduced* if for all states $q \in Q$, there exists a ground term $t \in \mathcal{T}(F)$ such that $t \xrightarrow[\Delta]{*} q$. A TA $\mathcal{A}$ is *deterministic* if $s \xrightarrow[\Delta]{*} q$ and $s \xrightarrow[\Delta]{*} q'$ implies $q = q'$ for any $s \in \mathcal{T}(F)$. A TA $\mathcal{A}$ is *complete* if there exists $q \in Q$ such that $s \xrightarrow[\Delta]{*} q$ for any $s \in \mathcal{T}(F)$.

A *run* of $\mathcal{A}$ on a term $t \in \mathcal{T}(\mathcal{F})$ is a mapping ρ from $\mathrm{Pos}(t)$ into $Q_{\mathcal{A}}$ such that for every $p \in \mathrm{Pos}(t)$, $t(p)\big(\rho(p.1), \ldots, \rho(p.n)\big) \to \rho(p)$ is in $\Delta_{\mathcal{A}}$, where n is the arity of the symbol $t(p)$ in $\mathcal{F}$. The run ρ is called *successful* (or *accepting*) if $\rho(\varepsilon)$ is in $F_{\mathcal{A}}$. The set of successful runs of $\mathcal{A}$ on t is denoted $\mathit{sruns}(\mathcal{A}, t)$. For the sake of conciseness, we shall sometimes apply term-like notations (subterm, replacement…) to runs.

The following properties on TA are known.

Theorem 2 ([5]) *1. For a given regular set $T \subset \mathcal{T}(F)$, we can construct a deterministic, complete, and reduced TA $\mathcal{A} = (F, Q, Q^f, \Delta)$ that recognizes T.*

2. The class of regular sets on $\mathcal{T}(F)$ is closed under union, intersection, and complementation.

3. The membership problem and the emptiness problem are decidable.

Regular Model checking (RMC) is a problem to decide, given two TA languages L_{in} and L_{err} and a TRS R whether $\xrightarrow[R]{}[L_{\mathsf{in}}] \cap L_{\mathsf{err}} = \emptyset$. The name of the problem is coined after state exploration techniques for checking safety properties. In this setting, L_{in} and L_{err} represent (possibly infinite) sets of initial, respectively error, states.

A relation $\to$ is effectively preserves regularity if for a given TA $\mathcal{A}$ we can construct a tree automaton $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = \to[\mathcal{L}(\mathcal{A})]$. If the rewrite relation of a TRS R effectively preserves regularity, we say that R effectively preserves regularity. If TRS R effectively preserves regularity, some problem of R is decidable.

Theorem 3 ([12]) *If TRS R effectively preserves regularity, then the reachability problem for R is decidable.*

Chapter 3

Decidability of Reachability for Linear and Right-Shallow TRSs

In this chapter, we show the results on decidability of reachability for linear and right-shallow TRSs. With respect to the ordinary reduction, it was shown that reachability for linear and right-shallow TRSs is decidable by [18]. However, it has been an open problem whether reachability with respect to other rewrite strategies is decidable or not. In the following, we show that reachability is decidable with respect to the context-sensitive, the innermost, and the context-sensitive innermost reduction for linear and right-shallow TRSs.

3.1 On the Context-Sensitive Reduction

This section shows that the context-sensitive reduction of a linear and right-shallow TRS R and a replacement map μ effectively preserves regularity, and thus reachability is decidable if the TRS R is linear and right-shallow. In order to show the property, we introduce an algorithm $\mathrm{P}_{\mathrm{cs}}^{\mathrm{lrs}}$ that inputs a TA $\mathcal{A}$, a TRS R, and a replacement map μ, and outputs a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = \xhookrightarrow[R]{}^{\mu}[\mathcal{L}(\mathcal{A})]$, i.e. $\mathcal{A}_*$ recognizes the set of terms reachable from the terms accepted by $\mathcal{A}$. By this algorithm $\mathrm{P}_{\mathrm{cs}}^{\mathrm{lrs}}$, it turns out that the relation $\xhookrightarrow[R]{}^{\mu}$ effectively preserves regularity.

The main idea employed in $\mathrm{P}_{\mathrm{cs}}^{\mathrm{lrs}}$ is to divide each state q of input automata to $\langle q, \mathtt{a}\rangle$ and $\langle q, \mathtt{i}\rangle$. $\mathtt{a}$ is the initial of "active". $\mathtt{i}$ is the initial of "inactive". The state $\langle q, \mathtt{a}\rangle$ basically occurs at rewritable positions, and hence used for accepting terms in $\underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A}, q)]$, i.e. accepting the set of terms reachable from terms accepted by q. The state $\langle q, \mathtt{i}\rangle$ occurs at unrewritable positions and hence keeps accepting terms only in $\mathcal{L}(\mathcal{A}, q)$. Note that there are the cases that $\langle q, \mathtt{a}\rangle$ occurs at unrewritable positions. We note the reason after introducing $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$

We define $\mathrm{RS}(R)$ as the set of the ground subterms of all right-hand-sides of TRS R. This is used to construct the TA $\mathcal{A}_{\mathrm{RS}}$ that recognizes $\mathrm{RS}(R)$ in $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$.

Definition 4 For the TRS R, $\mathrm{RS}(R)$ is the set $\{t_1, \dots, t_n \mid l \to f(t_1, \dots, t_n) \in R\}$.

We show the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ in the following. In the first step of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$, we initialize the input TA $\mathcal{A}$ by adding the transition rules of $\mathcal{A}_{\mathrm{RS}}$ and dividing each state q in $\mathcal{A}$ to $\langle q, \mathtt{a}\rangle$ and $\langle q, \mathtt{i}\rangle$. In the second step, we augment transition rules of $\mathcal{A}$ to accept reachable terms by the inference rules. The idea of the inference rules is that for a rewrite rule $l \to r$ and a substitution θ from variables to states, if $l\theta$ is accepted by a state of the form $\langle q, \mathtt{a}\rangle$, then we add transition rules to accept $r\theta$. Even if $l\theta$ is accepted by a state of the form $\langle q, \mathtt{i}\rangle$, no transition rule is added. Finally, if no rule is produced by the inference rules, we output the TA.

Algorithm $\mathbf{P}^{\mathrm{lrs}}_{\mathrm{cs}}$

Input TA $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$, a linear and right-shallow TRS R, and a replacement map μ .

Output TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_*\rangle$ such that $\mathcal{L}(\mathcal{A}_*) = \underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A})]$.

Step 1 (initialize) 1. Prepare a TA $\mathcal{A}_{\mathrm{RS}} = \langle Q_{\mathrm{RS}}, Q^f_{\mathrm{RS}}, \Delta_{\mathrm{RS}}\rangle$ that recognizes $\mathrm{RS}(R)$. Here we assume $Q = \{q^s \mid s \trianglelefteq t, t \in \mathrm{RS}(R)\}$, $Q^f_{\mathrm{RS}} = \{q^t \mid t \in \mathrm{RS}(R)\}$ and $\Delta_{\mathrm{RS}} = \{f(q^{t_1}, \dots, q^{t_n}) \to q^{f(t_1, \dots, t_n)} \mid f(t_1, \dots, t_n) \trianglelefteq t, t \in \mathrm{RS}(R)\}$.

2. Let

- $k := 0$,

- $Q_* = (Q \uplus Q_{\mathrm{RS}}) \times \{\mathtt{a}, \mathtt{i}\}$,
- $Q_*^f = Q^f \times \{\mathtt{a}\}$
- Δ_0 consists of the following transition rules:
  - $\langle q', \mathtt{x}\rangle \to \langle q, \mathtt{x}\rangle$ where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$ and $q' \to q \in \Delta \cup \Delta_{\mathrm{RS}}$,
  - $f(\langle q_1, \mathtt{i}\rangle, \dots, \langle q_n, \mathtt{i}\rangle) \to \langle q, \mathtt{i}\rangle$ where $f(q_1, \dots, q_n) \to q \in \Delta \cup \Delta_{\mathrm{RS}}$, and
  - $f(\langle q_1, \mathtt{x}\rangle, \dots, \langle q_n, \mathtt{x}\rangle) \to \langle q, \mathtt{a}\rangle$ where $f(q_1, \dots, q_n) \to q \in \Delta \cup \Delta_{\mathrm{RS}}$, $\mathtt{x}_i = \mathtt{a}$ if $i \in \mu(f)$, and $\mathtt{x}_i = \mathtt{i}$ otherwise.

Step 2 Let Δ_{k+1} be the set of transition rules produced by augmenting transition rules of Δ_k by the following inference rules:

$$\frac{f(l_1, \dots, l_n) \to g(r_1, \dots, r_m) \in R,\ f(\langle q_1, \mathtt{x}_1\rangle, \dots, \langle q_n, \mathtt{x}_n\rangle) \to \langle q, \mathtt{a}\rangle \in \Delta_k}{g(\langle q'_1, \mathtt{x}'_1\rangle, \dots, \langle q'_m, \mathtt{x}'_m\rangle) \to \langle q, \mathtt{a}\rangle \in \Delta_{k+1}}$$

if there exists $\theta : X \to Q_*$ such that $l_i\theta \xrightarrow[\Delta_k]{*} \langle q_i, \mathtt{x}_i\rangle$ for all $1 \le i \le n$ and $\mathcal{L}(\Delta_k, x\theta) \neq \emptyset$ for all $x \in \mathrm{Var}(f(l_1, \dots, l_n))$. Let $r_j\theta = \langle q''_j, \mathtt{x}''_j\rangle$ for every $r_j \in X$. Each $\langle q'_j, \mathtt{x}'_j\rangle$ is determined as

- $q'_j = \begin{cases} q''_j & \cdots \text{ if } r_j \in X \\ q^{r_j} & \cdots \text{ if } r_j \notin X \end{cases}$
- $\mathtt{x}'_j = \begin{cases} \mathtt{x}''_j & \cdots \text{ if } j \notin \mu(g) \wedge r_j \in X \\ \mathtt{i} & \cdots \text{ if } j \notin \mu(g) \wedge r_j \notin X \\ \mathtt{a} & \cdots \text{ if } j \in \mu(g) \end{cases}$

for all $1 \le j \le m$, and

$$\frac{f(l_1, \dots, l_n) \to x \in R,\ f(\langle q_1, \mathtt{x}_1\rangle, \dots, \langle q_n, \mathtt{x}_n\rangle) \to \langle q, \mathtt{a}\rangle \in \Delta_k}{\langle q', \mathtt{a}\rangle \to \langle q, \mathtt{a}\rangle \in \Delta_{k+1}}$$

if there exists $\theta : X \to Q_*$ such that $l_i\theta \xrightarrow[\Delta_k]{*} \langle q_i, \mathtt{x}_i\rangle$ for all $1 \le i \le n$ and $\mathcal{L}(\Delta_k, x\theta) \neq \emptyset$ for all $x \in \mathrm{Var}(f(l_1, \dots, l_n))$. If $\langle q'', \mathtt{x}\rangle = x\theta$, then $q' = q''$.

Step 3 If $\Delta_{k+1} = \Delta_k$ then stop and set $\Delta_* = \Delta_k$; Otherwise $k := k + 1$ and go to step 2.

Example 5 Let us follow how the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ works. Let

$$R = \{a \to b, f(x) \to g(x)\},\ \mu(f) = \emptyset,\ \mu(g) = \{1\},$$

and $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ be a TA as:

$$Q = \{q^a, q^{f(a)}\},\ Q^f = \{q^{f(a)}\},\ \text{and } \Delta = \{a \to q^a, f(q^a) \to q^{f(a)}\}$$

and hence $\mathcal{L}(\mathcal{A}) = \{f(a)\}$.

In the initializing step, we have

$$\begin{aligned}&\Delta_{\mathrm{RS}} = \emptyset,\ Q_* = \{\langle q^a, \mathtt{x}\rangle, \langle q^{f(a)}, \mathtt{x}\rangle\},\ Q^f_* = \{\langle q^{f(a)}, \mathtt{a}\rangle\}, \text{ and}\\ &\Delta_0 = \{a \to \langle q^a, \mathtt{x}\rangle, f(\langle q^a, \mathtt{i}\rangle) \to \langle q^{f(a)}, \mathtt{x}\rangle\}\end{aligned}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$.

The saturation steps stop at $k = 1$. In this step, we have

$$\Delta_1 = \Delta_0 \cup \{b \to \langle q^a, \mathtt{a}\rangle, g(\langle q^a, \mathtt{a}\rangle) \to \langle q^{f(a)}, \mathtt{a}\rangle\}$$

and $\Delta_2 = \Delta_1$. $b \to \langle q^a, \mathtt{a}\rangle \in \Delta_1$ is produced by $a \to b \in R$ and $a \to \langle q^a, \mathtt{a}\rangle \in \Delta_0$. $g(\langle q^a, \mathtt{a}\rangle) \to \langle q^{f(a)}, \mathtt{a}\rangle \in \Delta_1$ is produced by $f(x) \to g(x) \in R$ and $f(\langle q^a, \mathtt{i}\rangle) \to \langle q^{f(a)}, \mathtt{a}\rangle \in \Delta_0$.

Finally, we have

$$\Delta_* = \Delta_1 = \{a \to \langle q^a, \mathtt{x}\rangle, f(\langle q^a, \mathtt{i}\rangle) \to \langle q^{f(a)}, \mathtt{x}\rangle, b \to \langle q^a, \mathtt{a}\rangle, g(\langle q^a, \mathtt{a}\rangle) \to \langle q^{f(a)}, \mathtt{a}\rangle\}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$.

The TA $\mathcal{A}_*$ recognizes the set of terms $\{f(a), g(a), g(b)\} = \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu} [\mathcal{L}(\mathcal{A})]$, and does not accept $f(b)$.

$\square$

If there is a rewrite rule where there is a variable that occurs at a rewrite position of the left-hand-side and at a unrewritable position of the right-hand-side, then states that have $\mathtt{a}$ occur at unrewritable position as Example 6. However, no problem occur.

Example 6 Let $R = \{a \to b, f(x) \to g(x)\}$, $\mu(f) = \{1\}, \mu(g) = \emptyset$, and consider the TA $\mathcal{A}$ in Example 5. Here, we have $\overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}[\mathcal{L}(\mathcal{A})] = \{f(a), f(b), g(a), g(b)\}$.

In this case, $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ outputs the TA $\mathcal{A}_*$ such that

$$\Delta_* = \left\{a \to \langle q^a, \mathtt{x}\rangle, b \to \langle q^a, \mathtt{a}\rangle, f(\langle q^a, \mathtt{x}\rangle) \to \langle q^{f(a)}, \mathtt{x}\rangle, g(\langle q^a, \mathtt{a}\rangle) \to \langle q^{f(a)}, \mathtt{a}\rangle\right\}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$, and hence $\mathcal{L}(\mathcal{A}_*) = \underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A})]$. □

In Example 6, the state $\langle q^a, \mathtt{a}\rangle$ occurs at an unrewritable position since $\mu(g) = \emptyset$. If the rule $g(\langle q^a, \mathtt{i}\rangle) \to \langle q^{f(a)}, \mathtt{a}\rangle$ is added instead of $g(\langle q^a, \mathtt{a}\rangle) \to \langle q^{f(a)}, \mathtt{a}\rangle$, TA is not output correctly since $\langle q^a, \mathtt{i}\rangle$ only accepts a and hence $g(b)$ is not accepted.

The algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ eventually terminates at some k, because rewrite rules in R and states Q_* are finite and hence possible transitions rules are finite. Apparently $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$.

To show the correctness of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$, we show several technical propositions and lemmas.

Proposition 7 *Let $s \in \mathcal{T}(F)$, then $s \xrightarrow[\Delta]{*} q$ for $q \in Q$ or $s \xrightarrow[\Delta_{\mathrm{RS}}]{*} q \in Q_{\mathrm{RS}}$ if and only if $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}\rangle$.*

Proof: Direct consequence of the construction of Δ_0. □

Proposition 8 *Let $t \in \mathcal{T}(F)$. $t \xrightarrow[\Delta_0]{*} \langle q^{t'}, \mathtt{i}\rangle \in Q_{\mathrm{RS}} \times \{\mathtt{i}\}$ iff $t = t'$.*

Proof: From the construction of Δ_{RS}, we have $t \xrightarrow[\Delta_{\mathrm{RS}}]{*} q^{t'}$ iff $t = t'$. Therefore, this proposition holds from Proposition 7. □

Proposition 9 *Let $t \in \mathcal{T}(F)$. For any k, if $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{i}\rangle \in Q_*$, then $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}\rangle \in Q_*$.*

Proof: The proposition follows from the fact that $\Delta_0 \subseteq \Delta_k$ and transition rules having $\mathtt{i}$ on the second component of right-hand sides are in Δ_0. □

Proposition 10 *Let $t \in \mathcal{T}(F)$. $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{a}\rangle \in Q_*$ if and only if $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}\rangle \in Q_*$.*

Proof: Direct consequence of the construction of Δ_0. □

Proposition 11 *Let $t \in \mathcal{T}(F)$. For any k, if $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{i} \rangle \in Q_*$, then $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{a} \rangle$.*

Proof: Let $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{i} \rangle$, then $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i} \rangle$ by Proposition 9. The lemma follows from Proposition 10 and the fact $\Delta_0 \subseteq \Delta_k$. □

The following lemmas 12 and 13 are used to prove the completeness of $\mathrm{P}_{\mathrm{cs}}^{\mathrm{lrs}}$. Since these lemmas are similarly proved in the rest of chapters and sections, we divide them from the key lemma of the proof of the completeness.

Lemma 12 *If $\alpha : t[t']_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$ and $p \in \mathrm{Pos}^\mu(t)$, then there exists $\langle q', \mathtt{a} \rangle$ such that $t' \xrightarrow[\Delta_*]{*} \langle q', \mathtt{a} \rangle$ and $t[\langle q', \mathtt{a} \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$.*

Proof: We show the lemma by induction on $|\alpha|$. Let $p \in \mathrm{Pos}^\mu(t)$.

1. If α is represented as $t[t']_p \xrightarrow[\Delta_*]{*} \langle q'', \mathtt{a} \rangle \xrightarrow[\Delta_*]{} \langle q, a \rangle$, then this lemma holds from the induction hypothesis.

2. Consider the case that the last transition rule applied in α is represented as $t[t']_p = f(\ldots, t_{i-1}, t_i[t']_{p'}, t_{i+1}, \ldots) \xrightarrow[\Delta_*]{*} f(\ldots, \langle q_{i-1}, \mathtt{x}_{i-1} \rangle, \langle q_i, \mathtt{x}_i \rangle, \langle q_{i+1}, \mathtt{x}_{i+1} \rangle, \ldots) \xrightarrow[\Delta_*]{} \langle q, \mathtt{a} \rangle$. Since $ip' = p \in \mathrm{Pos}^\mu(t)$, we have $i \in \mu(f)$. Hence $\mathtt{x}_i = \mathtt{a}$ follows from the construction of Δ_*.

   By the induction hypothesis, there exists $\langle q', \mathtt{a} \rangle$ such that $t' \xrightarrow[\Delta_*]{*} \langle q', \mathtt{a} \rangle$ and $t_i[\langle q', \mathtt{a} \rangle]_{p'} \xrightarrow[\Delta_*]{*} \langle q_i, \mathtt{a} \rangle$. Here we have $t[\langle q', \mathtt{a} \rangle]_p = f(\ldots, t_{i-1}, t_i[\langle q', \mathtt{a} \rangle]_{p'}, t_{i+1}, \ldots) \xrightarrow[\Delta_*]{*} f(\ldots, \langle q_{i-1}, \mathtt{x}_{i-1} \rangle, \langle q_i, \mathtt{a} \rangle, \langle q_{i+1}, \mathtt{x}_{i+1} \rangle, \ldots) \xrightarrow[\Delta_*]{} \langle q, \mathtt{a} \rangle$. □

Lemma 13 *Let R be a linear and right-shallow TRS and $l \to g(r_1, \ldots, r_m) \in R$. For $r_j \notin X$ and any substitution σ, $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q^{r_j}, \mathtt{i} \rangle$ and $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q^{r_j}, \mathtt{a} \rangle$.*

Proof: Let $r_j \in X$. Since R is right-shallow, $r_j\sigma = r_j$. From Proposition 8, we have $r_j \xrightarrow[\Delta_0]{*} \langle q^{r_j}, \mathtt{i} \rangle$. Thus, we also have $r_j \xrightarrow[\Delta_*]{*} \langle q^{r_j}, \mathtt{i} \rangle$ from $\Delta_0 \subseteq \Delta_*$.
Moreover, we have $r_j \xrightarrow[\Delta_*]{*} \langle q^{r_j}, \mathtt{a} \rangle$ from Proposition 11. □

The following lemma is obtained from the above propositions and lemmas.

Lemma 14 *If R is a linear and right-shallow TRS, then $s \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$ and $s \xhookrightarrow[R]{*}{}^{\mu} t$ imply $t \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$.*

Proof: We present the proof in the case of $s \xhookrightarrow[R]{}{}^{\mu} t$ because the proof in the case of $s = t$ is trivial and in the case of $s \xhookrightarrow[R]{*}{}^{\mu} t' \xhookrightarrow[R]{}{}^{\mu} t$, we can prove by applying the proof for $s \xhookrightarrow[R]{}{}^{\mu} t$ repeatedly. Let $s \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$ and $s = s[l\sigma]_p \xhookrightarrow[R]{}{}^{\mu} s[r\sigma]_p = t$ for some k and rewrite rule $l \to r \in R$ where $p \in \mathrm{Pos}^{\mu}(s)$. We have a transition sequence $s \xrightarrow[\Delta_*]{*} s[\langle q', \mathtt{a} \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$ by Lemma 12.

1. Consider the case where the rewrite rule is of the form $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m)$. We show the diagram of this proof at Figure 3.1.

   Since the rewrite rule is left-linear, $s \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$ is represented as $s = s[f(l_1, \ldots, l_n)\sigma]_p \xrightarrow[\Delta_*]{*} s[f(l_1, \ldots, l_n)\theta]_p \xrightarrow[\Delta_*]{*} s[f(\langle q_1, \mathtt{x}_1 \rangle, \ldots, \langle q_n, \mathtt{x}_n \rangle)]_p \xrightarrow[\Delta_*]{} s[\langle q', \mathtt{x}' \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$ for some $\theta : X \to Q_*$. Here we have $l_i\theta \xrightarrow[\Delta_k]{*} \langle q_i, \mathtt{x}_i \rangle$ and $\mathcal{L}(\Delta_k, x\theta) \neq \emptyset$ for all $x \in \mathrm{Var}(f(l_1, \ldots, l_n))$. Let $r_j\theta = \langle q''_j, \mathtt{x}''_j \rangle$ for every $r_j \in X$. From the construction of Δ_*, there exists a transition rule $g(\langle q'_1, \mathtt{x}'_1 \rangle, \ldots, \langle q'_n, \mathtt{x}'_n \rangle) \to \langle q', \mathtt{a} \rangle \in \Delta_*$ such that

   - $q'_j = \begin{cases} q''_j & \cdots \text{ if } r_j \in X \\ q^{r_j} & \cdots \text{ if } r_j \notin X \end{cases}$
   - $\mathtt{x}'_j = \begin{cases} \mathtt{x}''_j & \cdots \text{ if } j \notin \mu(g) \wedge r_j \in X \\ \mathtt{i} & \cdots \text{ if } j \notin \mu(g) \wedge r_j \notin X \\ \mathtt{a} & \cdots \text{ if } j \in \mu(g) \end{cases}$

   for all $1 \leq j \leq m$. Here, we have $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j \rangle$ for $r_j \notin X$ from $q'_j = q^{r_j}$ and Lemma 13. For j such that $r_j \in X$, we have $r_j\sigma \xrightarrow[\Delta_*]{*} r_j\theta = \langle q''_j, \mathtt{x}''_j \rangle = \langle q'_j, \mathtt{x}''_j \rangle$. Here $\mathtt{x}'_j = \mathtt{x}''_j$ or $\mathtt{x}'_j = \mathtt{a}$. If $\mathtt{x}''_j = \mathtt{i}$, then we also have $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q''_j, \mathtt{a} \rangle$ from Proposition 11. Hence, we also have $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j \rangle$ in either case of $\mathtt{x}''_j = \mathtt{a}$ or $\mathtt{x}''_j = \mathtt{i}$ for $r_j \in X$.

   Thus, we have $t = s[g(r_1, \ldots, r_m)\sigma]_p \xrightarrow[\Delta_*]{*} s[g(\langle q'_1, \mathtt{x}'_1 \rangle, \ldots, \langle q'_m, \mathtt{x}'_m \rangle)]_p \xrightarrow[\Delta_*]{} s[\langle q', \mathtt{a} \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a} \rangle$.

2. In the case where the rewrite rule is in the form $f(l_1, \ldots, l_n) \to x$, we can show the lemma similarly to the previous case. $\square$

$$\begin{tikzcd}
s[f(l_1,\ldots,l_n)\theta]_p \arrow[r, "*", "\Delta_*"'] & s[f(\langle q_1,\mathtt{x}_1\rangle,\ldots,\langle q_n,\mathtt{x}_n\rangle)]_p \arrow[d, "*"', "\Delta_*"] & \\
s = s[f(l_1,\ldots,l_n)\sigma]_p \arrow[u, "*", "\Delta_*"'] \arrow[d, hook, "R", "\mu"'] & s[\langle q',\mathtt{a}\rangle]_p \arrow[r, "*", "\Delta_*"'] & \langle q,\mathtt{a}\rangle \\
t = s[g(r_1,\ldots,r_m)\sigma]_p \arrow[r, "*", "\Delta_*"'] & s[g(\langle q'_1,\mathtt{x}'_1\rangle,\ldots,\langle q'_m,\mathtt{x}'_m\rangle)]_p \arrow[u, "*", "\Delta_*"'] &
\end{tikzcd}$$

Figure 3.1: The diagram of proof of lemma 14

Lemma 15 *If R is a linear and right-shallow TRS, then $\mathcal{L}(\mathcal{A}_*) \supseteq \xhookrightarrow[R]{}^{\mu}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let $s \xhookrightarrow[R]{*}^{\mu} t$ and $s \xrightarrow[\Delta]{*} q \in Q^f$. Since $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}\rangle$ from Proposition 9, we have $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{a}\rangle$ by Proposition 11. Thus, we obtain $t \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}\rangle \in Q^f_*$ by lemma 14. □

Before proving the soundness of the procedure $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$, we define a measurement of transitions.

A measurement of transitions of Δ_* is defined as $||s \xrightarrow[\Delta_0]{} t|| = 0$ and $||s \xrightarrow[\Delta_{i+1}\setminus\Delta_i]{} t|| = i+1$ for $i \geq 0$. This is extended on transition sequences as a multiset:

$$||s_0 \xrightarrow[\Delta_*]{} s_1 \xrightarrow[\Delta_*]{} \cdots \xrightarrow[\Delta_*]{} s_{n+1}|| = \{||s_i \xrightarrow[\Delta_*]{} s_{i+1}|| \mid 0 \leq i < n\}.$$

Now we can define the following order $\sqsupset$ on transition sequences by Δ_*, which is necessary in proofs:

$$\alpha \sqsupset \beta \overset{\mathrm{def}}{\Leftrightarrow} ||\alpha|| >_{\mathrm{mul}} ||\beta||$$

where $>_{\mathrm{mul}}$ is the multiset extension of $>$ on $\mathbb{N}$.

Lemma 16 *If R is a linear and right-shallow TRS, then $\alpha :\ t \xrightarrow[\Delta_*]{*} \langle q, \mathtt{x}\rangle$ implies $s \xhookrightarrow[R]{*}^{\mu} t$ and $\beta :\ s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{x}\rangle$ for some term s.*

Proof: If $\mathtt{x} = \mathtt{i}$, then this lemma holds from Proposition 9. In the following, we show this lemma for $\mathtt{x} = \mathtt{a}$ by induction on $||\alpha||$ with respect to $\sqsupset$.

1. Let $t = g(t_1, \ldots, t_m)$ and consider the case where the last transition rule applied in α is of the form $g(\langle q'_1, \mathtt{x}'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m\rangle) \to \langle q, \mathtt{a}\rangle \in \Delta_k$.

   (a) If the last transition rule applied at α is in Δ_0, then α is represented as $t = g(t_1, \ldots, t_m) \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1\rangle, \ldots, \langle q'_n, \mathtt{x}'_n\rangle) \xrightarrow[\Delta_0]{} \langle q, \mathtt{a}\rangle$. For j such that $\mathtt{x}'_j = \mathtt{i}$, we have $t_j \xrightarrow[\Delta_0]{*} \langle q'_j, \mathtt{x}'_j\rangle$ from Proposition 9 and we take s_j as t_j. If $\mathtt{x}'_j = \mathtt{a}$, then there exists s_j such that $s_j \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}\, t_j$ and $s_j \xrightarrow[\Delta_0]{*} \langle q'_j, \mathtt{x}'_j\rangle$ from the induction hypothesis. Here, we have no j that satisfies both $j \notin \mu(f)$ and $\mathtt{x}'_j = \mathtt{a}$, since $j \in \mu(f)$ coincides with $\mathtt{x}'_j = \mathtt{a}$ from the construction of Δ_0. Therefore, we have $g(s_1, \ldots, s_m) \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}\, g(t_1, \ldots, t_m)$ and $g(s_1, \ldots, s_m) \xrightarrow[\Delta_0]{*} g(\langle q'_1, \mathtt{x}'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m\rangle) \xrightarrow[\Delta_0]{} \langle q, \mathtt{a}\rangle$. Therefore the claim holds by letting $s = g(s_1, \ldots, s_m)$.

   (b) Consider the case where the last transition rule applied at α is in $\Delta_k \backslash \Delta_{k-1}$ for $k > 0$. We show the diagram of this proof at Figure 3.1. Assume that α is represented as $t = g(t_1, \ldots, t_m) \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m\rangle) \xrightarrow[\Delta_k \backslash \Delta_{k-1}]{*} \langle q, \mathtt{x}\rangle$. Let $\alpha_j : t_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j\rangle$. Since the rule $g(\langle q'_1, \mathtt{x}'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m\rangle) \xrightarrow[\Delta_0]{} \langle q, \mathtt{a}\rangle$ is produced by the first inference rule at Step 2, there exist $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$, $f(\langle q_1, \mathtt{x}_1\rangle, \ldots, \langle q_n, \mathtt{x}_n\rangle) \to \langle q, \mathtt{a}\rangle \in \Delta_{k-1}$, and $\theta : X \to Q_*$ such that

   - $l_i\theta \xrightarrow[\Delta_{k-1}]{*} \langle q_i, \mathtt{x}_i\rangle$,
   - $\mathcal{L}(\Delta_0, x\theta) \neq \emptyset$ for each erasing variable x.

   Then, each $\langle q'_j, \mathtt{x}'_j\rangle$ is given as follows:

   - $q'_j = \begin{cases} q''_j & \cdots \text{ if } r_j \in X \\ q^{r_j} & \cdots \text{ if } r_j \notin X \end{cases}$
   - $\mathtt{x}'_j = \begin{cases} \mathtt{x}''_j & \cdots \text{ if } j \notin \mu(g) \wedge r_j \in X \\ \mathtt{i} & \cdots \text{ if } j \notin \mu(g) \wedge r_j \notin X \\ \mathtt{a} & \cdots \text{ if } j \in \mu(g) \end{cases}$

   where $r_j\theta = \langle q''_j, x''_j\rangle$ for $r_j \in X$.

The following (i) – (iv) show that for each j, there exists s_j such that $s_j \xhookrightarrow[R]{*}{}^{\mu} t_j$, $s_j \xrightarrow[\Delta_*]{*} r_j\theta$ and $\alpha'_j : s_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j\rangle$ with $\alpha_j \sqsupseteq \alpha'_j$.

i. For j such that $r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a})$, we have $t_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j\rangle = \langle q''_j, \mathtt{x}''_j\rangle = r_j\theta$. We take t_j as s_j ant hence $\alpha_j \sqsupseteq \alpha'_j : s_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j\rangle$.

ii. For j such that $r_j \in X \wedge j \in \mu(g) \wedge \mathtt{x}''_j = \mathtt{i}$, there exists s_j such that $s_j \xrightarrow[\Delta_0]{*} \langle q'_j, \mathtt{x}'_j\rangle = \langle q''_j, \mathtt{a}\rangle$ and $s_j \xhookrightarrow[R]{*}{}^{\mu} t_j$ from the induction hypothesis. Then we have $\alpha'_j : s_j \xrightarrow[\Delta_0]{*} \langle q''_j, \mathtt{i}\rangle = r_j\theta$ from Proposition 9 and $\alpha_j \sqsupseteq \alpha'_j$.

iii. For j such that $r_j \notin X \wedge j \notin \mu(g)$, we have $q'_j = q^{r_j}$ and $\mathtt{x}'_j = \mathtt{i}$. Since $t_j \xrightarrow[\Delta_0]{*} \langle q^{r_j}, \mathtt{i}\rangle$ by Proposition 9, we have $t_j = r_j$ from Proposition 7 and 8. Therefore $t_j = r_j\theta$ follows from right-shallowness of R. We take s_j as t_j. ant hence $\alpha_j \sqsupseteq \alpha'_j : s_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j\rangle$.

iv. For j such that $r_j \notin X \wedge j \in \mu(g)$, there exists s_j such that $s_j \xhookrightarrow[R]{*}{}^{\mu} t_j$ and $s_j \xrightarrow[\Delta_0]{*} \langle q'_j, \mathtt{x}'_j\rangle = \langle q^{r_j}, \mathtt{a}\rangle$ from the induction hypothesis. Then, we have $\alpha'_j : s_j \xrightarrow[\Delta_0]{*} \langle q^{r_j}, \mathtt{i}\rangle$ by Proposition 10, and hence we have $s_j = r_j$ from Proposition 7 and 8. Therefore $s_j = r_j\theta$ follows from right-shallowness of R and $\alpha_j \sqsupseteq \alpha'_j$.

Thus we have $g(s_1, \ldots, s_m) \xhookrightarrow[R]{*}{}^{\mu} g(t_1, \ldots, t_m)$, $g(s_1, \ldots, s_m) \xrightarrow[\Delta_*]{*} g(r_1, \ldots, r_m)\theta$, and $\alpha' : g(s_1, \ldots, s_m) \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m\rangle) \xrightarrow[\Delta_k \setminus \Delta_{k-1}]{} \langle q, \mathtt{a}\rangle$ where $\alpha \sqsupseteq \alpha'$ since $\alpha_j \sqsupseteq \alpha'_j$ for all $q \leq j \leq m$.
We define a substitution $\sigma : \mathrm{Var}(f(l_1, \ldots, l_n)) \to \mathcal{T}(F)$ as follows:

$$x\sigma = \begin{cases} s_j \cdots & \text{if there exists } j \text{ such that } r_j = x \\ s' \cdots & \text{otherwise, choose an arbitrary } s' \text{ such that } s' \xrightarrow[\Delta_{k-1}]{*} x\theta \end{cases}$$

where σ is well-defined from the right-linearity of rewrite rules. Here, we have $g(s_1, \ldots, s_m) = g(r_1, \ldots, r_m)\sigma$ and we can construct $\beta : f(l_1, \ldots, l_n)\sigma \xrightarrow[\Delta_*]{*} f(l_1, \ldots, l_n)\theta \xrightarrow[\Delta_{k-1}]{*} f(\langle q_1, \mathtt{x}_1\rangle, \ldots, \langle q_n, \mathtt{x}_n\rangle) \xrightarrow[\Delta_{k-1}]{} \langle q, \mathtt{a}\rangle$.
Hence we have $f(l_1, \ldots, l_n)\sigma \xhookrightarrow[R]{}{}^{\mu} g(s_1, \ldots, s_m)$. Here $\alpha \sqsupseteq \alpha' \sqsupset \beta$ follows from left-linearity of R. Thus, there exists s such that $s \xhookrightarrow[R]{*}{}^{\mu} f(l_1, \ldots, l_n)\sigma \xhookrightarrow[R]{}{}^{\mu}_{\mathrm{in}} g(s_1, \ldots, s_m) \xhookrightarrow[R]{*}{}^{\mu} g(t_1, \ldots, t_m) = t$ and $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{a}\rangle$ by the induction hypothesis.

$$
\begin{tikzcd}
g(t_1,\ldots,t_m) \arrow[r, "*", "\Delta_*"'] & g(\langle q_1',\mathtt{x}_1'\rangle,\ldots,\langle q_m',\mathtt{x}_m'\rangle) \arrow[r, "\Delta_k\setminus\Delta_{k-1}"'] & \langle q,\mathtt{a},u'\rangle \\
g(s_1,\ldots,s_m) = g(r_1,\ldots,r_m)\sigma \arrow[u, hook, "\mu\,*", "R"'] \arrow[ur, "*", "\Delta_*"'] \arrow[r, phantom, "\mathrm{I.H.}"] & & \\
f(l_1,\ldots,l_n)\sigma \arrow[u, hook, "\mu", "R"'] \arrow[r, "*", "\Delta_*"'] & f(\langle q_1,\mathtt{x}_1\rangle,\ldots,\langle q_n,\mathtt{x}_n\rangle) \arrow[uur, "*", "\Delta_{k-1}"'] & \\
s \arrow[u, hook, "\mu\,*", "R"'] \arrow[uuurr, "*", "\Delta_0"'] \arrow[r, phantom, "\mathrm{I.H.}"] & &
\end{tikzcd}
$$

Figure 3.2: The diagram of proof of lemma 16

2. In the case where the transition rule applied last in α is (in the form of) $\langle q',\mathtt{x}\rangle \to \langle q,\mathtt{x}\rangle \in \Delta_k \setminus \Delta_{k-1}$, the lemma can be shown similarly to the previous case. $\square$

Lemma 17 *If R be a linear and right-shallow TRS and μ be a replacement map, then $\mathcal{L}(\mathcal{A}_*) \subseteq \hookrightarrow_R^{\mu}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let $t \xrightarrow[\Delta_*]{*} \langle q,\mathtt{x}\rangle \in Q_*^f$ then we have $s \overset{*}{\underset{R}{\hookleftarrow}}{}^{\mu}\ t$ and $s \xrightarrow[\Delta_0]{*} \langle q,\mathtt{x}\rangle \in Q_*^f$ by Lemma 16. Since $s \xrightarrow[\Delta_0]{*} \langle q,\mathtt{i}\rangle$ by Proposition 10, we have $s \xrightarrow[\Delta]{*} q \in Q^f$ by Proposition 7. $\square$

The following theorem is proved by lemma 15, lemma 17 and theorem 18.

Theorem 18 *Context-sensitive reductions for linear and right-shallow TRSs effectively preserves regularity. Thus reachability is decidable for linear and right-shallow TRSs with respect to the context-sensitive reduction.*

3.2 On the Context-Sensitive Innermost Reduction

In this section, we show that reachability is decidable for linear and right-shallow TRSs with respect to the context-sensitive innermost reduction. At first, we show the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ that output a TA recognizing the set of reachable terms.

The main idea employed in the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is that the states in the output automata has three components. The first component originates in the input automata and remembers a reachable set. The second component is `a` or `i` and is used similarly to the case of context-sensitive. The third component remembers whether the terms accepted by the state are normal forms or not. This component is necessary because every proper subterm of the innermost redex must be a normal form or at an unrewritable position. We use states of deterministic complete reduced TA recognizing the set of normal forms as the third component. We show the construction of the TA in the first subsection.

However, $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ only works correctly if the set of transition rules of an input automaton Δ fulfill the restriction that $s \to q \in \Delta$ and $t \to q \in \Delta$ implies $s = t$. Therefore the set of terms recognized by an input automaton must be finite and hence we cannot prove effective preservation of regularity for the context-sensitive innermost reduction of linear and right-shallow TRSs by introducing $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$. This restriction is not a problem to prove decidability of reachability since we can solve any reachability problem for the context-sensitive innermost reduction of linear and right-shallow TRSs. The method for solve the reachability problem for terms s, t and a TRS R and a replacement map μ by using $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is that we construct a TA $\mathcal{A}$ recognizing s, construct a TA $\mathcal{A}_*$ recognizing $\underset{R}{\hookrightarrow}{}^{\mu}_{\mathrm{in}}[\mathcal{L}(\mathcal{A})]$, and check the membership of t for $\mathcal{A}_*$.

Besides, we introduce the other algorithm $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ to show effective preservation of regularity. Effective preservation of regularity is more powerful property than decidability of reachability and is useful for program verification. Therefore, we also prove this property. The algorithm $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ is obtained by removing the restriction of input automata and adding a preprocess on the input into $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$. The preprocess enriches states by augmenting information whether the term is a normal form or not again.

3.2.1 Tree Automata Accepting Normal Forms

In this section, we give how to construct a deterministic, complete, and reduced TA $\mathcal{A}_{\mathrm{NF}}$ that recognizes the set of normal forms over F for the context-sensitive reduction of left-linear TRS R and a replacement map μ. This algorithm is similar to ones for the ordinary reduction [5].

The steps of the algorithm to construct TA $\mathcal{A}_{\mathrm{NF}}$ is as follows:

1. Construct the TA $\mathcal{A}_l$ that recognizes the set of terms having a redex $l\sigma$ at μ-replacing position for each $l \to r \in R$.

2. Construct the union of all $\mathcal{A}_l$'s and take the complementation.

3. Convert the TA constructed in the step (2) into a deterministic, complete, and reduced TA. We output the TA as $\mathcal{A}_{\mathrm{NF}}$.

The step (2) and (3) are obviously possible from Theorem 2. Now we show the detail of the step (1). We use $s^{\perp}$ to denote the term obtained from a term s by replacing every variable with $\perp$.

Each component of $\mathcal{A}_l$ is as follows:

- $Q_l = \{u^{\circ}\} \cup \{u_{t^{\perp}} \mid t \trianglelefteq l\}$

- $Q_l^f = \{u^{\circ}\}$

- Δ_l consists of following transition rules:

  (i) $f(u_{\perp}, \ldots, u_{\perp}) \to u_{\perp}$ for each $f \in F$,

  (ii) $f(u_{t_1^{\perp}}, \ldots, u_{t_n^{\perp}}) \to u_{f(t_1,\ldots,t_n)^{\perp}}$ for each $f \in F$ and state $u_{f(t_1,\ldots,t_n)^{\perp}}$,

  (iii) $u_{l^{\perp}} \to u^{\circ}$,

  (iv) $f(u_1, \ldots, u_n) \to u^{\circ}$ for each $f \in F$ if exactly one u_j such that $j \in \mu(f)$ is u° and the other u_i's are $u_{\perp}$.

  Note that transition rules in (i), (ii), (iii) are the same in the case of the ordinary reduction.

The state $u_\perp$ accepts arbitrary terms by the rules of (i), the states of the form $u_{t^\perp}$ accepts instances of t by the rules of (ii), and the state u° accepts terms having a instance of l at a rewritable position by the rules of (iii) and (iv). Thus, $\mathcal{A}_l$ accepts terms having a instance of l at a rewritable position. The following lemmas imply that $\mathcal{A}_l$ constructed correctly.

Lemma 19 *If l is linear, then $\mathcal{L}(\mathcal{A}_l, u_{l^\perp})$ is equal to the set of all ground instances of l, (that is, $\mathcal{L}(\mathcal{A}_l, u_{l^\perp}) = \{l\sigma \in \mathcal{T}(F)\}$.)*

Proof:

($\supseteq$) By induction on height $|t|$ of t, we show the claim that $t\sigma \xrightarrow[\Delta_l]{*} u_{t^\perp}$ for every substitution σ and subterm t of l such that $t\sigma$ is ground.

1. In the case where t is a variable, we have $t^\perp = \perp$. Since a transition rule in (i) guarantees that $s \xrightarrow[\Delta_l]{*} u_\perp$ for any ground term s, the claim follows.

2. Otherwise, t is represented by $f(t_1, \ldots, t_n)$ for $n \geq 0$. Since $t_i\sigma \xrightarrow[\Delta_l]{*} u_{t_i^\perp}$ by the induction hypothesis, we have $t\sigma = f(t_1, \ldots, t_n)\sigma \xrightarrow[\Delta_l]{*} f(u_{t_1^\perp}, \ldots, u_{t_n^\perp}) \xrightarrow[\Delta_l]{} u_{t^\perp}$ from rules in (ii).

($\subseteq$) We show that for subterm t of l, if $\alpha : \ s \xrightarrow[\Delta_l]{*} u_{t^\perp}$ then there exists a substitution σ such that $s = t\sigma$ by induction on length $|\alpha|$. If t is a variable x, we can take σ so that $x\sigma = s$. Otherwise, since the transition rule used at the last step of α is in (ii), α is represented by $s = f(s_1, \ldots, s_n) \xrightarrow[\Delta_l]{*} f(u_{t_1^\perp}, \ldots, u_{t_n^\perp}) \xrightarrow[\Delta_l]{} u_{t^\perp}$. From $s_i \xrightarrow[\Delta_l]{*} u_{t_i^\perp}$, there exists σ_i such that $s_i = t_i\sigma_i$ by the induction hypothesis. Since l is linear, there exists a σ such that $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma_1, \ldots, t_n\sigma_n)$ and hence we have $s = f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)\sigma = t\sigma$. $\square$

Lemma 20 *If l is linear then $\mathcal{L}(\mathcal{A}_l) = \{t[s]_p \mid t \in \mathcal{T}(F),\ s$ is a ground instance of l, $p \in Pos^\mu(t)\}$.*

Proof:

($\supseteq$) From Lemma 19 and the rules in (iii), we have $t[s]_p \xrightarrow[\Delta_l]{*} t[u_{l^\perp}]_p \xrightarrow[\Delta_l]{} t[u^\circ]_p$. Since $p \in \mathrm{Pos}^\mu(t)$, we have $t[u^\circ]_p \xrightarrow[\Delta_l]{*} u^\circ$ from the definition of Pos^μ and the rules in (i) and (iv).

($\subseteq$) Let $t \xrightarrow[\Delta_l]{*} u^\circ$, then we have $t \xrightarrow[\Delta_l]{*} t[u_{l^\perp}]_p \xrightarrow[\Delta_l]{} t[u^\circ]_p \xrightarrow[\Delta_l]{*} u^\circ$ for some $p \in \mathrm{Pos}^\mu(t)$ from the rules in (iii) and (iv). Hence we have $t = t[s]_p$ for some ground instance s of l from Lemma 19. $\square$

As shown by the Lemma 20, the TA $\mathcal{A}_l$ recognizes the set of terms having a redex $l\sigma$ at a μ-replacing position. Now we obtain the following lemma.

Lemma 21 *For a left-linear TRS R and a replacement map μ, we can construct a deterministic, complete, and reduced TA $\mathcal{A}_{\mathrm{NF}}$ that recognizes NF^μ_R.*

Proof: In the step (1) and (2) of the algorithm constructing $\mathcal{A}_{\mathrm{NF}}$, we obtain a TA $\mathcal{A}'$ that recognizes the complementation of the following set:

$$\bigcup_{l \to r \in R} \mathcal{L}(\mathcal{A}_l)$$

From Lemma 20 and (2) of Theorem 2, $\mathcal{L}(\mathcal{A}')$ is the set of terms having a redex of the context-sensitive reduction of R and μ. The TA $\mathcal{A}_{\mathrm{NF}}$ obtained in the step (3) of the algorithm is a deterministic, complete, and reduced TA that recognizes NF^μ_R from (1) of Theorem 2. $\square$

We show an example of $\mathcal{A}_{\mathrm{NF}}$.

Example 22 Consider the following TRS R and the replacement map μ:

$$R = \left\{ \begin{array}{rclrclrcl} a & \to & b, & c & \to & e, & f(g(x)) & \to & x, \\ g(b) & \to & f(e), & g(c) & \to & e, & & & \end{array} \right\},$$
$$\mu(f) = \{1\}, \mu(g) = \emptyset.$$

Final states and transition rules of the deterministic, complete, and TA $\mathcal{A}_{\mathrm{NF}}$

that is minimized are as follows:

$$Q = \{u_\perp, u_c, u_g, \overline{u_b}, \overline{u_g}, \overline{u_\perp}\},$$
$$Q^f_{\mathrm{NF}} = \{\overline{u_b}, \overline{u_g}, \overline{u_\perp}\},$$
$$\Delta_{\mathrm{NF}} = \left\{ \begin{array}{rclrclrclrcl} a & \to & u_\perp, & b & \to & \overline{u_b}, & c & \to & u_c, & e & \to & \overline{u_\perp}, \\ g(u_\perp) & \to & \overline{u_g}, & g(\overline{u_\perp}) & \to & \overline{u_g}, & g(\overline{u_b}) & \to & u_g, & g(u_c) & \to & u_g, \\ g(u_g) & \to & \overline{u_g}, & h(\overline{u_g}) & \to & \overline{u_g}, & f(u_\perp) & \to & u_\perp, & f(\overline{u_\perp}) & \to & \overline{u_\perp}, \\ f(\overline{u_b}) & \to & \overline{u_\perp}, & f(u_c) & \to & u_\perp, & f(u_g) & \to & u_\perp, & f(\overline{u_g}) & \to & u_\perp \end{array} \right\}$$

□

For the constructed TA $\mathcal{A}_{\mathrm{NF}}$, following Proposition 23 holds from Lemma 21. Proposition 23 is used in the next subsection.

Proposition 23 *If $f(u_1, \ldots, u_n) \to u \in \Delta_{\mathrm{NF}}$ and $u \in Q^f_{\mathrm{NF}}$, then $i \in \mu(f)$ implies $u_i \in Q^f_{\mathrm{NF}}$.*

Proof: Let $f(u_1, \ldots, u_n) \to u \in \Delta_{\mathrm{NF}}$, $u \in Q^f_{\mathrm{NF}}$, and assume $u_i \notin Q^f_{\mathrm{NF}}$ for some $i \in \mu(f)$. Since $\mathcal{A}_{\mathrm{NF}}$ is reduced by Lemma 21, there exists a terms $t_1, \ldots, t_n$ such that $t_j \xrightarrow[\Delta_{\mathrm{NF}}]{*} u_j$ for each $1 \leq j \leq n$. Hence we have $f(t_1, \ldots, t_n) \xrightarrow[\Delta_{\mathrm{NF}}]{*} f(u_1, \ldots, u_n) \xrightarrow[\Delta_{\mathrm{NF}}]{} u$. Here we have $f(t_1, \ldots, t_n) \in \mathrm{NF}^\mu_R$ and $t_i \notin \mathrm{NF}^\mu_R$ from Lemma 21. Since t_i is not a normal form and $i \in \mu(f)$, the term $f(t_1, \ldots, t_n)$ is not a normal form, which contradicts $f(t_1, \ldots, t_n) \in \mathrm{NF}^\mu_R$. □

3.2.2 Tree Automata Accepting Reachable Terms

In this subsection, we give an algorithm to construct a tree automaton that recognizes the set of reachable terms by the context-sensitive innermost reduction of a linear and right-shallow TRS. An algorithm to construct a TA recognizing the set of reachable terms from a given set of terms by the context-sensitive reduction of a linear and right-shallow TRSs have already shown in the previous section. One may think that we can construct a tree automaton by adding a state of $\mathcal{A}_{\mathrm{NF}}$ in the previous subsection to each state of output TA of the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$. However, this is not true and we need some more modification.

In the following, we give the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ that constructs a TA recognizing the set of reachable terms by the context-sensitive innermost reduction. The algorithm design is based on modifying the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$. More precisely, each state of output TA consists of three components; the first and second component remember the same information as states of output TA of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$. The third one is a state of $\mathcal{A}_{\mathrm{NF}}$ and hence remembers whether the terms accepted by the states are normal forms or not.

Differently from $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$, input automata of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ has the restriction such that $w_1 \to q \in \Delta$ and $w_2 \to q \in \Delta$ imply $w_1 = w_2$. If $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ does not have the restriction, the output automata does not always work correctly. Due to the restriction, we cannot show effective preservation of regularity for the context-sensitive innermost reduction of the linear and right-shallow TRSs by $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$. However, this restriction is not a problem to prove decidability of reachability since an input of reachability problem is a term and we can construct a tree automaton that accepts only the input term under the restriction.

In the next section, we show effective preservation of regularity for the context-sensitive innermost reduction of the linear and right-shallow TRSs by introducing another algorithm.

Here we show the concrete description of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$.

Algorithm $\mathbf{P}^{\mathrm{lrs}}_{\mathrm{csin}}$:

Input A TA $\mathcal{A} = \langle Q, Q^f, \Delta \rangle$ such that $w_1 \to q \in \Delta$ and $w_2 \to q \in \Delta$ imply $w_1 = w_2$, a linear and right-shallow TRS R, and a replacement map μ.

Output A TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_* \rangle$ such that $\mathcal{L}(\mathcal{A}_*) = \overset{\mu}{\underset{R}{\hookrightarrow}}{}_{\mathrm{in}}[\mathcal{L}(\mathcal{A})]$.

Step 1 (initialize) 1. Prepare $\mathcal{A}_{\mathrm{RS}}$ and $\mathcal{A}_{\mathrm{NF}}$. $\mathcal{A}_{\mathrm{RS}}$ is constructed similarly in $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$.

2. Let
   - $k := 0$,
   - $Q_* = (Q \uplus Q_{\mathrm{RS}}) \times \{\mathtt{a}, \mathtt{i}\} \times Q_{\mathrm{NF}}$,
   - $Q^f_* = Q^f \times \{\mathtt{a}\} \times Q_{\mathrm{NF}}$, and
   - Δ_0 as follows:
     (a) $\langle q', \mathtt{x}, u \rangle \to \langle q, \mathtt{x}, u \rangle \in \Delta_0$ for $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$ where $q' \to q \in \Delta$ and $u \in Q_{\mathrm{NF}}$,

(b) $f(\langle q_1, \mathtt{i}, u_1\rangle, \ldots, \langle q_n, \mathtt{i}, u_n\rangle) \rightarrow \langle q, \mathtt{i}, u\rangle \in \Delta_0$ where $f(q_1, \ldots, q_n) \rightarrow q \in \Delta \cup \Delta_{\mathrm{RS}}$ and $f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{\mathrm{NF}}$, and

(c) $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \rightarrow \langle q, \mathtt{a}, u\rangle \in \Delta_0$ where $f(q_1, \ldots, q_n) \rightarrow q \in \Delta \cup \Delta_{\mathrm{RS}}$, $f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{\mathrm{NF}}$, and if $i \in \mu(f)$ then $\mathtt{x}_i = \mathtt{a}$, otherwise $\mathtt{x}_i = \mathtt{i}$.

Step 2 Let Δ_{k+1} be the set of transition rules produced by augmenting transition rules of Δ_k by the following inference rules:

$$\frac{\begin{array}{c}f(l_1, ..., l_n) \rightarrow g(r_1, ..., r_m) \in R,\\ f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \rightarrow \langle q, \mathtt{a}, u\rangle \in \Delta_k\end{array}}{g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \rightarrow \langle q, \mathtt{a}, u'\rangle \in \Delta_{k+1}}$$

if $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, and there exists $\theta : X \rightarrow (Q \uplus Q_{\mathrm{RS}}) \times \{\mathtt{a}, \mathtt{i}\} \times Q^f_{\mathrm{NF}}$ such that $\mathcal{L}(\Delta_k, x\theta) \neq \emptyset$ for all $x \in \mathrm{Var}(f(l_1, \ldots, l_n))$ and $l_i\theta \xrightarrow[\Delta_k]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$. Let $r_j\theta = \langle q''_j, \mathtt{x}''_j, u''_j\rangle$ for each $r_j \in X$. Each $\langle q'_j, \mathtt{x}'_j, u'_j\rangle$ is determined as follows

- $q'_j = \begin{cases} q''_j & \cdots \text{ if } r_j \in X \\ q^{r_j} & \cdots \text{ if } r_j \notin X \end{cases}$
- $\mathtt{x}'_j = \begin{cases} \mathtt{x}''_j & \cdots \text{ if } j \notin \mu(g) \wedge r_j \in X \\ \mathtt{i} & \cdots \text{ if } j \notin \mu(g) \wedge r_j \notin X \\ \mathtt{a} & \cdots \text{ if } j \in \mu(g) \end{cases}$
- $u'_j = \begin{cases} u''_j & \cdots \text{ if } r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a}) \\ v \in Q_{\mathrm{NF}} & \cdots \text{ if } r_j \in X \Rightarrow (j \in \mu(g) \wedge \mathtt{x}''_j = \mathtt{i}) \end{cases}$

where $g(u'_1, \ldots, u'_m) \xrightarrow[\Delta_{\mathrm{NF}}]{} u'$ for all $1 \leq j \leq m$, and

$$\frac{f(l_1, ..., l_n) \rightarrow x \in R,\ f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \rightarrow \langle q, \mathtt{a}, u\rangle \in \Delta_k}{\langle q', \mathtt{a}, u'\rangle \rightarrow \langle q, \mathtt{a}, u'\rangle \in \Delta_{k+1}}$$

if $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, and there exists $\theta : X \rightarrow (Q \uplus Q_{\mathrm{RS}}) \times \{\mathtt{a}, \mathtt{i}\} \times Q^f_{\mathrm{NF}}$ such that $\mathcal{L}(\Delta_k, x\theta) \neq \emptyset$ for all $x \in \mathrm{Var}(f(l_1, \ldots, l_n))$ and $l_i\theta \xrightarrow[\Delta_k]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$. Let $\langle q'', \mathtt{x}'', u''\rangle = x\theta$ then

- $q' = q''$ and
- $u' = \begin{cases} u'' & \cdots \text{ if } \mathtt{x}'' = \mathtt{a} \\ v \in Q_{\mathrm{NF}} & \cdots \text{ if } \mathtt{x}'' = \mathtt{i} \end{cases}$

Step 3 If $\Delta_{k+1} = \Delta_k$ then stop and set $\Delta_* = \Delta_k$. Otherwise, $k := k+1$, and go to step 2.

Example 24 Let us follow how the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ works. Consider R and μ in Example 22, and let $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ be the TA where

$$\begin{array}{l}
Q = \{q^a, q^c, q^{g(a)}, q^{g(c)}, q^{f(g(a))}, q^{f(g(c))}\},\\
Q^f = \{q^{f(g(a))}, q^{f(g(c))}\},\\
\Delta = \left\{ \begin{array}{rclrclrcl} a & \to & q^a, & c & \to & q^c, & g(q^a) & \to & q^{g(a)},\\ g(q^c) & \to & q^{g(c)}, & f(q^{g(a)}) & \to & q^{f(g(a))}, & f(q^{g(c)}) & \to & q^{f(g(c))} \end{array} \right\},
\end{array}$$

and hence $\mathcal{L}(\mathcal{A}) = \{f(g(a)), f(g(c))\}$.

At 1 of Step 1, we have $\mathcal{A}_{\mathrm{RS}}$ and $\mathcal{A}_{\mathrm{NF}}$. $\mathcal{A}_{\mathrm{RS}} = \langle Q_{\mathrm{RS}}, Q^f_{\mathrm{RS}}, \Delta_{\mathrm{RS}}\rangle$ is defined as:

$$\begin{array}{l} Q_{\mathrm{RS}} = Q^f_{\mathrm{RS}} = \{q^e\},\\ \Delta_{\mathrm{RS}} = \{e \to q^e\}, \end{array}$$

and $\mathcal{A}_{\mathrm{NF}}$ is defined as Example 22.

At 2 of Step 1, we have

$$\begin{array}{l}
Q_* = \left\{ \begin{array}{l} \langle q^a, \mathtt{x}, u\rangle, \langle q^c, \mathtt{x}, u\rangle, \langle q^{g(a)}, \mathtt{x}, u\rangle, \langle q^{g(c)}, \mathtt{x}, u\rangle,\\ \langle q^{f(g(a))}, \mathtt{x}, u\rangle, \langle q^{f(g(c))}, \mathtt{x}, u\rangle, \langle q^e, \mathtt{x}, u\rangle \end{array} \right\}\\
Q^f_* = \{\langle q^{f(g(a))}, \mathtt{a}, u\rangle, \langle q^{f(g(c))}, \mathtt{a}, u\rangle\}\\
\Delta_0 = \left\{ \begin{array}{rclrcl}
a & \to & \langle q_a, \mathtt{x}, u_\perp\rangle, & c & \to & \langle q_c, \mathtt{x}, u_c\rangle,\\
e & \to & \langle q^e, \mathtt{x}, \overline{u_\perp}\rangle, & g(\langle q_a, \mathtt{i}, u_1\rangle) & \to & \langle q^{g(a)}, \mathtt{x}, u_g\rangle,\\
g(\langle q_a, \mathtt{i}, u_2\rangle) & \to & \langle q^{g(a)}, \mathtt{x}, \overline{u_g}\rangle, & g(\langle q_c, \mathtt{i}, u_1\rangle) & \to & \langle q^{g(c)}, \mathtt{x}, u_g\rangle,\\
g(\langle q_c, \mathtt{i}, u_2\rangle) & \to & \langle q^{g(c)}, \mathtt{x}, \overline{u_g}\rangle, & f(\langle q^{g(a)}, \mathtt{x}, u_3\rangle) & \to & \langle q^{f(g(a))}, \mathtt{x}, u_\perp\rangle,\\
f(\langle q^{g(a)}, \mathtt{x}, u_4\rangle) & \to & \langle q^{f(g(a))}, \mathtt{x}, \overline{u_\perp}\rangle, & f(\langle q^{g(c)}, \mathtt{x}, u_3\rangle) & \to & \langle q^{f(g(c))}, \mathtt{x}, u_\perp\rangle,\\
f(\langle q^{g(c)}, \mathtt{x}, u_4\rangle) & \to & \langle q^{f(g(c))}, \mathtt{x}, \overline{u_\perp}\rangle & & &
\end{array} \right\}
\end{array}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$, $u_1 \in \{\overline{u_b}, u_c\}$, $u \in Q_{\mathrm{NF}}$, $u_2 \in \{u_\perp, \overline{u_\perp}, u_g, \overline{u_g}\}$, $u_3 \in \{u_\perp, u_c, u_g, \overline{u_g}\}$, and $u_4 \in \{\overline{u_b}, \overline{u_\perp}\}$.

The saturation step stops at $k = 1$, and we have

$$\Delta_1 = \Delta_0 \cup \left\{ \begin{array}{rclrcl} b & \to & \langle q_a, \mathtt{a}, \overline{u_b}\rangle, & e & \to & \langle q_c, \mathtt{a}, \overline{u_\perp}\rangle,\\ e & \to & \langle q^{g(c)}, \mathtt{a}, \overline{u_\perp}\rangle, & \langle q_a, \mathtt{a}, u\rangle & \to & \langle q^{f(g(a))}, \mathtt{a}, u\rangle \end{array} \right\}$$

where $u \in Q_{\mathrm{NF}}$, Finally we have $\Delta_2 = \Delta_1$ and hence $\Delta_* = \Delta_1$.

The output TA $\mathcal{A}_*$ recognizes the set $\{f(g(a)), f(g(c)), a, b, f(e)\}$ that is equal to $\hookrightarrow^{\mu}_{R\,\mathrm{in}} [f(g(a)), f(g(c))]$. □

Similarly to $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$, $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ eventually terminates at some k and $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$.

To show the correctness of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$, we show several technical lemmas. We abbreviate the proofs for Proposition 25–29 and Lemma 30 and 31 since the third parameter of each state of automata does not affect to proofs and hence the proofs are almost same in the case of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$.

Proposition 25 *Let $s \in \mathcal{T}(F)$, then $s \xrightarrow[\Delta]{*} q$ for $q \in Q$ or $s \xrightarrow[\Delta_{\mathrm{RS}}]{*} q \in Q_{\mathrm{RS}}$ if and only if $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u\rangle$ for some $u \in Q_{NF}$.*

Proof: This proposition is proved similarly to the proof of Proposition 7. □

Proposition 26 *Let $t \in \mathcal{T}(F)$. $t \xrightarrow[\Delta_0]{*} \langle q^{t'}, \mathtt{i}, u\rangle \in Q_{\mathrm{RS}} \times \{\mathtt{i}\} \times Q_{\mathrm{NF}}$ iff $t = t'$.*

Proof: This proposition is proved similarly to the proof of Proposition 8. □

Proposition 27 *Let $t \in \mathcal{T}(F)$. $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{i}, u\rangle \in Q_*$ implies $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u\rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 9. □

Proposition 28 *Let $t \in \mathcal{T}(F)$. $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{a}, u\rangle \in Q_*$ if and only if $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u\rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 10. □

Proposition 29 *Let $t \in \mathcal{T}(F)$. $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{i}, u\rangle \in Q_*$ implies $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{a}, u\rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 11. □

Lemma 30 *If $\alpha :\ t[t']_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u\rangle$ and $p \in \mathrm{Pos}^{\mu}(t)$, then there exists $\langle q', \mathtt{a}, u'\rangle$ such that $t' \xrightarrow[\Delta_*]{*} \langle q', \mathtt{a}, u'\rangle$ and $t[\langle q', \mathtt{a}, u'\rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u\rangle$.*

Proof: This lemma is proved similarly to the proof of Lemma 12. □

Lemma 31 *Let R be a linear and right-shallow TRS and $l \to g(r_1, \ldots, r_m) \in R$. For $r_j \notin X$ and any substitution σ, $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q^{r_j}, \mathtt{i}, u\rangle$ and $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q^{r_j}, \mathtt{a}, u\rangle$ for some u.*

Proof: By Proposition 26 and 29, this lemma is proved similarly to the proof of Lemma 13. □

Proposition 32 *$f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle q, \mathtt{x}, u\rangle \in \Delta_*$ implies $f(u_1, \ldots, u_n) \to u \in \Delta_{\mathrm{NF}}$.*

Proof: Direct consequence of the construction of Δ_0 and Δ_*. □

Proposition 33 *If a rule $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle q, \mathtt{i}, u\rangle$ is in Δ_*, then it is also in Δ_0. Moreover, $\mathtt{x}_i = \mathtt{i}$ for all $1 \le i \le n$.*

Proof: Such rules are introduced at Step 1 and hence the claim follows from the construction of Δ_0. □

Lemma 34 *For any k, if $\alpha : t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{x}, u\rangle$, then $t \xrightarrow[\Delta_{\mathrm{NF}}]{*} u$.*

Proof: We show the proof by induction on $|\alpha|$. If the last transition rule applied in α is of the form $\langle q', \mathtt{x}, u'\rangle \to \langle q, \mathtt{x}, u\rangle$, then we have $u' = u$ from the construction of Δ_k and hence $t \xrightarrow[\Delta_{\mathrm{NF}}]{*} u$ from the induction hypothesis. Otherwise, assume that t is represented as $f(t_1, \ldots, t_n)$ and the last transition rule applied in α is $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle P, \mathtt{x}, u\rangle$.

If $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle q, \mathtt{x}, u\rangle \in \Delta_k$, then we have $f(u_1, \ldots, u_n) \to u \in \Delta_{\mathrm{NF}}$ from Proposition 32. Since we have $t|_i \xrightarrow[\Delta_{\mathrm{NF}}]{*} u_i$ from the induction hypothesis, we have $t \xrightarrow[\Delta_{\mathrm{NF}}]{*} f(u_1, \ldots, u_n) \xrightarrow[\Delta_{\mathrm{NF}}]{} u$. □

Lemma 35 *Let $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{x}, u\rangle$, and $t \xrightarrow[\Delta_0]{*} \langle q', \mathtt{x}', u'\rangle$. Then, $q = q'$ implies $s = t$*

Proof: At first, we have s $\xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u\rangle$ and $t \xrightarrow[\Delta_0]{*} \langle q', \mathtt{i}, u'\rangle$ from Proposition 28. If $q = q'$, then we have either $s \xrightarrow[\Delta]{*} q \wedge t \xrightarrow[\Delta]{*} q$ or $s \xrightarrow[\Delta_{\mathrm{RS}}]{*} q^s(= q) \wedge t \xrightarrow[\Delta_{\mathrm{RS}}]{*} q^t(= q)$ from Proposition 25 and the construction of $\mathcal{A}_{\mathrm{RS}}$. In the case of $s \xrightarrow[\Delta_{\mathrm{RS}}]{*} q^s$ and $t \xrightarrow[\Delta_{\mathrm{RS}}]{*} q^t$, $q^s = q^t$ implies $s = t$ from the construction of $\mathcal{A}_{\mathrm{RS}}$. We show the case of $\alpha :\ s \xrightarrow[\Delta]{*} q$ and $t \xrightarrow[\Delta]{*} q$ by induction on $|\alpha|(> 0)$.

1. In the case of $|\alpha| = 1$, we have $s \xrightarrow[\Delta]{} q$. In this case, s is the only term to transit to q from the restriction of Δ. Hence $t \xrightarrow[\Delta]{} q$ implies $s = t$.

2. (a) In the case of $|\alpha| > 1$, we consider the last transition rule applied in α is (in the form of) $f(q_1, \ldots, q_n) \xrightarrow[\Delta]{} q$. Let $s = f(s_1, \ldots, s_n)$, then we have the transition sequence $s = f(s_1, \ldots, s_n) \xrightarrow[\Delta]{*} f(q_1, \ldots, q_n) \xrightarrow[\Delta]{} q$. From the restriction of Δ, $f(q_1, \ldots, q_n)$ is only term to transit to q, and hence we have $t = f(t_1, \ldots, t_n) \xrightarrow[\Delta]{*} f(q_1, \ldots, q_n) \xrightarrow[\Delta]{} q$ by letting $t = f(t_1, \ldots, t_n)$. Since we have $s_i = t_i$ for each i from the induction hypothesis, we also have $s = t$.

   (b) In the case where $|\alpha| > 0$ and the last transition rule applied in α is (in the form of) $q'' \xrightarrow[\Delta]{} q$, we can show the lemma by the induction hypothesis. □

Lemma 36 *Let* $\alpha :\ s[\langle q, \mathtt{a}, u\rangle]_p \xrightarrow[\Delta_*]{*} \langle q', \mathtt{a}, u'\rangle$ *and* $p \in \mathit{Pos}^{\mu}(s)$. *Then* $u' \in Q^f_{\mathrm{NF}}$ *implies* $u \in Q^f_{\mathrm{NF}}$.

Proof: We show this lemma by induction on $|\alpha|$. In the case of $|\alpha| = 0$, this lemma trivially holds from $s[\langle q, \mathtt{a}, u\rangle]_p = \langle q, \mathtt{a}, u\rangle = \langle q', \mathtt{a}, u'\rangle$. Hence we suppose $|\alpha| > 0$.

1. Consider the case where the last transition rule applied in α is of the form $g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \rightarrow \langle q', \mathtt{a}, u'\rangle \in \Delta_*$. Then α can be represented as $s[\langle q, \mathtt{a}, u\rangle]_p \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow[\Delta_*]{} \langle q', \mathtt{a}, u'\rangle$.

   In this case, the position p can be represented as jp' for $1 \leq j \leq m$. From $j \in \mu(g)$, Proposition 23, and Proposition 25, we have $u'_j \in Q^f_{\mathrm{NF}}$ and hence we also have $u \in Q^f_{\mathrm{NF}}$ by the induction hypothesis.

2. In the case where the last transition rule applied in α is (in the form of) $\langle q'', \mathtt{a}, u''\rangle \rightarrow \langle q', \mathtt{a}, u'\rangle \in \Delta_*$, we have $u'' = u'$ from the construction of Δ_0 or the second inference rule of Step 2. Hence this lemma holds by the induction hypothesis. □

Lemma 37 *If $j \notin \mu(g)$ and $g(\ldots, \langle q'_j, \mathtt{x}'_j, u'_j\rangle, \ldots) \rightarrow \langle q, \mathtt{x}', u'\rangle \in \Delta_*$, then $u'_j \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}'_j = \mathtt{i}$.*

Proof: From the construction of Δ_*, we can assume that $g(\ldots, \langle q'_j, \mathtt{x}'_j, u'_j\rangle, \ldots) \rightarrow \langle q, \mathtt{x}', u'\rangle \in \Delta_k$. We show the proof of this lemma for the following two cases.

1. If $k = 0$, then $\mathtt{x}'_j = \mathtt{i}$ from the construction of Δ_0

2. In the case of $k > 0$, we can assume $g(\ldots, \langle q'_j, \mathtt{x}'_j, u'_j\rangle, \ldots) \rightarrow \langle q, \mathtt{x}', u'\rangle \in \Delta_k \backslash \Delta_{k-1}$ without loss of generality. Since this rule is introduced by the first inference rule of Step 2, we have $\mathtt{x}' = \mathtt{a}$, and there exist $f(l_1, \ldots, l_n) \rightarrow g(r_1, \ldots, r_m) \in R$ and $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \rightarrow \langle q, \mathtt{a}, u\rangle \in \Delta_{k-1}$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, and $\theta : X \rightarrow Q_*$ such that $l_i\theta \xrightarrow[\Delta_{k-1}]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$. From $j \notin \mu(g)$, we have either $\langle q'_j, \mathtt{x}'_j, u'_j\rangle = r_j\theta$ if $r_j \in X$, or $\mathtt{x}'_j = \mathtt{i}$ if $r_j \notin X$. In the case $r_j \in X$, since we have $l_i|_p = r_j$ for some i and p, we have $l_i\theta = l_i[\langle q'_j, \mathtt{x}'_j, u'_j\rangle]_p \xrightarrow[\Delta_{k-1}]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$. If $\mathtt{x}'_j = \mathtt{a}$, then we have $\mathtt{x}_i = \mathtt{a}$ by Proposition 33 and hence $u_i \in Q^f_{\mathrm{NF}}$. Therefore, $u'_j \in Q^f_{\mathrm{NF}}$ follows from Lemma 36. □

Lemma 38 *Let $\alpha :\ s[\langle q, \mathtt{x}, u\rangle]_p \xrightarrow[\Delta_k]{*} \langle q', \mathtt{x}', u'\rangle$. If $k = 0$ or $u \in Q_{NF} \setminus Q^f_{NF} \wedge \mathtt{x} = \mathtt{x}' = \mathtt{a} \wedge p \in Pos^{\mu}(s)$, then there exists $v' \in Q_{NF}$ such that $s[\langle q, \mathtt{x}, v\rangle]_p \xrightarrow[\Delta_k]{*} \langle q', \mathtt{x}', v'\rangle$ for any $v \in Q_{NF}$.*

Proof: We prove the lemma by induction on $|\alpha|$. In the case of $|\alpha| = 0$, we have $s[\langle q, \mathtt{x}, u\rangle]_p = \langle q, \mathtt{x}, u\rangle = \langle q', \mathtt{x}, u'\rangle$. Hence this lemma holds from $s[\langle q, \mathtt{x}, v\rangle]_p = \langle q, \mathtt{x}, v\rangle = \langle q', \mathtt{x}, v'\rangle$.

Following we suppose $|\alpha| > 0$.

1. Consider the case where the last transition rule applied in α is (in the form of) $g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \to \langle q', \mathtt{x}', u'\rangle \in \Delta_k$. Then α can be represented as $s[\langle q, \mathtt{x}, u\rangle]_p \xrightarrow[\Delta_k]{*} g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow[\Delta_k]{} \langle q', \mathtt{x}', u'\rangle$. Let $p = jp'$ for some $1 \leq j \leq m$.

   If $k = 0$, then the rule $g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \to \langle q, \mathtt{x}', u\rangle$ is introduced at (b) or (c) of Step 1 of (2). Thus for any $u''_j \in Q_{\mathrm{NF}}$, there exists $u'' \in Q_{\mathrm{NF}}$ such that $g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_j, \mathtt{x}'_j, u''_j\rangle, \ldots, \langle q'_n, \mathtt{x}'_n, u'_n\rangle) \to \langle q, \mathtt{x}, u''\rangle \in \Delta_0$ by the completeness of $\mathcal{A}_{\mathrm{NF}}$. Hence this lemma holds by the induction hypothesis.

   Consider the case of $k \neq 0$ and $u \in Q_{\mathrm{NF}} \setminus Q^f_{\mathrm{NF}} \wedge \mathtt{x} = \mathtt{x}' = \mathtt{a} \wedge p \in \mathrm{Pos}^{\mu}(s)$. In this case, j is in $\mu(g)$ from $p \in \mathrm{Pos}^{\mu}(s)$, and we have $\mathtt{x}'_j = \mathtt{a}$ from $\mathtt{x} = \mathtt{a}$ and Proposition 33. From $\alpha_j : (s|_j)[\langle q, \mathtt{a}, u\rangle]_{p'} \xrightarrow[\Delta_k]{*} \langle q'_j, \mathtt{a}, u'_j\rangle$, we have $(s|_j)[\langle q, \mathtt{a}, v\rangle]_{p'} \xrightarrow[\Delta_k]{*} \langle q'_j, \mathtt{a}, v'_j\rangle$ for some $v'_j \in Q_{\mathrm{NF}}$ by the induction hypothesis. Thus we have $s[\langle q, \mathtt{a}, v\rangle]_p \xrightarrow[\Delta_k]{*} g(\ldots, \langle q'_j, \mathtt{a}, v'_j\rangle, \ldots)$. Note that $u'_j \notin Q^f_{\mathrm{NF}}$ from $u \notin Q^f_{\mathrm{NF}}$ and Lemma 36.

   (a) If the transition rule in the last step of α is in Δ_0, then we also have $g(\ldots, \langle q'_j, \mathtt{a}, v'_j\rangle, \ldots) \to \langle q', \mathtt{a}, v'\rangle \in \Delta_0$ from the construction, where v' is determined by $g(\ldots, u'_{i-1}, v'_j, u'_{i+1}, \ldots) \to v' \in \Delta_{\mathrm{NF}}$.

   (b) Otherwise we assume that the transition rule in the last step of α is in $\Delta_k \setminus \Delta_{k-1}$ without loss of generality. It is known that the rule is produced by the first inference rule of Step 2. Hence there exist $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$ and $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle q', \mathtt{a}, u''\rangle \in \Delta_{k-1}$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, and $\theta : X \to Q_*$ such that $l_i\theta \xrightarrow[\Delta_k]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$ for all $1 \leq i \leq n$ where $r_j\theta = \langle q''_j, x''_j, u''_j\rangle$ for each $r_j \in X$.

   In the subcase $r_j \notin X$, we have $g(\ldots, \langle q'_j, \mathtt{a}, v'_j\rangle, \ldots) \to \langle q', \mathtt{a}, v'\rangle \in \Delta_k \setminus \Delta_{k-1}$ where $q'_j = q^{r_j}$ for some $v' \in Q_{\mathrm{NF}}$.

   In the remaining subcase $r_j \in X$, we have $l_i|_{p'} = r_j$ for some i and p'. In this case, we have $\mathtt{x}''_j = \mathtt{i}$: otherwise we have $u'_j = u''_j$ from $j \in \mu(g)$ and $\mathtt{x}_i = \mathtt{a}$ from $l_i\theta \xrightarrow[\Delta_k]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$ and Proposition 33. Hence we have $u_i \in Q^f_{\mathrm{NF}}$. Then we have $u''_j \in Q^f_{\mathrm{NF}}$ from Lemma 36. This contradicts

$u'_j = u''_j$ and $u'_j \notin Q^f_{\mathrm{NF}}$. Since $r_j \in X$, $j \in \mu(g)$, and $\mathtt{x}''_j = \mathtt{i}$, we have $g(\ldots, \langle q'_j, \mathtt{a}, v'_j \rangle, \ldots) \to \langle q', \mathtt{a}, v' \rangle \in \Delta_k \setminus \Delta_{k-1}$ for some $v' \in Q_{\mathrm{NF}}$ from the construction.

2. In the case where the last transition rule applied in α is (in the form of) $\langle q', \mathtt{x}', u' \rangle \to \langle q, \mathtt{x}, u \rangle \in \Delta_k$, we have $u' = u$ from the construction of Δ_0 or the second inference rule of Step 2. Hence this lemma holds from the induction hypothesis. □

Following Lemma 39 is the key lemma to prove the completeness of the algorithm $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$. The basic idea of the proof of Lemma 39 is similar to the proof of Lemma 14.

Lemma 39 *If R is a linear and right-shallow TRS, then $s \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u \rangle$ and $s \xhookrightarrow[R\ \mathrm{in}]{*\ \mu} t$ imply $t \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u' \rangle$ for some $u' \in Q_{NF}$.*

Proof: Similar to the proof of Lemma 14, we present a proof in the case where $s \xhookrightarrow[R\ \mathrm{in}]{\mu} t$, and let $s \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u \rangle$ and $s = s[l\sigma]_p \xhookrightarrow[R\ \mathrm{in}]{\mu} s[r\sigma]_p = t$ for some rewrite rule $l \to r \in R$, where $p \in \mathrm{Pos}^\mu(s)$. Then we have the transition sequence $s \xrightarrow[\Delta_*]{*} s[\langle q'', \mathtt{a}, u'' \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u \rangle$ by Lemma 30.

We show the proof in the case where the rewrite rule is of the form $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m)$ since the proof in the other case is similar. The diagram of this case is shown in Figure 3.3.

First, we have $s = s[f(l_1, \ldots, l_n)\sigma]_p \xhookrightarrow[R\ \mathrm{in}]{\mu} s[g(r_1, \ldots, r_m)\sigma]_p = t$, and then $s \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u \rangle$ is represented as $s = s[f(l_1, \ldots, l_n)\sigma]_p \xrightarrow[\Delta_*]{*} s[f(l_1, \ldots, l_n)\theta]_p \xrightarrow[\Delta_*]{*} s[f(\langle q_1, \mathtt{x}_1, u_1 \rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n \rangle)]_p \xrightarrow[\Delta_*]{} s[\langle q'', \mathtt{a}, u'' \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u \rangle$ similarly to the proof of Lemma 14. Here, for i such that $i \in \mu(f)$, $l_i\sigma$ is a normal form from Lemma 34 and hence we have $u_j \in Q^f_{\mathrm{NF}}$.Note that $u'' \notin Q^f_{\mathrm{NF}}$ since $s|_p$ is not a normal form.

Moreover, for i such that $i \notin \mu(f)$, we have $\mathtt{x}_i = \mathtt{i}$ or $u_j \in Q^f_{\mathrm{NF}}$ from Lemma 37.

Since $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$, $f(\langle q_1, \mathtt{x}_1, u_1 \rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n \rangle) \to \langle q'', \mathtt{a}, u'' \rangle \in \Delta_*$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for $1 \leq i \leq n$, and there exists θ such that $l_i\theta \xrightarrow[\Delta_*]{*} \langle q_i, \mathtt{x}_i, u_i \rangle$, there exist transition rules $g(\langle q'_1, \mathtt{x}'_1, u'_1 \rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m \rangle) \to \langle q'', \mathtt{a}, v' \rangle \in \Delta_*$ such that

- $q'_j = \begin{cases} q''_j & \cdots \text{ if } r_j \in X \\ q^{r_j} & \cdots \text{ if } r_j \notin X \end{cases}$

- $\mathtt{x}'_j = \begin{cases} \mathtt{x}''_j & \cdots \text{ if } j \notin \mu(g) \wedge r_j \in X \\ \mathtt{i} & \cdots \text{ if } j \notin \mu(g) \wedge r_j \notin X \\ \mathtt{a} & \cdots \text{ if } j \in \mu(g) \end{cases}$

- $u'_j = \begin{cases} u''_j & \cdots \text{ if } r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a}) \\ v \in Q_{\mathrm{NF}} & \cdots \text{ if } r_j \in X \Rightarrow (j \in \mu(g) \wedge \mathtt{x}''_j = \mathtt{i}) \end{cases}$

where $r_j\theta = \langle q''_j, \mathtt{x}''_j, u''_j \rangle$.

Here we have $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j, u'_j \rangle$ for $r_j \notin X$ from $q'_j = q^{r_j}$ and Lemma 31. Moreover, for j such that $r_j \in X$, since we have $q'_j = q''_j$ and we can take $u'_j = u''_j$, we have $r_j\sigma \xrightarrow[\Delta_*]{*} r_j\theta = \langle q'_j, \mathtt{x}''_j, u'_j \rangle$. In the case of $\mathtt{x}''_j = \mathtt{i}$, we have $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{i}, u'_j \rangle = r_j\theta$. Therefore, we also have $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{a}, u'_j \rangle$ by Proposition 29 and hence $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j, u'_j \rangle$ in either case of $\mathtt{x}'_j = \mathtt{a}$ and $\mathtt{x}'_j = \mathtt{i}$. In the case of $\mathtt{x}''_j = \mathtt{a}$, we have $\mathtt{x}''_j = \mathtt{a}$ and hence $r_j\sigma \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}''_j, u'_j \rangle$.

It follows from Lemma 38 and $s[\langle q'', \mathtt{a}, u'' \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u \rangle$ where $u'' \notin Q^f_{\mathrm{NF}}$ that $s[\langle q'', \mathtt{a}, v' \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u' \rangle$ for some $u' \in Q_{\mathrm{NF}}$. Therefore we have $t = s[g(r_1, \ldots, r_m)\sigma]_p \xrightarrow[\Delta_*]{*} s[g(\langle q'_1, \mathtt{x}'_1, u'_1 \rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m \rangle)]_p \xrightarrow[\Delta_*]{} s[\ \langle q'', \mathtt{a}, v' \rangle]_p \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u' \rangle$. □

From Lemma 39, the completeness of the algorithm is shown.

Lemma 40 *If R is a linear and right-shallow TRS, then $\mathcal{L}(\mathcal{A}_*) \supseteq \hookrightarrow^{\mu}_{R\,\mathrm{in}}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let $s \overset{*}{\hookrightarrow}{}^{\mu}_{R\,\mathrm{in}}\ t$ and $s \xrightarrow[\Delta]{*} q \in Q^f$. Since $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u \rangle \in Q^f_*$ from Proposition 25, we have $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{a}, u \rangle \in Q^f_*$ by Proposition 29. Hence $t \xrightarrow[\Delta_*]{*} \langle q, \mathtt{a}, u' \rangle \in Q^f_*$ for some $u' \in Q_{\mathrm{NF}}$ by Lemma 39. □

The following lemma is the key lemma to prove the soundness of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$. Similarly to the proof of Lemma 39, the basic idea is similar in the case of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$.

Lemma 41 *If R is linear and right-shallow Then α : $t \xrightarrow[\Delta_*]{*} \langle q, \mathtt{x}, u' \rangle$ implies $s \overset{*}{\hookrightarrow}{}^{\mu}_{R\,\mathrm{in}}\ t$ and β : $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{x}, u \rangle$ for some term s and $u \in Q_{\mathrm{NF}}$.*

$$\begin{tikzcd}
s[f(l_1, \ldots, l_n)\theta]_p \arrow[r, "*", "\Delta_*"'] & s[f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle)]_p \arrow[d, "\Delta_*", "*"'] & \\
 & s[\langle q'', \mathtt{a}, u''\rangle]_p \arrow[r, "*", "\Delta_*"'] & \langle q, \mathtt{a}, u\rangle \\
s = s[f(l_1, \ldots, l_n)\sigma]_p \arrow[uu, "\Delta_*"', "*"] \arrow[d, hook, "R", "\mu"' , "\text{in}"] & s[\langle q'', \mathtt{a}, v'\rangle]_p \arrow[r, "*", "\Delta_*"'] & \langle q, \mathtt{a}, u'\rangle \\
t = s[g(r_1, \ldots, r_m)\sigma]_p \arrow[r, "*", "\Delta_*"'] & s[g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle)]_p \arrow[u, "\Delta_*"', "*"] &
\end{tikzcd}$$

Figure 3.3: The diagram of the proof of Lemma 39

Proof: Similarly to the proof of Lemma 16, we show the detailed proof in the case of $\mathtt{x} = \mathtt{a}$ and t is of the form $g(t_1, \ldots, t_m)$ and hence α is represented as $g(t_1, \ldots, t_m) \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow[\Delta_k]{} \langle q, \mathtt{a}, u'\rangle$. The proofs of the other cases is easier or similar. We show by induction on $||\alpha||$ with respect to $\sqsupset$. Let $\alpha_j : t_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j, u'_j\rangle$. In the subcase $k = 0$, from Proposition 27, there exists $s_j \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\text{in}} t_j$ and $s_j \xrightarrow[\Delta_0]{*} \langle q'_j, \mathtt{x}'_j, v'_j\rangle$ for $1 \leq j \leq m$ and some v'_j similarly in the same part of the proof of Lemma 16. Moreover, we also have the transition rule $g(\langle q'_1, \mathtt{x}'_1, v'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, v'_m\rangle) \xrightarrow[\Delta_0]{} \langle q, \mathtt{a}, v\rangle$ for some v from the construction of Δ_0. Thus, we have $g(s_1, \ldots, s_m) \xrightarrow[\Delta_0]{*} g(\langle q'_1, \mathtt{x}'_1, v'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, v'_m\rangle) \xrightarrow[\Delta_0]{} \langle q, \mathtt{a}, v\rangle$.

In the subcase $k > 0$, we assume that the transition rule in the last step of α is in $\Delta_k \backslash \Delta_{k-1}$ without loss of generality. The diagram of this case is shown in Figure 3.4. Since this rule is produced by the first inference rule at Step 2, there exist $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$, $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle q, \mathtt{a}, v\rangle \in \Delta_{k-1}$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, and $\theta : X \to Q_*$ such that

- $\mathcal{L}(\Delta_{k-1}, x\theta) \neq \emptyset$ for all $x \in \mathrm{Var}(f(l_1, \ldots, l_n))$,
- $l_i\theta \xrightarrow[\Delta_{k-1}]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$.

Then, $\langle q'_j, \mathtt{x}'_j, u'_j\rangle$ is given as:

- $q'_j = \begin{cases} q''_j & \cdots \text{ if } r_j \in X \\ q^{r_j} & \cdots \text{ if } r_j \notin X \end{cases}$

- $\mathtt{x}'_j = \begin{cases} \mathtt{x}''_j & \cdots \text{ if } j \notin \mu(g) \wedge r_j \in X \\ \mathtt{i} & \cdots \text{ if } j \notin \mu(g) \wedge r_j \notin X \\ \mathtt{a} & \cdots \text{ if } j \in \mu(g) \end{cases}$

- $u'_j = \begin{cases} u''_j & \cdots \text{ if } r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a}) \\ v \in Q_{\mathrm{NF}} & \cdots \text{ if } r_j \in X \Rightarrow (j \in \mu(g) \wedge \mathtt{x}''_j = \mathtt{i}) \end{cases}$

where $r_j\theta = \langle q''_j, \mathtt{x}''_j, u''_j\rangle$ for $r_j \in X$.

In the following, we show that for each j, there exists s_j such that $s_j \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\text{in}}\, t_j$, $\alpha'_j :\ s_j \xrightarrow[\Delta_*]{*} r_j\theta$ and $s_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j, v'_j\rangle$ with $\alpha_j \sqsupseteq \alpha'_j$ for some $v'_j \in Q_{\mathrm{NF}}$. Similarly to the proof of Lemma 16, we can take s_j as t_j if $r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a})$ or $r_j \notin X \wedge j \notin \mu(g)$, and there exists s_j such that $s_j \xrightarrow[\Delta_0]{*} \langle q'_j, \mathtt{x}'_j\rangle$ if $r_j \notin X \wedge j \in \mu(g)$ from Proposition, 25, 26 and 28. Therefore we assume that $r_j \in X \wedge (j \in \mu(g) \wedge \mathtt{x}''_j = \mathtt{i})$. In this case, $s_j \xrightarrow[\Delta_0]{*} \langle q'_j, \mathtt{x}'_j, v'_j\rangle = \langle q''_j, \mathtt{a}, v'_j\rangle$ and $s_j \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\text{in}}\, t_j$ for some $v'_j \in Q_{\mathrm{NF}}$ from the induction hypothesis. Since $\mathcal{L}(\Delta_{k-1}, r_j\theta) \neq \emptyset$, there exists a term s''_j such that $s''_j \xrightarrow[\Delta_*]{*} \langle q''_j, \mathtt{i}, u''_j\rangle = r_j\theta$. Then we have $s''_j \xrightarrow[\Delta_0]{*} \langle q''_j, \mathtt{i}, u''_j\rangle$ from Proposition 27 and $s_j \xrightarrow[\Delta_0]{*} \langle q''_j, \mathtt{i}, v'_j\rangle$ from Proposition 28. Hence we have $s_j = s''_j$ from Lemma 35 and $u''_j = v'_j$ from determinacy of Δ_{NF}. Thus we have $\alpha'_j : s_j \xrightarrow[\Delta_0]{*} r_j\theta = \langle q'_j, \mathtt{x}'_j, v'_j\rangle$ and obviously $\alpha_j \sqsupseteq \alpha'_j$.

Thus we have $g(s_1, \ldots, s_m) \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\text{in}}\, g(t_1, \ldots, t_m)$, $g(s_1, \ldots, s_m) \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1, v'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, v'_m\rangle)$, and $g(s_1, \ldots, s_m) \xrightarrow[\Delta_*]{*} g(r_1\theta, \ldots, r_m\theta)$. Since v'_j is u'_j for j such that $r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a})$, we also have $g(\langle q'_1, \mathtt{x}'_1, v'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, v'_m\rangle) \xrightarrow[\Delta_k]{} \langle q, \mathtt{a}, v'\rangle$ from the definition of u'_j of Step 2. Let $\alpha' : g(s_1, \ldots, s_m) \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1, v'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, v'_m\rangle) \xrightarrow[\Delta_k]{} \langle q, \mathtt{a}, v'\rangle$. Note that $\alpha \sqsupseteq \alpha'$ holds from $\alpha_j \sqsupseteq \alpha'_j$ for $1 \leq j \leq m$.

We define a substitution $\sigma : \mathrm{Var}(f(l_1, \ldots, l_n)) \to \mathcal{T}(F)$ similarly to the proof of Lemma 16. Then, we have $g(s_1, \ldots, s_m) = g(r_1, \ldots, r_m)\sigma$ and $\beta : f(l_1, \ldots, l_n)\sigma \xrightarrow[\Delta_*]{*} f(l_1, \ldots, l_n)\theta \xrightarrow[\Delta_{k-1}]{*} f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \xrightarrow[\Delta_{k-1}]{} \langle q, \mathtt{a}, v\rangle$.

Since $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$, $l_i\sigma$ is a normal form or $i \notin \mu(f)$ for each i from the construction of Δ_*. Hence we have $f(l_1, \ldots, l_n)\sigma \underset{R}{\hookrightarrow}{}^{\mu}_{\text{in}}\, g(r_1, \ldots, r_m)\sigma =$

$$
\begin{tikzcd}
g(t_1,\ldots,t_m) \arrow[r, "*", "\Delta_*"'] \arrow[dr, phantom, "\text{I.H.}"] & g(\langle q'_1,\mathtt{x}'_1,u'_1\rangle,\ldots,\langle q'_m,\mathtt{x}'_m,u'_m\rangle) \arrow[r, "\Delta_k\setminus\Delta_{k-1}"'] & \langle q,\mathtt{a},u'\rangle \\
g(s_1,\ldots,s_m) = g(r_1,\ldots,r_m)\sigma \arrow[u, hook, "\mu\ \mathrm{in}", "*\ R"'] \arrow[r, "*", "\Delta_*"'] & g(\langle q'_1,\mathtt{x}'_1,v'_1\rangle,\ldots,\langle q'_m,\mathtt{x}'_m,v'_m\rangle) \arrow[r, "\Delta_k"'] & \langle q,\mathtt{a},v'\rangle \\
f(l_1,\ldots,l_n)\sigma \arrow[u, hook, "\mu\ \mathrm{in}", "R"'] \arrow[r, "*", "\Delta_*"'] \arrow[dr, phantom, "\text{I.H.}"] & f(\langle q_1,\mathtt{x}_1,u_1\rangle,\ldots,\langle q_n,\mathtt{x}_n,u_n\rangle) \arrow[r, "\Delta_{k-1}"'] & \langle q,\mathtt{a},v\rangle \\
s \arrow[u, hook, "\mu\ \mathrm{in}", "*\ R"'] \arrow[rr, "*", "\Delta_0"'] & & \langle q,\mathtt{a},u\rangle
\end{tikzcd}
$$

Figure 3.4: The diagram of the proof of Lemma 41

$g(s_1,\ldots,s_m)$. Here $\alpha \sqsupseteq \alpha' \sqsupset \beta$ follows from the left-linearity of R. Hence we have $s \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}} f(l_1,\ldots,l_n)\sigma \underset{R}{\hookrightarrow}{}^{\mu}_{\mathrm{in}} g(s_1,\ldots,s_m) \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}} g(t_1,\ldots,t_m) = t$ and $s \xrightarrow[\Delta_0]{*} \langle q,\mathtt{a},u\rangle$ for some u by the induction hypothesis. □

From Lemma 41, the soundness of the algorithm is shown.

Lemma 42 *If R be a linear and right-shallow TRS, then $\mathcal{L}(\mathcal{A}_*) \subseteq \underset{R}{\hookrightarrow}{}^{\mu}_{\mathrm{in}}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let $t \xrightarrow[\Delta_*]{*} \langle q,\mathtt{x},u'\rangle \in Q^f_*$ then there exists s such that $s \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}} t$ and $s \xrightarrow[\Delta_0]{*} \langle q,\mathtt{x},u\rangle \in Q^f_*$ by Lemma 41. Since $s \xrightarrow[\Delta_0]{*} \langle q,\mathtt{i},u\rangle$ by Proposition 28, we have $s \xrightarrow[\Delta]{*} q \in Q^f$ by Proposition 25 and the construction of Δ_0. □

Finally, we have the following theorem from Lemma 40 and 42.

Theorem 43 *Reachability is decidable for linear and right-shallow TRSs with respect to the context-sensitive innermost reduction.*

3.3 On Effective preservation of regularity

In this subsection, we give an algorithm $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ that construct a TA recognizing the set of reachable terms from the terms in a regular tree language specified by an input automaton. While input automata of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ has the restriction, we can input arbitrary automata to $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$. By $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$, it turns out that the context-sensitive innermost reduction of a linear and right-shallow TRSs effectively preserves regularity.

Difference from $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is a preprocess for initializing step. In the preprocess, we augment a state of $\mathcal{A}_{\mathrm{NF}}$ to states of an output TA. It is useful to remember what a term is reached from.

The concrete description of $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ is as follows:

Algorithm $\mathbf{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$

Input A TA $\mathcal{A} = \langle Q, Q^f, \Delta \rangle$, a linear and right-shallow TRS R, and a replacement map μ.

Output A TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_* \rangle$ such that $\mathcal{L}(\mathcal{A}_*) = \underset{R}{\hookrightarrow}{}^{\mu}_{\mathrm{in}}[\mathcal{L}(\mathcal{A})]$.

Step 1 (initialize)

1. Prepare $\mathcal{A}_{\mathrm{RS}}$ and $\mathcal{A}_{\mathrm{NF}}$ (whose constructions similar in $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$).
2. **(Preprocess:)** Prepare $\mathcal{A}' = \langle Q', Q'^f, \Delta' \rangle$ as follows:
   - $Q' := (Q \uplus Q_{\mathrm{RS}}) \times Q_{\mathrm{NF}}$,
   - $Q'^f := Q^f \times Q_{\mathrm{NF}}$, and
   - Δ' be
     - $\langle p, u \rangle \to \langle p', u \rangle \in \Delta'$ where $p' \to p \in \Delta$ and $u \in Q_{\mathrm{NF}}$, and
     - $f(\langle p_1, u_1 \rangle, \ldots, \langle p_n, u_n \rangle) \to \langle p, u \rangle \in \Delta'$ where $f(p_1, \ldots, p_n) \to p \in \Delta \cup \Delta_{\mathrm{RS}}$ and $f(u_1, \ldots, u_n) \to u \in \Delta_{\mathrm{NF}}$.
3. This step is similar to the 2 of Step 1 of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ except that Δ in the 2 of Step 1 of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is Δ' in this algorithm.

Step 2 Similarly to the Step 2 of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$.

Step 3 If $\Delta_{k+1} = \Delta_k$ then stop and set $\Delta_* = \Delta_k$. Otherwise, $k := k+1$, and go to step 2. $\square$

Example 44 Let us follow how procedure $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ works. Consider the TRS R, the replacement map μ, and the tree automaton $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ as follows:

$$\begin{aligned}
&R = \{f(g(x)) \to x, g(b) \to e, a \to d, b \to e\}\,,\\
&\mu(g) = \emptyset, \mu(f) = \{1\},\\
&Q^f = \{q^{f(g(a|b))}\},\\
&\Delta = \left\{a \to q^{(a|b)}, b \to q^{(a|b)}, g(q^{(a|b)}) \to q^{g(a|b)}, f(q^{g(a|b)}) \to q^{f(g(a|b))}\right\},
\end{aligned}$$

and hence $\mathcal{L}(\mathcal{A}) = \{f(g(a)), f(g(b))\}$.

At 1 of Step 1, $\Delta_{\mathrm{RS}} = \emptyset$ is enough since $\mathrm{RS}(R) = \emptyset$, and the following $\mathcal{A}_{\mathrm{NF}}$ satisfies the required condition.

$$\begin{aligned}
&Q_{\mathrm{NF}} = \{u_\perp, u_b, u_g, \overline{u_\perp}, \overline{u_g}\},\\
&Q^f_{\mathrm{NF}} = \{\overline{u_\perp}, \overline{u_g}\},\\
&\Delta_{\mathrm{NF}} = \left\{\begin{array}{rclrclrclrcl}
a & \to & u_\perp, & b & \to & u_b, & d & \to & \overline{u_\perp}, & e & \to & \overline{u_\perp},\\
g(u_\perp) & \to & \overline{u_g}, & g(u_b) & \to & u_g, & g(u_g) & \to & \overline{u_g}, & g(\overline{u_\perp}) & \to & \overline{u_g},\\
g(\overline{u_g}) & \to & \overline{u_g}, & f(u_\perp) & \to & u_\perp, & f(u_b) & \to & u_\perp, & f(u_g) & \to & u_\perp,\\
f(\overline{u_\perp}) & \to & \overline{u_\perp}, & f(\overline{u_g}) & \to & u_\perp & & & & & &
\end{array}\right\}
\end{aligned}$$

At 2 and 3 of Step 1, we have the following:

$$\Delta_0 = \left\{\begin{array}{rcl}
a & \to & \langle\langle q^{(a|b)}, u_\perp\rangle, \mathtt{x}, u_\perp\rangle,\\
b & \to & \langle\langle q^{(a|b)}, u_b\rangle, \mathtt{x}, u_b\rangle,\\
g(\langle\langle q^{(a|b)}, u'_1\rangle, \mathtt{i}, u_1\rangle) & \to & \langle\langle q^{g(a|b)}, \overline{u_g}\rangle, \mathtt{x}, \overline{u_g}\rangle,\\
g(\langle\langle q^{(a|b)}, u_b\rangle, \mathtt{i}, u_1\rangle) & \to & \langle\langle q^{g(a|b)}, u_g\rangle, \mathtt{x}, \overline{u_g}\rangle,\\
g(\langle\langle q^{(a|b)}, u'_1\rangle, \mathtt{i}, u_b\rangle) & \to & \langle\langle q^{g(a|b)}, \overline{u_g}\rangle, \mathtt{x}, u_g\rangle,\\
g(\langle\langle q^{(a|b)}, u_b\rangle, \mathtt{i}, u_b\rangle) & \to & \langle\langle q^{g(a|b)}, u_g\rangle, \mathtt{x}, u_g\rangle,\\
f(\langle\langle q^{g(a|b)}, u'_2\rangle, \mathtt{x}, u_2\rangle) & \to & \langle\langle q^{f(g(a|b))}, u_\perp\rangle, \mathtt{x}, u_\perp\rangle,\\
f(\langle\langle q^{g(a|b)}, \overline{u_\perp}\rangle, \mathtt{x}, u_2\rangle) & \to & \langle\langle q^{f(g(a|b))}, \overline{u_\perp}\rangle, \mathtt{x}, u_\perp\rangle,\\
f(\langle\langle q^{g(a|b)}, u'_2\rangle, \mathtt{x}, \overline{u_\perp}\rangle) & \to & \langle\langle q^{f(g(a|b))}, u_\perp\rangle, \mathtt{x}, \overline{u_\perp}\rangle,\\
f(\langle\langle q^{g(a|b)}, \overline{u_\perp}\rangle, \mathtt{x}, \overline{u_\perp}\rangle) & \to & \langle\langle q^{f(g(a|b))}, \overline{u_\perp}\rangle, \mathtt{x}, \overline{u_\perp}\rangle
\end{array}\right\}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}, u_1, u'_1 \in \{u_\perp, u_g, \overline{u_\perp}, \overline{u_g}\}, u_2, u'_2 \in \{u_\perp, u_b, u_g, \overline{u_g}\}$.

After execution of Step 2's, Δ_k saturates at $k = 2$, and we have

$$\Delta_1 = \Delta_0 \cup \left\{ \begin{array}{rcl} d & \to & \langle\langle q^{(a|b)}, u_\perp\rangle, \mathtt{a}, \overline{u_\perp}\rangle, \\ e & \to & \langle\langle q^{(a|b)}, u_b\rangle, \mathtt{a}, \overline{u_\perp}\rangle, \\ e & \to & \langle\langle q^{g(a|b)}, \overline{u_g}\rangle, \mathtt{a}, \overline{u_\perp}\rangle, \\ \langle\langle q^{(a|b)}, u_\perp\rangle, \mathtt{a}, u_\perp\rangle & \to & \langle\langle q^{f(g(a|b))}, u_\perp\rangle, \mathtt{a}, u_\perp\rangle \end{array} \right\},$$

$$\Delta_2 = \Delta_1 \cup \{\langle\langle q^{(a|b)}, u_\perp\rangle, \mathtt{a}, \overline{u_\perp}\rangle \to \langle\langle q^{f(g(a|b))}, u_\perp\rangle, \mathtt{a}, \overline{u_\perp}\rangle\},$$

$\Delta_3 = \Delta_2$, and $\Delta_* = \Delta_2$.

The output TA $\mathcal{A}_*$ recognizes the set $\{f(g(a)), f(g(b)), a, d, f(e)\}$ that is equal to $\hookrightarrow^{\mu}_{R\ \mathrm{in}} [f(g(a)), f(g(b))]$. □

This algorithm $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ eventually terminates at some k and apparently $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$ similarly to the other algorithms in this paper.

As previous noted, the difference from $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is 2 of Step 1. In Example 44, if we omit this step, transition rules $d \to \langle q^{(a|b)}, \mathtt{a}, \overline{u_\perp}\rangle$ and $e \to \langle q^{(a|b)}, \mathtt{a}, \overline{u_\perp}\rangle$ is produced instead of $d \to \langle\langle q^{(a|b)}, u_\perp\rangle, \mathtt{a}, \overline{u_\perp}\rangle$ and $e \to \langle\langle q^{(a|b)}, u_b\rangle, \mathtt{a}, \overline{u_\perp}\rangle$. Since the former transition rules does not distinguish terms d and e, the acceptances of d and e coincide. However, we must distinguish d and e because d is in $\hookrightarrow^{\mu}_{R\ \mathrm{in}}[\mathcal{L}(\mathcal{A})]$ but e is not. In fact, by modifying the transition rules of $\mathcal{A}$ in this example to meet the restriction of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$, we can construct a correct TA. It is because we can modify the input TA to fulfill the restriction of $\mathcal{A}$ by using states q_a and q_b instead of $q^{(a|b)}$. However, if input automata recognizes infinite terms, we can't modify input TA to fulfill the restriction. Thus the procedure $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is not guaranteed to construct a correct TA if input automata recognize the set of infinite terms.

On the other hand, $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ has no restriction on input automata. Thus $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ is guaranteed to construct a correct TA if input automata recognizes the set of infinite terms. However, $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ is not better than $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ on complexity. Obviously, the cardinality of the set of states of an output TA of $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ is more than that of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$. Therefore, if we only want to check reachability, $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is better.

The completeness of $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$ can be proved similarly to $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$. Therefore, in the following, we show the proof of the soundness of $\mathrm{P}'^{\mathrm{lrs}}_{\mathrm{csin}}$. At first, we show several propositions and lemma. These are necessary to prove soundness. We omit the proofs of some propositions because they are same as those in the previous subsection.

Proposition 45 *(a) Let $s \in \mathcal{T}(F)$, then $s \xrightarrow[\Delta]{*} p \in Q$ if and only if $s \xrightarrow[\Delta']{*} \langle p, u\rangle$ for some $u \in Q_{NF}$.*

(b) Let $s \in \mathcal{T}(F)$, then $s \xrightarrow[\Delta']{*} q \in Q'$ if and only if $s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u\rangle$ for some $u \in Q_{NF}$.

Proof: The claim is a direct consequence of the construction of Δ' and Δ_0. □

Proposition 46 *(a) $f(\langle p_1, u_1\rangle, \ldots, \langle p_n, u_n\rangle) \to \langle p, u\rangle \in \Delta'$ implies $f(u_1, \ldots, u_n) \to u \in \Delta_{\mathrm{NF}}$.*

(b) $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle q, \mathtt{x}, u\rangle \in \Delta_*$ implies $f(u_1, \ldots, u_n) \to u \in \Delta_{\mathrm{NF}}$.

Proof: From the construction of Δ' and Δ_*. □

Proposition 47 *Let $t \in \mathcal{T}(F)$, $q \in Q'$, and $u \in Q_{\mathrm{NF}}$, then $t \xrightarrow[\Delta_k]{*} \langle q, \mathtt{i}, u\rangle$ implies $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u\rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 27. □

Proposition 48 *Let $t \in \mathcal{T}(F)$, $q \in Q'$, and $u \in Q_{\mathrm{NF}}$. Then, $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{a}, u\rangle$ if and only if $t \xrightarrow[\Delta_0]{*} \langle q, \mathtt{i}, u\rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 28. □

Lemma 49 *If $s \in \mathcal{T}(F)$, $\alpha : \ s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{x}, u\rangle$, and $q = \langle p, u'\rangle$, then $s \xrightarrow[\Delta_{\mathrm{NF}}]{*} u$, $s \xrightarrow[\Delta_{\mathrm{NF}}]{*} u'$, and $u = u'$.*

Proof: From (b) of Proposition 45, we have $s \xrightarrow[\Delta']{*} q = \langle p, u'\rangle$. Hence this lemma obviously holds from determinacy of $\mathcal{A}_{\mathrm{NF}}$ by applying Proposition 46 repeatedly. □

Following Lemma 50 is the key lemma of the proof of the soundness.

Lemma 50 *If R is a linear and right-shallow TRS, then $\alpha : \ t \xrightarrow[\Delta_*]{*} \langle q, \mathtt{x}, u'\rangle$ implies $s \overset{*}{\underset{R}{\hookleftarrow}}{}^{\mu}_{\mathrm{in}} t$ and $\beta : \ s \xrightarrow[\Delta_0]{*} \langle q, \mathtt{x}, u\rangle$ for some term s and $u \in Q_{\mathrm{NF}}$.*

The proof of this lemma is very similar to Lemma 41.

Proof: Similarly to the proof of Lemma 41, we show the detailed proof in the case of $\mathtt{x} = \mathtt{a}$ and t is of the form $g(t_1, \ldots, t_m)$ and hence α is represented as $g(t_1, \ldots, t_m) \xrightarrow[\Delta_*]{*} g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow[\Delta_k]{} \langle q, \mathtt{a}, u'\rangle$. The proofs of the other cases is easier or similar. We show by induction on $||\alpha||$ with respect to $\sqsupset$. Let $\alpha_j : t_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j, u'_j\rangle$.

In the subcase of $k = 0$, from Proposition 47, this lemma is proved similarly to the proof of Lemma 41.

In the subcase of $k > 0$, we assume that the transition rule in the last step of α is in $\Delta_k \backslash \Delta_{k-1}$ without loss of generality. Since this rule is introduced at 1 of Step 2, there exist $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$, $f(\langle q_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle q_n, \mathtt{x}_n, u_n\rangle) \to \langle q, \mathtt{a}, v\rangle \in \Delta_{k-1}$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, and $\theta : X \to Q_*$ such that

- $\mathcal{L}(\Delta_{k-1}, x\theta) \neq \emptyset$ for all $x \in \mathrm{Var}(f(l_1, \ldots, l_n))$,
- $l_i\theta \xrightarrow[\Delta_{k-1}]{*} \langle q_i, \mathtt{x}_i, u_i\rangle$,

and each $\langle q'_j, \mathtt{x}'_j, u'_j\rangle$ is given as follows:

- $q'_j = \begin{cases} q''_j & \cdots \text{ if } r_j \in X \\ q^{r_j} & \cdots \text{ if } r_j \notin X \end{cases}$
- $\mathtt{x}'_j = \begin{cases} \mathtt{x}''_j & \cdots \text{ if } j \notin \mu(g), r_j \in X \\ \mathtt{i} & \cdots \text{ if } j \notin \mu(g), r_j \notin X \\ \mathtt{a} & \cdots \text{ if } j \in \mu(g) \end{cases}$
- $u'_j = \begin{cases} u''_j & \cdots \text{ if } r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a}) \\ v \in Q_{\mathrm{NF}} & \cdots \text{ if } r_j \in X \Rightarrow (j \in \mu(g) \wedge \mathtt{x}''_j = \mathtt{i}) \end{cases}$

where $r_j\theta = \langle q''_j, \mathtt{x}''_j, u''_j\rangle$ for $r_j \in X$.

Here, we have to show that for each j, there exists s_j such that $s_j \xhookrightarrow[R]{*}{}^{\mu}_{\mathrm{in}}\, t_j$, $s_j \xrightarrow[\Delta_*]{*} r_j\theta$ and $\alpha'_j : s_j \xrightarrow[\Delta_*]{*} \langle q'_j, \mathtt{x}'_j, v'_j\rangle$ with $\alpha_j \sqsupseteq \alpha'_j$ for some $v'_j \in Q_{\mathrm{NF}}$. We can prove this claim for j such that $r_j \notin X$ or $r_j \in X \wedge (j \in \mu(g) \Rightarrow \mathtt{x}''_j = \mathtt{a})$ similarly to the proof of Lemma 41. from using Proposition 45, 47, and 48.

We describe the detail of the proof only in the case where $r_j \in X \wedge (j \in \mu(g) \wedge \mathtt{x}''_j = \mathtt{i})$ in the following. In this case, we have a term s_j such that $s_j \xrightarrow[\Delta_0]{*}$

$\langle q'_j, \mathtt{x}'_j, v'_j\rangle = \langle q''_j, \mathtt{a}, v'_j\rangle$ and $s_j \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\text{in}} t_j$ for some $v'_j \in Q_{\text{NF}}$ from the induction hypothesis. Since $\mathcal{L}(\Delta_{k-1}, r_j\theta) \neq \emptyset$ from Step 2 of $\mathrm{P}'^{\text{lrs}}_{\text{csin}}$, we also have a term s'_j such that $s'_j \xrightarrow[\Delta_*]{*} \langle q''_j, \mathtt{i}, u''_j\rangle = r_j\theta$. From Proposition 47, we have $s'_j \xrightarrow[\Delta_0]{*} \langle q''_j, \mathtt{i}, u''_j\rangle = r_j\theta$, and from Proposition 48, we also have $s_j \xrightarrow[\Delta_0]{*} \langle q''_j, \mathtt{i}, v'_j\rangle$. Hence we have $v'_j = u''_j$ from Lemma 49, and this implies $\alpha'_j : s_j \xrightarrow[\Delta_0]{*} r_j\theta = \langle q'_j, \mathtt{x}'_j, v'_j\rangle$. Obviously, $\alpha_j \sqsupseteq \alpha'_j$ holds.

Since the following part of this proof is similar to Lemma 41, we omit it. □

Lemma 51 shows the soundness of procedure $\mathrm{P}'^{\text{lrs}}_{\text{csin}}$.

Lemma 51 *If R is a linear and right-shallow TRS, then $\mathcal{L}(\mathcal{A}_*) \subseteq \underset{R}{\hookrightarrow}{}^{\mu}_{\text{in}}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let $t \xrightarrow[\Delta_*]{*} \langle\langle p, u\rangle, \mathtt{x}, u'\rangle \in Q^f_*$ then we have $s \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\text{in}} t$ and $s \xrightarrow[\Delta_0]{*} \langle\langle p, u\rangle, \mathtt{x}, u\rangle \in Q^f_*$ by Lemma 50. Since $s \xrightarrow[\Delta_0]{*} \langle\langle p, u\rangle, \mathtt{i}, u\rangle \in Q^f_*$ by Proposition 28, we have $s \xrightarrow[\Delta]{*} p \in Q^f$ by Proposition 46 (a) and (b). □

The proof of Lemma 40 also works for a proof of the completeness of $\mathrm{P}'^{\text{lrs}}_{\text{csin}}$. Thus we can construct a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = \underset{R}{\hookrightarrow}{}^{\mu}_{\text{in}}[\mathcal{L}(\mathcal{A})]$ for any TA $\mathcal{A}$, which leads the main theorem.

Theorem 52 *The context-sensitive innermost reduction of a linear and right-shallow TRSs effectively preserves regularity.*

3.4 On the Innermost Reduction

Innermost reductions are particular cases of context-sensitive innermost reductions, i.e. a context-sensitive innermost reduction in which a replacement map μ is given as $\mu(f) = \{1, \ldots, \text{ar}(f)\}$ for all $f \in F$ is an innermost reduction. Therefore, $\mathrm{P}^{\text{lrs}}_{\text{csin}}$ and $\mathrm{P}'^{\text{lrs}}_{\text{csin}}$ works correctly for innermost reductions and the following corollary is derived from Theorem 52.

Corollary 53 *The innermost reduction for linear and right-shallow TRSs effectively preserves regularity. Thus reachability is decidable for linear and right-shallow TRSs with respect to innermost reductions.*

Note that this corollary is also derived in [10] independently.

Remark In fact, we proved the above corollary in [21] without Theorem 43 or 52. Similarly to the case of context-sensitive innermost, the proof is to show the algorithm that outputs a TA recognizing the set of reachable terms. The algorithm is almost equivalent to $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ except for augmenting parameter `a` or `i`. Since `a` and `i` are introduced to adopt context-sensitive, it is not necessary for the innermost case. Moreover, the algorithm for the innermost case can deal with an arbitrary TA as an input, while an input of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ is restricted. It means that the simple merge of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$ and the algorithm for the innermost case is insufficient for context-sensitive innermost, and hence studies about context-sensitive innermost is not an easy topic.

Chapter 4

Decidability of Reachability for Right-Linear and Right-Shallow TRSs

With respect to the ordinary reduction, it was shown that reachability for right-linear and right-shallow TRSs is decidable in [24]. However, similarly to linear and right-shallow TRSs, it has been an open problem whether reachability is decidable or undecidable for the context-sensitive reduction. In this chapter, we show that reachability is decidable for right-linear and right-shallow TRSs with respect to the context-sensitive reduction.

4.1 On the Context-Sensitive Reduction

In this section, we prove that reachability for right-linear and right-shallow TRSs is decidable with respect to the context-sensitive reduction. To this end, we show the algorithm $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ that constructs a TA recognizing the set of terms reachable by a right-linear and right-shallow TRS from an input term.

The algorithm $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ is a modification of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$. $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ can only deal with linear and right-shallow TRSs. To deal with the left-non-linear system, the power set of the set of states of input TA is used as the set of states of output TA in [24] and [32]. $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ employs this idea, thus each state of the output TA of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ is of the form $\langle\{q_1,\ldots,q_n\},\mathtt{x}\rangle$ where $q_1,\ldots,q_n$ are states of the input TA and $\mathtt{x}\in\{\mathtt{a},\mathtt{i}\}$. We show a benefit of this idea in Example 54.

Remark: Since right-linear and right-shallow is a super class of linear and right-shallow, decidability of reachability for linear and right-shallow TRSs with respect to the context-sensitive reduction can be proved by the result in this section. However, the result in this chapter does not properly includes the result in Section 1 of Chapter 1. It is because Section 1 of Chapter 1 shows effective preservation of regularity while this section only shows decidability of reachability. Maybe, we can prove effective preservation of regularity for linear and right-shallow TRSs with respect to the context-sensitive reduction by introducing some modifications to $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$, but since $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ is easier and simpler than $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$, we show the results for linear and right-shallow TRSs previously.

Since the algorithm $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ is more complex than the algorithms in Chapter 3, we show an example of automata construction in the following where it can be seen that the TA obtained by $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ recognizes the set of terms reachable from an input term correctly.

Example 54 Let the the TRS R, the replacement map μ, and the TA $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ as follows:

$$\begin{aligned}
&R = \{a\to b, b\to d, c\to d, f(x,x)\to g(x,c), g(x,x)\to h(x)\}\,,\\
&\mu(f)=\{1\}, \mu(g)=\{1,2\}, \mu(h)=\emptyset,\\
&Q=\{q^a,q^b,q^c,q^{f(a,b)}\},\\
&Q^f=\{q^{f(a,b)}\},\\
&\Delta=\{a\to q^a, b\to q^b, c\to q^c, f(q^a,q^b)\to q^{f(a,b)}\},
\end{aligned}$$

and hence $\mathcal{L}(\mathcal{A})=\{f(a,b)\}$.

$\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ output the automaton $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_*\rangle$ that recognizes $\overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}[\{f(a,b)\}]$. Each component of $\mathcal{A}_*$ is obtained as follows:

$$\begin{aligned}
Q_* &= \{\langle P,\mathtt{a}\rangle, \langle\{p\},\mathtt{i}\rangle \mid P\in 2^Q, P\neq\emptyset, p\in Q\},\\
Q^f_* &= \{\langle P^f,\mathtt{a}\rangle \mid P^f\in 2^Q, P^f\cap Q^f\neq\emptyset\},
\end{aligned}$$

$$\Delta_* = \left\{ \begin{array}{rcl} a & \to & \langle \{q^a\}, \mathtt{x} \rangle, \\ b & \to & \langle \{q^b\}, \mathtt{x} \rangle, \\ b & \to & \langle \{q^a\}, \mathtt{a} \rangle, \\ c & \to & \langle \{q^c\}, \mathtt{x} \rangle, \\ d & \to & \langle P_1, \mathtt{a} \rangle, \\ f(\langle \{q^a\}, \mathtt{x} \rangle, \langle \{q^b\}, \mathtt{i} \rangle) & \to & \langle \{q^{f(a,b)}\}, \mathtt{x} \rangle, \\ g(\langle \{q^b\}, \mathtt{a} \rangle, \langle \{q^c\}, \mathtt{a} \rangle) & \to & \langle \{q^{f(a,b)}\}, \mathtt{a} \rangle, \\ h(\langle \{q^b, q^c\}, \mathtt{a} \rangle) & \to & \langle \{q^{f(a,b)}\}, \mathtt{a} \rangle \} \end{array} \right\}$$

where $P_1 \subseteq \{q^a, q^b, q^c\}$ and $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$. We obtain $\mathcal{L}(\mathcal{A}_*) = \{f(a,b), f(b,b), f(d,b), g(b,c), g(d,c), g(b,d), g(d,d), h(d)\} = \underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A})]$ □

Since this example is complex, we describe it here. At first, we obtain Q_* by augmenting parameter $\mathtt{a}$ or $\mathtt{i}$ to each state and taking powerset of Q for the first components of the states. From the set of transition rules Δ_*, it can be seen that $\langle \{q^a\}, \mathtt{i} \rangle$, $\langle \{q^b\}, \mathtt{i} \rangle$, $\langle \{q^c\}, \mathtt{i} \rangle$, and $\langle \{q^{f(a,b)}\}, \mathtt{i} \rangle$ only accept the terms in $\mathcal{L}(\mathcal{A}, q^a)$, $\mathcal{L}(\mathcal{A}, q^b)$, $\mathcal{L}(\mathcal{A}, q^c)$ and $\mathcal{L}(\mathcal{A}, q^{f(a,b)})$, and $\langle \{q^a\}, \mathtt{a} \rangle$, $\langle \{q^b\}, \mathtt{a} \rangle$, $\langle \{q^c\}, \mathtt{a} \rangle$, and $\langle \{q^{f(a,b)}\}, \mathtt{a} \rangle$ accept the terms in $\underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A}, q^a)]$, $\underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A}, q^b)]$, $\underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A}, q^c)]$, $\underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A}, q^{f(a,b)})]$, respectively. From $\mu(f) = \{1\}$, the state in the second argument of f in the transition rule must have $\mathtt{i}$ as its second component. In this way, $\mathcal{A}_*$ does not accepts the terms obtained by rewriting the second argument of f. Moreover, we have $\mathcal{L}(\mathcal{A}_*, \langle \{q^b, q^c\}, \mathtt{a} \rangle) = (\mathcal{L}(\mathcal{A}_*, \langle \{q^b\}, \mathtt{a} \rangle) \cap \mathcal{L}(\mathcal{A}_*, \langle \{q^c\}, \mathtt{a} \rangle))$. Indeed, the state $\langle \{q^b, q^c\}, \mathtt{a} \rangle$ accepts only d, which is the term reachable from both b and c. Since the term that is reachable from $f(a,b)$ and matches $f(x,x)$ is only $f(b,b)$, we have to produce the transition rule $g(\langle \{q^b\}, \mathtt{a} \rangle, \langle \{q^c\}, \mathtt{a} \rangle) \to \langle \{q^{f(a,b)}\}, \mathtt{a} \rangle$ from the rewrite rule $f(x,x) \to g(x,c)$. Since the term that is reachable from $f(a,b)$ and matches $g(x,x)$ is only $g(d,d)$, we have to produce the transition rule $h(\langle \{q^b, q^c\}, \mathtt{a} \rangle) \to \langle \{q^{f(a,b)}\}, \mathtt{a} \rangle$ from the rewrite rule $g(x,x) \to h(x)$.

Concrete description of the algorithm $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ is the following.

Algorithm $\mathbf{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$:

Input A term t, a right-linear and right-shallow TRS R, and a replacement map μ.

Output The TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_* \rangle$ such that $\mathcal{L}(\mathcal{A}_*) = \underset{R}{\hookrightarrow}^{\mu}[\{t\}]$.

Step 1 (initialize)

1. Prepare a TA $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ where each state q^s accepts $s \in \{t\} \cup \mathrm{RS}(R)$. Here we assume $Q = \{q^s \mid s \trianglelefteq s', s' \in \{t\} \cup \mathrm{RS}(R)\}$, $Q^f = \{q^t\}$, and $\Delta = \{f(q^{t_1}, \ldots, q^{t_n}) \to q^{f(t_1,\ldots,t_n)} \mid f(t_1, \ldots, t_n) \trianglelefteq s, s \in \{t\} \cup \mathrm{RS}(R)\}$. Hence $\mathcal{L}(\mathcal{A}, q^s) = \{s\}$ for all q^s.
2. Let
   - $k := 0$,
   - $Q_* := \{P \mid P \in 2^Q, P \neq \emptyset\} \times \{\mathtt{a}\} \cup \{\{q\} \mid q \in Q\} \times \{\mathtt{i}\}$,
   - $Q^f_* = \{P^f \mid P^f \in 2^Q, P^f \cap Q^f \neq \emptyset\} \times \{\mathtt{a}\}$, and
   - Δ_0 consists of the following transition rules:
     - $f(\langle\{q_1\}, \mathtt{i}\rangle, \ldots, \langle\{q_n\}, \mathtt{i}\rangle) \to \langle\{q\}, \mathtt{i}\rangle$ where $f(q_1, \ldots, q_n) \to q \in \Delta$, and
     - $f(\langle\{q_1\}, \mathtt{x}_i\rangle, \ldots, \langle\{q_n\}, \mathtt{x}_n\rangle) \to \langle\{q\}, \mathtt{a}\rangle$ where $f(q_1, \ldots, q_n) \to q \in \Delta\cup$, $\mathtt{x}_i = \mathtt{a}$ if $i \in \mu(f)$, and $\mathtt{x}_i = \mathtt{i}$ otherwise.

   Note that Δ has no ε-rule from the construction.

Step 2 Let Δ_{k+1} be the set of transition rules produced by augmenting transition rules of Δ_k by the following inference rules. Let C be the context that has no variable:

If there exists $\sigma : X \to T(F)$ such that $x_i\sigma \xrightarrow[\Delta_k]{*} \langle P_i, \mathtt{x}_i\rangle$ for all $1 \leq i \leq n$[1], we apply the following inference rule:

$$\frac{C[x_1, \ldots, x_n] \to g(r_1, \ldots, r_m) \in R, C[\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle] \xrightarrow[\Delta_k]{*} \langle\{q\}, \mathtt{a}\rangle}{g(\langle P'_1, \mathtt{x}'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m\rangle) \to \langle\{q\}, \mathtt{a}\rangle \in \Delta_{k+1}}$$

Let $I_j = \{i \mid x_i = r_j\}$. Each P'_j and $\mathtt{x}'_j$ is determined as follows:

- $P'_j = \begin{cases} \{q^{r_j}\} & \cdots \text{ if } r_j \notin X, \\ P_i & \cdots \text{ if } r_j \in X \wedge \exists i \in I_j.\mathtt{x}_i = \mathtt{i}, \text{ and} \\ \bigcup_{i \in I_j} P_i & \cdots \text{ if } r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a}. \end{cases}$
- $\mathtt{x}'_j = \begin{cases} \mathtt{i} \cdots \text{ if } j \notin \mu(g) \wedge (r_j \in X \Rightarrow \exists i \in I_j.\mathtt{x}_i = \mathtt{i}), \text{ and} \\ \mathtt{a} \cdots \text{ if } j \in \mu(g) \vee (r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a})). \end{cases}$

[1]Since the rewrite rule is left-non-linear and hence there exists some $x_i = x_j$ such that $i = j$, the condition $\mathcal{L}(\Delta_k, \langle P_i, \mathtt{x}_i\rangle) \neq \emptyset$ which is similar in the algorithms in Chapter 3 is insufficient.

for all $1 \leq j \leq m$, and if there exists $\sigma : X \to T(F)$ such that $x_i\sigma \xrightarrow[\Delta_k]{*} \langle P_i, \mathtt{x}_i \rangle$ for all $1 \leq i \leq n$, we apply the following inference rule:

$$\frac{C[x_1, \ldots, x_n] \to x_i \in R, C[\langle P_1, \mathtt{x}_1 \rangle, \ldots, \langle P_n, \mathtt{x}_n \rangle] \xrightarrow[\Delta_k]{*} \langle \{q\}, \mathtt{a} \rangle}{\langle P', \mathtt{a} \rangle \to \langle \{q\}, \mathtt{a} \rangle \in \Delta_{k+1}}$$

Let $I = \{j \mid x_j = x_i\}$. P' is determined as follows:

- $P' = \begin{cases} P_i & \cdots \text{ if } \exists i \in I. \mathtt{x}_i = \mathtt{i}, \text{ and} \\ \bigcup_{i \in I} P_i & \cdots \text{ if } \forall i \in I. \mathtt{x}_i = \mathtt{a}. \end{cases}$

Step 3 For all states $\langle P^1 \cup P^2, \mathtt{a} \rangle \in Q_*$ where $P^1 \neq P^2$, we add new transition rules to Δ_{k+1} as follows:

1. $f(\langle P_1, \mathtt{x}_1 \rangle, \ldots, \langle P_n, \mathtt{x}_n \rangle) \to \langle P^1 \cup P^2, \mathtt{a} \rangle \in \Delta_{k+1}$ where
   - $P_i = \begin{cases} P_i^j & \cdots \begin{array}{l} \text{if } \mathtt{x}_i^j = \mathtt{i} \text{ for some } j \in \{1,2\} \text{ and} \\ \mathcal{L}(\Delta_k, \langle P_i^1, \mathtt{x}_i^1 \rangle) \cap \mathcal{L}(\Delta_k, \langle P_i^2, \mathtt{x}_i^2 \rangle) \neq \emptyset, \text{ and} \end{array} \\ P_i^1 \cup P_i^2 & \cdots \text{ if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a} \end{cases}$
   - $\mathtt{x}_i = \begin{cases} \mathtt{a} \cdots \text{if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a}, \text{ and} \\ \mathtt{i} \cdots \text{if otherwise.} \end{cases}$

   if $f(\langle P_1^j, \mathtt{x}_1^j \rangle, \ldots, \langle P_n^j, \mathtt{x}_n^j \rangle) \to \langle P^j, \mathtt{a} \rangle \in \Delta_k$ for $j \in \{1, 2\}$.

   Note that if $\mathcal{L}(\Delta_k, \langle P_i^1, \mathtt{x}_i^1 \rangle) \cap \mathcal{L}(\Delta_k, \langle P_i^2, \mathtt{x}_i^2 \rangle) = \emptyset$ and $\mathtt{x}_i^j = \mathtt{i}$ for some $j \in \{1, 2\}$, then the transition rule is not produced.

2. $\langle P_1' \cup P_2', \mathtt{a} \rangle \to \langle P_1 \cup P_2, \mathtt{a} \rangle \in \Delta_{k+1}$ if $\langle P_1', \mathtt{a} \rangle \to \langle P_1, \mathtt{a} \rangle \in \Delta_k$, and, $\langle P_2', \mathtt{a} \rangle \to \langle P_2, \mathtt{a} \rangle \in \Delta_k$ or $P_2' = P_2$.

Step 4 If $\Delta_{k+1} = \Delta_k$ then stop and set $\Delta_* := \Delta_k$; Otherwise $k := k + 1$ and go to step 2.

Example 55 Let us follow how the algorithm $\mathrm{P}_{\mathrm{cs}}^{\mathrm{rlrs}}$ works. We input the right-linear and right-shallow TRS R and a replacement map μ in Example 54, and the term $f(a, b)$ to $\mathrm{P}_{\mathrm{cs}}^{\mathrm{rlrs}}$.

In the initializing step, at 1 of Step 1, we construct the automaton $\mathcal{A}$ in Example 54, and at 2 of Step 1, we have

$$\begin{aligned}
&Q_* = \{\langle P, \mathtt{a}\rangle, \langle \{p\}, \mathtt{i}\rangle \mid P \in 2^Q, P \neq \emptyset, p \in Q\},\\
&Q_*^f = \{\langle P_f, \mathtt{a}\rangle \mid P_f \subseteq Q, P_f \cap Q^f \neq \emptyset\}, \text{ and}\\
&\Delta_0 = \{a \to \langle \{q^a\}, \mathtt{x}\rangle, b \to \langle \{q^b\}, \mathtt{x}\rangle, f(\langle \{q^a\}, \mathtt{x}\rangle, \langle \{q^b\}, \mathtt{i}\rangle) \to \langle \{q^{f(a,b)}\}, \mathtt{x}\rangle\}
\end{aligned}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$.

In the saturation step at $k = 0$, we produce the transition rules

$$\Delta_0 = \left\{ \begin{array}{rclrcl} b & \to & \langle \{q^a\}, \mathtt{a}\rangle, & d & \to & \langle \{q^b\}, \mathtt{a}\rangle, \\ d & \to & \langle \{q^c\}, \mathtt{a}\rangle, & g(\langle \{q^b\}, \mathtt{a}\rangle, \langle \{q^a\}, \mathtt{a}\rangle) & \to & \langle \{q^{f(a,b)}\}, \mathtt{a}\rangle \end{array} \right\}$$

at Step 2.

At $k = 1$, we produce the transition rules

$$\{d \to \langle \{q^a\}, \mathtt{a}\rangle, h(\langle \{q^b, q^c\}, \mathtt{a}\rangle) \to \langle \{q^{f(a,b)}\}, \mathtt{a}\rangle\}$$

at Step 2 and

$$\{b \to \langle \{q^a, q^b\}, \mathtt{a}\rangle, d \to \langle \{q^b, q^c\}, \mathtt{a}\rangle\}$$

at Step 3.

At $k = 2$, we produce the transition rules

$$\{d \to \langle \{q^a, q^b\}, \mathtt{a}\rangle, d \to \langle \{q^a, q^c\}, \mathtt{a}\rangle, d \to \langle \{q^a, q^b, q^c\}, \mathtt{a}\rangle\}$$

at Step 3.

The saturation steps stop at $k = 3$, and we have $\Delta_* = \Delta_3$. □

Similar to the other algorithms in this paper, $\mathrm{P}_{\mathrm{cs}}^{\mathrm{rlrs}}$ eventually terminates at some k and apparently $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$.

Remark that one of the input for $\mathrm{P}_{\mathrm{cs}}^{\mathrm{rlrs}}$ is a term while it is an arbitrary tree automaton in the procedures at [24, 32], and $\mathrm{P}_{\mathrm{cs}}^{\mathrm{lrs}}$. Otherwise, $\mathrm{P}_{\mathrm{cs}}^{\mathrm{rlrs}}$ may output an incorrect automaton as shown in the following example:

Example 56 Let the TRS R, the replacement map μ, and the TA $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ as follows:

$$\begin{array}{l}
R = \{a \to b, a \to d, c \to d, f(x,x) \to g(x)\},\\
\mu(f) = \mu(g) = \{1\},\\
Q = \{q_1, q_2, q^f\},\\
Q^f = \{q^f\},\\
\Delta = \{a \to q_1, b \to q_2, c \to q_2, f(q_1, q_2) \to q^f\},
\end{array}$$

and hence $\mathcal{L}(\mathcal{A}) = \{f(a,b), f(a,c)\}$. Thus, $\xhookrightarrow[R]{*}{}^{\mu}[\mathcal{L}(\mathcal{A})]$ is the set $\{f(a,b), f(b,b), f(d,b), f(a,c), f(b,c), f(d,c), g(b)\}$.

$\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ output the automaton $\mathcal{A}_*$ whose transition rules in Δ_* are as

$$\Delta_* = \left\{ \begin{array}{rclrcl}
a & \to & \langle\{q_1\}, \mathtt{x}\rangle, & b & \to & \langle\{q_2\}, \mathtt{i}\rangle,\\
b & \to & \langle P, \mathtt{a}\rangle, & c & \to & \langle\{q_2\}, \mathtt{x}\rangle,\\
d & \to & \langle P, \mathtt{a}\rangle, & f(\langle\{q_1\}, \mathtt{x}\rangle, \langle\{q_2\}, \mathtt{i}\rangle) & \to & \langle\{q^f\}, \mathtt{x}\rangle,\\
g(\langle\{q_2\}, \mathtt{a}\rangle) & \to & \langle\{q^f\}, \mathtt{a}\rangle & & &
\end{array} \right\}$$

where $P \in \{\{q_1\}, \{q_2\}, \{q_1, q_2\}\}$ and $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$. Hence, $\mathcal{A}_*$ accepts the terms $g(d)$ that is not in $\xhookrightarrow[R]{*}{}^{\mu}[\mathcal{L}(\mathcal{A})]$. □

As for Example 56, preparing another state that accepts only b to construct a correct automaton is enough. However, it is difficult to guarantee the termination of a procedure if a new state is added in the procedure.

In the following, we show the correctness of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$.

At first, we show several propositions that are trivially derived from the definition of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$. Since the proofs of Proposition 59 and 60 are almost same as linear and right-shallow case, we abbreviate these proofs.

Proposition 57 *Let $t \in \mathcal{T}(F)$. For $q^t \in Q$, $t \xrightarrow[\Delta_0]{*} \langle\{q^t\}, \mathtt{i}\rangle$ iff $t \xrightarrow[\Delta]{*} q^t$.*

Proof: Direct consequence of the construction of Δ, and Δ_0. □

Proposition 58 *Let $t \in \mathcal{T}(F)$. For any k, if $t \xrightarrow[\Delta_k]{*} \langle P, \mathtt{i}\rangle \in Q_*$, then $t \xrightarrow[\Delta_0]{*} \langle P, \mathtt{i}\rangle$. Moreover, P is of the form $\{q\}$.*

Proof: The first claim is proved similarly to the proof of Proposition 9. The second claim follows from the construction of states. □

Proposition 59 *Let $t \in \mathcal{T}(F)$. Then, $t \xrightarrow[\Delta_0]{*} \langle P, \mathtt{i} \rangle \in Q_*$ iff $t \xrightarrow[\Delta_0]{*} \langle P, \mathtt{a} \rangle \in Q_*$.*

Proof: This proposition is proved similarly to the proof of Proposition 10. □

Proposition 60 *Let $t \in \mathcal{T}(F)$. For any k, If $t \xrightarrow[\Delta_k]{*} \langle P, \mathtt{i} \rangle$, then $t \xrightarrow[\Delta_k]{*} \langle P, \mathtt{a} \rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 11. □

Next we show several technical lemmas. Lemmas 61, 62, 65, and 67 below are necessary to prove Lemmas 68 and 72, which are key lemmas to prove the completeness and the soundness. Lemmas 64 and 66 are auxiliary lemmas for Lemmas 65 and 67, respectively. We abbreviate the proof of Lemma 62 since the proof is almost similar to the proof for linear and right-shallow case.

Lemma 61 *Let $s, t \in \mathcal{T}(F)$, $s \xrightarrow[\Delta_0]{*} \langle P, \mathtt{x} \rangle$, and $t \xrightarrow[\Delta_0]{*} \langle P', \mathtt{x}' \rangle$. Then, $P = P'$ iff $s = t$.*

Proof: First we have s $\xrightarrow[\Delta_0]{*} \langle P, \mathtt{i} \rangle$, $t \xrightarrow[\Delta_0]{*} \langle P', \mathtt{i} \rangle$, $P = \{q\}$, and $P' = \{q'\}$ for some $q, q' \in Q$ from Proposition 58 and Proposition 59. Then, we have $s \xrightarrow[\Delta]{*} q^s = q$ and $t \xrightarrow[\Delta]{*} q^t = q'$ from Proposition 57 and the construction of $\mathcal{A}$. Thus, we have $P = P'$ iff $s = t$. □

Lemma 62 *If $\alpha : t[t']_p \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a} \rangle$ and $p \in \mathrm{Pos}^{\mu}(t)$, then there exists $\langle P', \mathtt{a} \rangle$ such that $t' \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a} \rangle$ and $t[\langle P', \mathtt{a} \rangle]_p \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a} \rangle$.*

Proof: This lemma is proved similarly to the proof of Lemma 12. □

Lemma 63 *Let R be a right-linear and right-shallow TRS and $l \to g(r_1, \dots, r_m) \in R$. For $r_j \notin X$ and any substitution σ, $r_j\sigma \xrightarrow[\Delta_*]{*} \langle \{q^{r_j}\}, \mathtt{i} \rangle$ and $r_j\sigma \xrightarrow[\Delta_*]{*} \langle \{q^{r_j}\}, \mathtt{a} \rangle$.*

Proof: By Proposition 57, 60, and the construction of Δ_0, this lemma is proved similarly to the proof of Lemma 13. □

Lemma 64 *If* $\langle P_1', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P_1, \mathtt{a}\rangle$ *and* $\langle P_2', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P_2, \mathtt{a}\rangle$*, then we have* $\langle P_1' \cup P_2', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P_1 \cup P_2, \mathtt{a}\rangle$*.*

Proof: We can assume $\langle P_1', \mathtt{a}\rangle \xrightarrow[\Delta_*]{n} \langle P_1, \mathtt{a}\rangle$ and $\langle P_2', \mathtt{a}\rangle \xrightarrow[\Delta_*]{n} \langle P_2'', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P_2, \mathtt{a}\rangle$ without loss of generality.

At first, we prove the claim that $\langle P_1' \cup P_2', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P_1 \cup P_2'', \mathtt{a}\rangle$. If $n = 0$, the claim trivially holds. If $n = 1$, the claim holds from (2) of Step 3 of $\mathrm{P_{cs}}$. If $n > 1$, the claim holds by repeating the process for $n = 1$.

Moreover, we can prove the claim that $\langle P_1 \cup P_2'', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P_1 \cup P_2, \mathtt{a}\rangle$ similarly to the previous claim. □

Lemma 65 *If* $t \xrightarrow[\Delta_*]{*} \langle P^j, \mathtt{a}\rangle$ *for* $1 \leq j \leq m$*, then we have* $t \xrightarrow[\Delta_*]{*} \langle \bigcup_{1\leq j\leq m} P^j, \mathtt{a}\rangle$*.*

Proof: The proof for $m = 1$ is trivial. We show the proof for $m = 2$ by induction on $|t|$. By applying the proof for $m = 2$ repeatedly, we can prove this lemma.

Let $t = f(t_1, \ldots, t_n)$. Then, each transition sequence is represented by $f(t_1, \ldots, t_n) \xrightarrow[\Delta_*]{*} f(\langle P_1^j, \mathtt{x}_1^j\rangle, \ldots, \langle P_n^j, \mathtt{x}_n^j\rangle) \xrightarrow[\Delta_*]{} \langle P_r^j, \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P^j, \mathtt{a}\rangle$ for $j \in \{1, 2\}$. From Lemma 64, we have $\langle P_r^1 \cup P_r^2, \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P^1 \cup P^2, \mathtt{a}\rangle$. Therefore, we show that $f(t_1, \ldots, t_n) \xrightarrow[\Delta_*]{*} \langle P_r^1 \cup P_r^2, \mathtt{a}\rangle$.

From (1) of Step 3 of $\mathrm{P_{cs}}$, we have the transition rule $f(\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle) \to \langle P_r^1 \cup P_r^2, \mathtt{a}\rangle \in \Delta_*$ where

- $P_i = \begin{cases} P_i^j & \cdots \text{ if } \mathtt{x}_i^j = \mathtt{i} \text{ for some } j \in \{1, 2\}, \text{ and} \\ P_i^1 \cup P_i^2 & \cdots \text{ if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a}. \end{cases}$

- $\mathtt{x}_i = \begin{cases} \mathtt{a} \cdots \text{if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a}, \text{ and} \\ \mathtt{i} \cdots \text{otherwise.} \end{cases}$

Here we show that $t_i \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{x}_i\rangle$ for $1 \leq i \leq n$.

- For i such that $\mathtt{x}_i = \mathtt{i}$, P_i is P_i^1 or P_i^2 and hence we have $t_i \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{x}_i\rangle$.

- For i such that $\mathtt{x}_i = \mathtt{a}$, P_i is $P_i^1 \cup P_i^2$ and hence we have $t_i \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{x}_i\rangle$ from the induction hypothesis.

Thus, we have the transition $f(t_1, \ldots, t_n) \xrightarrow[\Delta_*]{*} f(\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle) \xrightarrow[\Delta_*]{} \langle P_r, \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$. □

Lemma 66 *If $\langle P_1, \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$, then there exists $P_1' \subseteq P_1$ such that $\langle P_1', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}\rangle$ for all $P' \subseteq P$ where $P' \neq \emptyset$.*

Proof: By induction on $|P| + |P_1|$, we show the proof for the case of $\langle P_1, \mathtt{a}\rangle \xrightarrow[\Delta_*]{} \langle P, \mathtt{a}\rangle$. If $\langle P_1, \mathtt{a}\rangle = \langle P, \mathtt{a}\rangle$, then this lemma holds trivially. If $|\langle P_1, \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle| > 1$, then we can prove this lemma by applying the proof for $\langle P_1, \mathtt{a}\rangle \xrightarrow[\Delta_*]{} \langle P, \mathtt{a}\rangle$ repeatedly.

Let $P' = P \backslash P''$. We show that there exists P_1' such that $\langle P_1', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}\rangle$ where $P_1' \subseteq P_1$. If $|P| = 1$, then the claim holds trivially. If $|P| > 1$, we can assume that the transition rule $\langle P_1, \mathtt{a}\rangle \to \langle P, \mathtt{a}\rangle \in \Delta_*$ is produced by the rules $\langle P_1^j, \mathtt{a}\rangle \to \langle P^j, \mathtt{a}\rangle \in \Delta_*$ where $j \in \{1, 2\}$, $P^1 \cup P^2 = P$, and $P_1^1 \cup P_1^2 = P_1$ by (2) of Step 3 of $\mathrm{P_{cs}}$. Note that we have $|P^j| + |P_1^j| < |P| + |P_1|$ for $j \in \{1, 2\}$ because if $|P^j| + |P_1^j| = |P| + |P_1|$ then we have $P^1 = P^2$ and $P_1^1 = P_1^2$, and hence the rule $\langle P_1, \mathtt{a}\rangle \to \langle P, \mathtt{a}\rangle \in \Delta_*$ is the same as $\langle P_1^j, \mathtt{a}\rangle \to \langle P^j, \mathtt{a}\rangle \in \Delta_*$ for $j \in \{1, 2\}$.

For each j, we also have the transition rule $\langle P_1'^j, \mathtt{a}\rangle \to \langle P^j \backslash P'', \mathtt{a}\rangle \in \Delta_*$ for some $P_1'^j \subseteq P_1^j$ from the induction hypothesis.

Thus, we obtain $\langle P_1'^1 \cup P_1'^2, \mathtt{a}\rangle \xrightarrow[\Delta_*]{} \langle (P^1 \cup P^2) \backslash P'', \mathtt{a}\rangle = \langle P', \mathtt{a}\rangle$ where $P_1'^1 \cup P_1'^2 \subseteq P_1^1 \cup P_1^2 = P_1$ by (2) of Step 3 of $\mathrm{P_{cs}}$. □

Lemma 67 *If $t \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$, then $t \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}\rangle$ for any $P' \subseteq P$.*

Proof: Assume that the transition $t \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$ is represented as $t = f(t_1, \ldots, t_n) \xrightarrow[\Delta_*]{*} f(\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle) \xrightarrow[\Delta_*]{} \langle P_{\mathtt{r}}, \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$. From Lemma 66, there exists $P_{\mathtt{r}}' \subseteq P_{\mathtt{r}}$ such that $\langle P_{\mathtt{r}}', \mathtt{a}\rangle \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}\rangle$ for all $P' \subseteq P$. Therefore, we prove the claim that we have $t = f(t_1, \ldots, t_n) \xrightarrow[\Delta_*]{*} f(\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle) \xrightarrow[\Delta_*]{} \langle P_{\mathtt{r}}', \mathtt{a}\rangle$. Let $P_{\mathtt{r}}' = P_{\mathtt{r}} \backslash P_{\mathtt{r}}''$. We prove the claim by simultaneous induction on $\Sigma_{i=1}^n |P_i| + |P_{\mathtt{r}}|$ and $|t|$. If $|P_{\mathtt{r}}| = 1$, then the claim holds trivially. Since the proof in the case of $|t| = 1$ is included in the case of $|P_{\mathtt{r}}| > 1$, we omit it. If $|P_{\mathtt{r}}| > 1$, the transition rule $f(\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle) \to \langle P_{\mathtt{r}}, \mathtt{a}\rangle$ is produced from the transition rules $f(\langle P_1^j, \mathtt{x}_1\rangle, \ldots, \langle P_n^j, \mathtt{x}_n\rangle) \to \langle P_{\mathtt{r}}^j, \mathtt{a}\rangle$ where $j \in \{1, 2\}$ by (1) of Step 3 of $\mathrm{P_{cs}}$ and $P_{\mathtt{r}}$, P_i's, and $\mathtt{x}_i$'s are represented as follows:

- $P_{\mathtt{r}} = P_{\mathtt{r}}^1 \cup P_{\mathtt{r}}^2$,
- $P_i = \begin{cases} P_i^j & \cdots \text{ if } \mathtt{x}_i^j = \mathtt{i} \text{ for some } j \in \{1,2\}, \text{ and} \\ P_i^1 \cup P_i^2 & \cdots \text{ if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a}. \end{cases}$
- $\mathtt{x}_i = \begin{cases} \mathtt{a} \cdots \text{if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a}, \text{ and} \\ \mathtt{i} \cdots \text{otherwise.} \end{cases}$

Here, we show that $t_i \xrightarrow[\Delta_*]{*} \langle P_i^j, \mathtt{x}_i^j\rangle$ for $j \in \{1, 2\}$ and $1 \leq i \leq n$.

- For i such that $\mathtt{x}_i = \mathtt{a}$, we have $\mathtt{x}_i^j = \mathtt{a}$ and $P_i^j \subseteq P_i$. Thus, we have $t_i \xrightarrow[\Delta_*]{*} \langle P_i^j, \mathtt{x}_i^j\rangle$ from the induction hypothesis.
- For i such that $\mathtt{x}_i = \mathtt{i}$, we have $\mathcal{L}(\Delta_*, \langle P_i^1, \mathtt{x}_i^1\rangle) \cap \mathcal{L}(\Delta_*, \langle P_i^2, \mathtt{x}_i^2\rangle) \neq \emptyset$, and, $\mathtt{x}_i^1 = \mathtt{i}$ or $\mathtt{x}_i^2 = \mathtt{i}$. From Lemma 61, t_i is the only term accepted by $\langle P_i^j, \mathtt{i}\rangle$ where j is 1 or 2, and from $\mathcal{L}(\Delta_*, \langle P_i^1, \mathtt{x}_i^1\rangle) \cap \mathcal{L}(\Delta_*, \langle P_i^2, \mathtt{x}_i^2\rangle) \neq \emptyset$, we have $t_i \xrightarrow[\Delta_*]{*} \langle P_i^j, \mathtt{x}_i\rangle$ for both $j = 1$ and $j = 2$.

Thus, we have $f(t_1, \ldots, t_n) \xrightarrow[\Delta_*]{*} f(\langle P_1^j, \mathtt{x}_1^j\rangle, \ldots, \langle P_n^j, \mathtt{x}_n^j\rangle) \xrightarrow[\Delta_*]{} \langle P_{\mathtt{r}}^j, \mathtt{a}\rangle$ for both $j = 1$ and $j = 2$.

Moreover, we have $\Sigma_{i=1}^n |P_i^j| + |P_{\mathtt{r}}^j| < \Sigma_{i=1}^n |P_i| + |P_{\mathtt{r}}|$ for both $j = 1$ and $j = 2$ because if it does not hold, then the rule for $j = 1$ or $j = 2$ become the same one as the rule $f(\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle) \to \langle P, \mathtt{a}\rangle$. Hence, we have $t \xrightarrow[\Delta_*]{*} \langle P_{\mathtt{r}}^j \backslash P_{\mathtt{r}}''\rangle$ for both $j = 1$ and $j = 2$ from the induction hypothesis.

Thus, we have $t \xrightarrow[\Delta_*]{*} \langle P_{\mathtt{r}}^1 \cup P_{\mathtt{r}}^2 \backslash P_{\mathtt{r}}'', \mathtt{a}\rangle$ from Lemma 65. □

The following lemma is a key lemma for the completeness of $\mathrm{P_{cs}^{rlrs}}$.

Lemma 68 *If R is a right-linear and right-shallow TRS, then $s \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$ and $s \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}\, t$ implies $t \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$.*

Proof: Similar to the proof of Lemma 14, we present the proof in the case where $s \underset{R}{\hookrightarrow}{}^{\mu}\, t$ and let $s \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$ and $s = s[l\sigma]_p \underset{R}{\hookrightarrow}{}^{\mu}\, s[r\sigma]_p = t$ for some rewrite rule $l \to r \in R$, where $p \in \mathrm{Pos}^{\mu}(s)$. Then we have the transition sequence $s[l\sigma]_p \xrightarrow[\Delta_*]{*} s[\langle P', \mathtt{a}\rangle]_p \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$ for some $\langle P', \mathtt{a}\rangle \in Q_*$ by Lemma 62.

From Lemma 67, we have $l\sigma \xrightarrow[\Delta_*]{*} \langle\{q\}, \mathtt{a}\rangle$ for all $q \in P$. Therefore, we prove that $r\sigma \xrightarrow[\Delta_*]{*} \langle\{q\}, \mathtt{a}\rangle$ for any $q \in P$, because if we can prove this claim, we have $s[r\sigma]_p \xrightarrow[\Delta_*]{*} s[\langle P', \mathtt{a}\rangle]_p \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$ from Lemma 65.

We show the proof in the case where the rewrite rule $l \to r$ is of the form $C[x_1, \ldots, x_n] \to g(r_1, \ldots, r_m)$ where C has no variable since the proof in the other case is similar.

Here, $C[x_1, \ldots, x_n]\sigma \xrightarrow[\Delta_*]{*} \langle\{q\}, \mathtt{a}\rangle$ is represented as $C[x_1, \ldots, x_n]\sigma \xrightarrow[\Delta_*]{*} C[\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle] \xrightarrow[\Delta_*]{} \langle\{q\}, \mathtt{a}\rangle$ for some $\langle P_i, \mathtt{x}_i\rangle \in Q_*$ for $1 \le i \le n$. Since we have $C[x_1, \ldots, x_n] \to g(r_1, \ldots, r_m) \in R$, $C[\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle] \xrightarrow[\Delta_*]{*} \langle\{q\}, \mathtt{a}\rangle$, and $\sigma : X \to \mathcal{T}(F)$ such that $x_i\sigma \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{x}_i\rangle$ for all $1 \le i \le n$, Δ_* has the transition rule $g(\langle P', \mathtt{x}'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m\rangle) \to \langle\{q\}, \mathtt{a}\rangle \in \Delta_*$ such that

- $P'_j = \begin{cases} \{q^{r_j}\} & \cdots \text{ if } r_j \notin X, \\ P_i & \cdots \text{ if } r_j \in X \wedge \exists i \in I_j.\mathtt{x}_i = \mathtt{i}, \text{ and} \\ \bigcup_{i \in I_j} P_i & \cdots \text{ if } r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a}. \end{cases}$

- $\mathtt{x}'_j = \begin{cases} \mathtt{i} \cdots \text{ if } j \notin \mu(g) \wedge (r_j \in X \Rightarrow \exists i \in I_j.\mathtt{x}_i = \mathtt{i}), \text{ and} \\ \mathtt{a} \cdots \text{ if } j \in \mu(g) \vee (r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a}). \end{cases}$

where $I_j = \{i \mid x_i = r_j\}$.

Here, we show that $r_j\sigma \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle$ for $1 \le j \le m$.

1. For j such that $r_j \notin X$, we have $r_j\sigma \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle$ from Lemma 63.

2. For j such that $r_j \in X$ and there exists $i \in I_j$ such that $\mathtt{x}_i = \mathtt{i}$, we have $r_j\sigma = x_i\sigma$ and hence $r_j\sigma \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{i}\rangle$. If $j \notin \mu(g)$, we have $\mathtt{x}'_i = \mathtt{i}$ and hence $r_j\sigma \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_i\rangle = \langle P_i, \mathtt{i}\rangle$. If $j \in \mu(g)$, we have $\mathtt{x}'_j = \mathtt{a}$ and hence $r_j\sigma \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle = \langle P_i, \mathtt{a}\rangle$ from Proposition 60.

3. For j such that $r_j \in X$ and there exists no $i \in I_j$ such that $\mathtt{x}_i = \mathtt{i}$, since we have $r_j\sigma = x_{i'}\sigma \xrightarrow[\Delta_k]{*} \langle P_{i'}, \mathtt{x}_{i'}\rangle = \langle P_{i'}, \mathtt{a}\rangle$ for all $i' \in I_j$, $r_j\sigma \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle = \langle \bigcup_{i' \in I_j} P_{i'}, \mathtt{a}\rangle$ follows from Lemma 65.

Therefore we have $g(r_1, \ldots, r_m)\sigma \xrightarrow[\Delta_*]{*} g(\langle P'_1, \mathtt{x}'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m\rangle) \xrightarrow[\Delta_*]{} \langle\{q\}, \mathtt{a}\rangle$.

By applying the above statement for all $q \in P$, this lemma holds. $\square$

The following lemma shows the completeness of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$.

Lemma 69 *If R is a right-linear and right-shallow TRS, then* $\mathcal{L}(\mathcal{A}_*) \supseteq \underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A})]$.

Proof: Let $s \overset{*}{\underset{R}{\hookrightarrow}}^{\mu} t$ and $s \xrightarrow[\Delta]{*} q \in Q^f$. Since we have $s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{i}\rangle$ from Proposition 57, we also have $s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{a}\rangle$ from Proposition 59. Hence $t \xrightarrow[\Delta_*]{*} \langle\{q\}, \mathtt{a}\rangle \in Q^f_*$ follows by Lemma 68. $\square$

In the previous chapter, we use the order of transition $\sqsupset$ to prove the soundness of algorithms, but we cannot prove the soundness of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ by induction on $\sqsupset$. Therefore, we define the following measure of transitions and the order.

Definition 70 Let $[[s \xrightarrow[\Delta_*]{*} p]]$ be the sequence of integers defined as follows:

$$[[s \xrightarrow[\Delta_*]{*} p]] = \begin{cases} k.[[s \xrightarrow[\Delta_*]{*} p']] & \cdots \text{ if } t \xrightarrow[\Delta_*]{*} p', \text{ and } \xrightarrow[\Delta_k \setminus \Delta_{k-1}]{} p, \\ & \quad \text{if } t = f(t_1, \ldots, t_n) \xrightarrow[\Delta_*]{*} f(p_i, \ldots, p_n) \\ k.[[t_i \xrightarrow[\Delta_*]{*} p_i]] & \cdots \quad \xrightarrow[\Delta_k \setminus \Delta_{k-1}]{} p \text{ and} \\ & \quad \forall j \neq i.[[t_i \xrightarrow[\Delta_*]{*} p_i]] \geq_{\mathrm{lex}} [[t_j \xrightarrow[\Delta_*]{*} p_j]]. \end{cases}$$

where $p, p', p_i, p_j \in Q_*$ and Δ_k for $k < 0$ is assumed as empty set.

Definition 71 Let $\succeq$ and $\succ$ be the order for transition sequences as follows:

$$\alpha \succeq \beta \text{ iff } \begin{cases} (1)\ \beta \text{ occurs in } \alpha, \text{or} \\ (2)\ \alpha \text{ does not occur in } \beta \text{ and } [[\alpha]] \geq_{\mathrm{lex}} [[\beta]]. \end{cases}$$

$$\alpha \succ \beta \text{ iff } \begin{cases} (1)\ \beta \text{ occurs in } \alpha \text{ and } \alpha \text{ does not occur in } \beta, \text{or} \\ (2)\ \alpha \text{ does not occur in } \beta \text{ and } [[\alpha]] >_{\mathrm{lex}} [[\beta]]. \end{cases}$$

Note that $\succ$ is well-founded, $\alpha \succ \beta$ implies $\alpha \succeq \beta$, and if $\alpha \succeq \beta$ then $\beta \not\succ \alpha$. The minimal components in the order $\succ$ are the transitions of the form $a \xrightarrow[\Delta_0]{} \langle P, \mathtt{x}\rangle$ where a is a constant.

The following lemma is the key lemma for the soundness of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$.

Lemma 72 *If R is a right-linear and right-shallow TRS, then $\alpha : t \xrightarrow[\Delta_*]{*} \langle P, \mathtt{a}\rangle$ implies that there exists s and $q \in P$ such that $s \xleftarrow[R]{*}{}^{\mu}\, t$ and $s \xrightarrow[\Delta_0]{*} \langle \{q\}, \mathtt{a}\rangle$.*

Proof: We show this lemma by induction on α with respect to $\succ$. Since we have $t \xrightarrow[\Delta_*]{*} \langle \{q\}, \mathtt{a}\rangle$ for all $q \in P$ from Lemma 67, we show the proof in the case where P is of the form $\{q\}$.

Similarly to the proof of Lemma 16, We show the detailed proof in the case of t is of the form $g(t_1, \ldots, t_m)$ and hence α is represented as $t = g(t_1, \ldots, t_m) \xrightarrow[\Delta_*]{*} g(\langle P'_1, \mathtt{x}'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m\rangle) \xrightarrow[\Delta_k]{} \langle \{q\}, \mathtt{a}\rangle$.

If $k = 0$, then each P'_j for $1 \leq j \leq m$ is represented by $\{q'_j\}$ and this lemma is proved similarly to the proof of Lemma 16 from Proposition 58. If the last transition rule applied at α is in $\Delta_k \backslash \Delta_{k-1}$ where $k > 0$, the transition rule $g(\langle P'_1, \mathtt{x}'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m\rangle) \to \langle \{q\}, \mathtt{a}\rangle \in \Delta_k \backslash \Delta_{k-1}$ is produced by the first inference rule of Step 2. Therefore, we have $C[x_1, \ldots, x_n] \to g(r_1, \ldots, r_m) \in R$ where C has no variable and $C[\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle] \xrightarrow[\Delta_{k-1}]{*} \langle \{q\}, \mathtt{a}\rangle$, and $\sigma' : X \to \mathcal{T}(F)$ such that $x_i\sigma' \xrightarrow[\Delta_{k-1}]{*} \langle P_i, \mathtt{x}_i\rangle$ for all $1 \leq i \leq n$, and each $\langle P'_j, \mathtt{x}'_j\rangle$ is represented as follows:

- $$P'_j = \begin{cases} \{q^{r_j}\} & \cdots \text{ if } r_j \notin X, \\ P_i & \cdots \text{ if } r_j \in X \wedge \exists i \in I_j.\mathtt{x}_i = \mathtt{i}, \text{ and} \\ \bigcup_{i \in I_j} P_i & \cdots \text{ if } r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a}. \end{cases}$$

- $$\mathtt{x}'_j = \begin{cases} \mathtt{i} \cdots \text{ if } j \notin \mu(g) \wedge (r_j \in X \Rightarrow \exists i \in I_j.\mathtt{x}_i = \mathtt{i}), \text{ and} \\ \mathtt{a} \cdots \text{ if } j \in \mu(g) \vee (r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a}). \end{cases}$$

where $I_j = \{i \mid x_i = r_j\}$. In the following, we show that there exists the substitution σ such that $g(r_1, \ldots, r_m)\sigma \xleftarrow[R]{*}{}^{\mu}\, g(t_1, \ldots, t_m)$ and $\alpha' : g(r_1, \ldots, r_m)\sigma \xrightarrow[\Delta_*]{*} g(\langle P'_1, \mathtt{x}'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m\rangle) \xrightarrow[\Delta_k \backslash \Delta_{k-1}]{} \langle \{q\}, \mathtt{a}\rangle$ where $\alpha \succeq \alpha'$.

1. For j such that $r_j \notin X \wedge j \notin \mu(g)$, we have $P'_j = \{q^{r_j}\}$ and $\mathtt{x}'_j = \mathtt{i}$. Since $t_j \xrightarrow[\Delta_0]{*} \langle \{q^{r_j}\}, \mathtt{i}\rangle$ by Proposition 59, we have $t_j = r_j$ from Proposition 57 and the construction of $\mathcal{A}$.

2. For j such that $r_j \notin X \wedge j \in \mu(g)$, we have $P'_j = \{q^{r_j}\}$ and $\mathtt{x}'_j = \mathtt{a}$. Hence, we have $s_j \xleftarrow[R]{*}{}^{\mu}\, t_j$ and $s_j \xrightarrow[\Delta_0]{*} \langle P'_j, \mathtt{x}'_j\rangle = \langle \{q^{r_j}\}, \mathtt{a}\rangle$ from the induction

hypothesis. Since we have $s_j \xrightarrow[\Delta_0]{*} \langle \{q^{r_j}\}, \mathtt{i}\rangle$ from Proposition 59, we have $s_j = r_j$ from Proposition 57 and the construction of Δ.

3. For j such that $r_j \in X \wedge j \notin \mu(g) \wedge \exists i \in I_j.\mathtt{x}_i = \mathtt{i}$, we have $t_j \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle = \langle P_i, \mathtt{i}\rangle$. Hence, we have $t_j \xrightarrow[\Delta_0]{*} \langle P'_j, \mathtt{x}'_j\rangle$ from Proposition 59, and let $r_j\sigma = t_j$.

4. For j such that $r_j \in X \wedge j \in \mu(g) \wedge \exists i \in I_j.\mathtt{x}_i = \mathtt{i}$, we have $t_j \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle = \langle P_i, \mathtt{a}\rangle$. Since we have $\langle P_i, \mathtt{i}\rangle = \langle P_i, \mathtt{x}_i\rangle$ where P_i is of the form $\{q_i\}$ from Proposition 58, there exists a s_j such that $s_j \xleftrightarrow[R]{*}{}^{\mu} t_j$ and $s_j \xrightarrow[\Delta_0]{*} \langle P'_j, \mathtt{x}'_j\rangle$ from the induction hypothesis. Let s_j be $r_j\sigma$.

5. For j such that $r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a}$, we take $r_j\sigma = t_j$.

Note that σ is well defined from the right-linearity of R and we have $\alpha' \preceq \alpha$ because α does not occur in α', and we have $(r_j\sigma \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle) \preceq (t_j \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j\rangle)$ for every $1 \leq j \leq m$.

Next, we define a substitution $\sigma'' : \mathrm{Var}(f(l_1, \ldots, l_n)) \to \mathcal{T}(F)$ as follows:

$$x\sigma'' = \begin{cases} x\sigma \cdots & \text{if there exists } r_j \text{ such that } r_j = x \\ x\sigma' \cdots & \text{otherwise.} \end{cases}$$

Here, we show that we have $x_i\sigma'' \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{x}_i\rangle$ for all $1 \leq i \leq n$.

1. For i such that there exists j such that $x_i = r_j$ and $i' \in I_j$ such that $\mathtt{x}_{i'} = \mathtt{i}$, we have $x_{i'}\sigma' \xrightarrow[\Delta_0]{*} \langle P_{i'}, \mathtt{i}\rangle = \langle P'_j, \mathtt{i}\rangle$ from Proposition 57. Thus we have $x_i\sigma' = x_i\sigma''$ from Lemma 61 and hence $x_i\sigma'' \xrightarrow[\Delta_{k-1}]{*} \langle P_i, \mathtt{x}_i\rangle$.

2. For i such that there exists j such that $x_i = r_j$ and no $i' \in I_j$ such that $\mathtt{x}_{i'} = \mathtt{i}$, we have $P_i \subseteq P'_j$ and hence $x_i\sigma'' \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{x}_i\rangle$ from Lemma 65.

3. For i such that there exists no j such that $x_i = r_j$, we have $x_i\sigma'' = x_i\sigma' \xrightarrow[\Delta_{k-1}]{*} \langle P_i, \mathtt{x}_i\rangle$ from the construction of the rule.

Thus, we have $\beta : C[x_1, \ldots, x_n]\sigma'' \xrightarrow[\Delta_*]{*} C[\langle P_1, \mathtt{x}_1\rangle, \ldots, \langle P_n, \mathtt{x}_n\rangle] \xrightarrow[\Delta_{k-1}]{*} \langle \{q\}, \mathtt{a}\rangle$.

Note that $\alpha \succeq \alpha' \succ \beta$ because transition rules in $\Delta_k \backslash \Delta_{k-1}$ are not applied at β except for the transitions $x_i\sigma'' \xrightarrow[\Delta_*]{*} \langle P_i, \mathtt{x}_i\rangle$ where there exists j such that

$x_i = r_j$. However, since $x_i\sigma''$ is a proper subterm of $g(r_1, \ldots, r_m)\sigma''$, α' does not occur in β.

Finally, we obtain the term s such that $s \overset{*}{\underset{R}{\hookrightarrow}}^{\mu} C[x_1, \ldots, x_n]\sigma'' \overset{*}{\underset{R}{\hookrightarrow}}^{\mu} t$ and $s \xrightarrow[\Delta_0]{*} \langle \{q\}, \mathtt{a} \rangle$ from the induction hypothesis. □

The following lemma shows the soundness of $\mathrm{P}_{\mathrm{cs}}^{\mathrm{rlrs}}$.

Lemma 73 *If R be a right-linear and right-shallow TRS, then $\mathcal{L}(\mathcal{A}_*) \subseteq \underset{R}{\hookrightarrow}^{\mu}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let t be $t \xrightarrow[\Delta_*]{*} \langle P, \mathtt{x} \rangle \in Q_*^f$ where P contains the state $q^f \in Q^f$. Then, we have $t \xrightarrow[\Delta_*]{*} \langle \{q^f\}, \mathtt{x} \rangle$ from Lemma 67. If $\mathtt{x} = \mathtt{i}$, we have $t \xrightarrow[\Delta_0]{*} \langle \{q^f\}, \mathtt{i} \rangle \in Q_*^f$ from Proposition 58. If $\mathtt{x} = \mathtt{a}$, there exists the term s such that $s \overset{*}{\underset{R}{\hookrightarrow}}^{\mu} t$ and $s \xrightarrow[\Delta_0]{*} \langle \{q^f\}, \mathtt{a} \rangle$ from Lemma 72 and we have $s \xrightarrow[\Delta_0]{*} \langle \{q^f\}, \mathtt{i} \rangle$ from Proposition 59.

Thus, we have $s \xrightarrow[\Delta]{*} q^f$ from Proposition 57. □

The following theorem is proved by Lemma 69 and 73.

Theorem 74 *Reachability is decidable for right-linear and right-shallow TRSs with respect to the context-sensitive reduction.*

Chapter 5

Decidability of Reachability for Shallow and Non-Erasing TRSs

With respect to the ordinary reduction, it was shown that reachability is undecidable for shallow TRSs by [19] and [17]. Moreover, since ordinary reductions are special cases of context-sensitive reductions, reachability is also undecidable for shallow TRSs with respect to the context-sensitive reduction. However, with respect to the innermost reduction, it was shown that reachability is decidable for shallow TRSs by [13].

In this chapter, we show that reachability is decidable for shallow and non-erasing TRSs with respect to the context-sensitive innermost reduction.

5.1 On the Context-Sensitive Innermost Reduction

Similarly to the previous chapter, we show the algorithm $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ that constructs the TA recognizing the set of terms reachable by the context-sensitive innermost reduction of a shallow and non-erasing TRS from an input term. $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ is a modification of the algorithm $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$, and the idea of modification is similar to modification from $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ to $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$, i.e. we augment the parameter that shows whether the state accepts normal forms or not. However, generally, normal forms with respect to the context-sensitive reduction of left-non-linear TRS cannot be recognized by a TA. Therefore, first we show the construction of a deterministic complete reduced tree automata with constraints between brothers (TACBB) recognizing the set of normal forms with respect to the context-sensitive reduction, and then we show $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

5.1.1 Tree automata with constraints between brothers

Here, we follow the definition of tree automata with constraints between brothers at [5].

A *tree automaton with constraints between brothers* (TACBB) is a extended TA in which transition rules have *constraints between brothers*. Constraints between brothers are recursively defined: $\top$, $\bot$, equality $i = j$, and disequality $i \neq j$ are constraints between brothers where $i, j \in \mathbb{N}$, and if c_1 and c_2 are constraints between brothers, then a conjunction $c_1 \wedge c_2$ and a disjunction $c_1 \vee c_2$ are also constraints between brothers. A term $f(t_1, \ldots, t_n)$ *satisfies* the constraints between brothers c if c is true by assigning true to $\top$, equality $i = j$ if $t_i = t_j$, and disequality $i \neq j$ if $t_i \neq t_j$, and false to $\bot$, equality $i = j$ if $t_i \neq t_j$, and disequality $i \neq j$ if $t_i = t_j$. Each transition rule is of the form $f(q_1, \ldots, q_n) \xrightarrow{c} q$ or $q_1 \xrightarrow{\top} q$ where c is a constraint between brothers. A term $f(t_1, \ldots, t_n)$ can reach to a state q by the transition rule $f(q_1, \ldots, q_n) \xrightarrow{c} q$ Δ of a TACBB if $t_i \xrightarrow[\Delta]{*} q_i$ for $1 \leq i \leq n$ and $f(t_1, \ldots, t_n)$ satisfies c.

The following properties on TACBB are known.

Theorem 75 ([5]) *All of the following holds for TACBBs:*

1. *For a given TACBB $\mathcal{A}$, we can construct a deterministic, complete, and reduced TACBB $\mathcal{A}'$ that recognizes $\mathcal{L}(\mathcal{A})$.*

2. *The class of recognizable tree languages is closed under union, intersection, and complementation.*

3. *The membership problem and the emptiness problem are decidable.*

5.1.2 Tree Automata Accepting Normal Forms

In this section, we give an algorithm to construct a deterministic, complete, and reduced tree automata with constraints between brothers recognizing the set of normal forms with respect to the context-sensitive reduction of a shallow TRS R and a replacement map μ.

The algorithm is similar to the ones for the context-sensitive reduction of linear right-shallow TRSs. However, the construction of $\mathcal{A}_l$, which recognizes the set of terms having a redex $l\sigma$ at a μ-replacing position for each $l \to r \in R$ is different. Now we show the details of the construction of $\mathcal{A}_l$.

Each component of $\mathcal{A}_l$ is as follows.

- $Q_l = \{u^\circ, u_\perp\} \cup \{u_t \mid t \lhd l, t \notin X\}$.
- $Q_l^f = \{u^\circ\}$
- Δ_l consists of the following transition rules:

  (i) $f(u_\perp, \ldots, u_\perp) \xrightarrow{\top} u_\perp$ for each $f \in F$,

  (ii) $f(u_{t_1}, \ldots, u_{t_n}) \xrightarrow{\top} u_{f(t_1,\ldots,t_n)}$ for each $f \in F$ and state $u_{f(t_1,\ldots,t_n)}$.

  (iii) $f(u_{s_1}, \ldots, u_{s_n}) \xrightarrow{c} u^\circ$ where $f(s_1, \ldots, s_n)$ is the term obtained by replacing all variables in $l = f(l_1, \ldots, l_n)$ by $\perp$, and c is the conjunction of all equalities $i = j$ where $l_i = l_j \in X$.

  (iv) $f(u_1, \ldots, u_n) \to u^\circ$ for each $f \in F$ if exactly one u_j such that $j \in \mu(f)$ is u° and the other u_i's are $u_\perp$.

  Each state u_t is associated with a proper non-variable subterm t of l. From the shallowness of R, t in u_t has no variable.

We obtain the following lemmas for $\mathcal{A}_l$. Lemma 77 shows that $\mathcal{A}_l$ recognizes the set of terms having a redex $l\sigma$ at a μ-replacing position.

Lemma 76 *$\mathcal{L}(\mathcal{A}_l, u_t)$ is equal to the singleton set that consists of $t \lhd l$, (that is, $\mathcal{L}(\mathcal{A}_l, u_t) = \{t \in \mathcal{T}(F)\}$.)*

Proof:

($\supseteq$) By induction on the height $|t|$ of t, we prove the claim that $t \xrightarrow[\Delta_l]{*} u_t$ for the proper subterm t of l. Since l is shallow, t has no variable. Hence we can represent t as $f(t_1, \ldots, t_n)$ where $n \geq 0$. From the construction of (ii) of Δ_l, we have the transition rule $f(u_{t_1}, \ldots, u_{t_n}) \xrightarrow{\top} u_{f(t_1,\ldots,t_n)}$

From the induction hypothesis, we have $t_i \xrightarrow[\Delta_l]{*} u_{t_i}$ for all $1 \leq i \leq n$. Thus, we have $t\sigma = f(t_1, \ldots, t_n)\sigma \xrightarrow[\Delta_l]{*} f(u_{t_1}, \ldots, u_{t_n}) \xrightarrow[\Delta_l]{} u_t$.

($\subseteq$) We show that if $\alpha : \ t \xrightarrow[\Delta_l]{*} u_{f(t_1,\ldots,t_n)}$ then we have $t = f(t_1, \ldots, t_n)$ by induction on $|\alpha|$. From the construction of Δ_l, the last transition rule applied in α is represented as $f(u_{t_1}, \ldots, u_{t_n}) \xrightarrow{c} u_{f(t_1,\ldots,t_n)}$.

From the induction hypothesis, we have $t_i \xrightarrow[\Delta_l]{*} u_{t_i}$ for all $1 \leq i \leq n$.

Thus, we have $f(t_1, \ldots, t_n) = f(l_1, \ldots, l_n)\sigma$. $\square$

Lemma 77 $\mathcal{L}(\mathcal{A}_l) = \{t[s]_p \mid t \in \mathcal{T}(F),\ s$ *is a ground instance of* $l,\ p \in Pos^\mu(t)\}$.

Proof:

($\supseteq$) Let $l = f(l_1, \ldots, l_n)$. First, we show that $s = f(l_1, \ldots, l_n)\sigma \xrightarrow[\Delta_l]{*} u^\circ$. For $l_i \notin X$, l_i is ground from shallowness of l and $l_i \xrightarrow[\Delta_l]{*} u_{l_i}$. For $l_i \in X$, we have $l_i\sigma \xrightarrow[\Delta_l]{*} u_\perp$. From (iii) of construction of Δ_l, we have the transition rule $f(u_{s_1}, \ldots, u_{s_n}) \xrightarrow{c} u^\circ$ where $s_i = l_i$ for $i \notin X$ and $s_i = \perp$ for $i \in X$, and c is the conjunction of all equalities $i = j$ where $l_i = l_j \in X$. Since we have $l_i\sigma = l_j\sigma$ for $l_i = l_j \in X$, s satisfies c. Thus, we have $s \xrightarrow[\Delta_l]{*} u^\circ$ and from (i) and (iv) of the construction of Δ_l, we have $t[s]_p \xrightarrow[\Delta_l]{*} t[u^\circ]_p \xrightarrow[\Delta_l]{} u^\circ$.

($\subseteq$) Let $l = f(l_1, \ldots, l_n)$ and $t \xrightarrow[\Delta_l]{*} u^\circ$, then we have $t \xrightarrow[\Delta_l]{*} t[f(u_{s_1}, \ldots, u_{s_n})]_p \xrightarrow[\Delta_l]{} t[u^\circ]_p \xrightarrow[\Delta_l]{*} u^\circ$ where $s_i = l_i$ for $l_i \notin X$, $s_i = \perp$ for $l_i \in X$, and $p \in \mathrm{Pos}^\mu(t)$ from (iii) and (iv) of the construction of Δ_l. From Lemma 76, we have $t|_{pi} = l_i \notin X$. Moreover, since the transition $t[f(u_{s_1}, \ldots, u_{s_n})]_p \xrightarrow[\Delta_l]{} t[u^\circ]_p$ has the constraint c that is the conjunction of all equalities $i = j$ where $l_i = l_j \in X$, we have $t|_{pi} = t|_{pj}$ for $l_i = l_j \in X$. Hence $t|_p$ is a ground substitution of l. Thus we have $t = t[s]_p$ for some ground instance s of l.

$\square$

The method of determinization is so called "*subset construction*". The claim "$t \xrightarrow[\mathcal{A}^d]{*} S$ iff $S = \{q \mid t \xrightarrow[\mathcal{A}]{*} q\}$" holds where $\mathcal{A}^d$ is determinized from $\mathcal{A}$ by subset construction. Therefore, the following lemma holds.

Lemma 78 *Let s be a proper subterm of l, u_s and S'' be a state and a subset of the set of states of TACBB $\mathcal{A}_l$ respectively, and $\mathcal{A}_l^d$ be a determinized TACBB from $\mathcal{A}_l$ by subset construction. Then, $t \xrightarrow[\mathcal{A}_l^d]{*} \{u_s\} \cup S'$ iff $t = s$.*

Proof: From Lemma 76 and shallowness of l, $t \xrightarrow[\mathcal{A}_l]{*} q_s$ iff $t = s$.

Thus, this lemma holds from the above claim. □

As shown in Lemma 77, the TACBB $\mathcal{A}_l$ recognizes the set of terms having a redex $l\sigma$ at a μ-replacing position. Now we obtain the following lemma.

Lemma 79 *For a shallow TRS R and a replacement map μ, there exists a deterministic, complete, and reduced TACBB $\mathcal{A}_{\mathrm{NF}}$ that recognizes NF_R^μ.*

Proof: At first, we obtain a TACBB $\mathcal{A}_l$ for each $l \to r \in R$ by Lemma **??**, and we can determinize them. Let the determinized TACBB from A_l be $A_l^d = \langle Q_l^d, Q_l^{df}, \Delta_l^d \rangle$.

Here, we obtain a TACBB $\mathcal{A}' = \langle F, Q', Q'^f, \Delta' \rangle$ that recognizes the following set:

$$\bigcup_{l \to r \in R} \mathcal{L}(\mathcal{A}_l').$$

Let $R = \{l_i \to r_i \mid 1 \leq i \leq m\}$. The concrete construction of the TACBB $\mathcal{A}' = \langle Q', Q'^f, \Delta' \rangle$ is as follows:

- $Q' = \{\langle u_1, \ldots, u_n \rangle \mid u_i \in Q_{l_i}^d\}$,
- $Q'^f = \{\langle u_1, \ldots, u_n \rangle (\in Q') \mid \exists i. u_i \in Q_{l_i}^{df}\}$
- $f(\langle u_{11}, \ldots, u_{1m} \rangle, \ldots, \langle u_{n1}, \ldots, u_{nm} \rangle) \xrightarrow{c} \langle u_1, \ldots, u_m \rangle \in \Delta'$ where $f(u_{1i}, \ldots, u_{mi}) \xrightarrow{c_i} u_i \in \Delta_{l_i}^d$ and $c = c_1 \wedge \cdots \wedge c_m$.

This is the construction of the union of all $\mathcal{A}_l'$'s. Since this construction preserves determinacy of TACBB, the constructed TACBB $\mathcal{A}'$ is deterministic.

Converting $\mathcal{A}'$ to complete one is not so difficult. By adding the new state q_{e} and new transition rules such that $f(q_1, \ldots, q_n) \xrightarrow{c} q_{\mathsf{e}}$ where $q_1, \ldots, q_n \in Q'$

and $c = \top$ if $f(q_1, \ldots, q_n)$, which does not occur in any transition rule of Δ', otherwise c is equivalent to $\neg(c_1 \vee \cdots \vee c_k)$ where $f(q_1, \ldots, q_n) \xrightarrow{c_j} q \in \Delta'$ for some $q \in Q'$.

Since the emptiness problem of TACBB is decidable from Theorem 2, we can check whether each state is accessible or not and hence we can construct a reduced TACBB A'' by erasing the inaccessible state of $\mathcal{A}'$.

Finally, since $\mathcal{A}''$ is deterministic and complete, we can easily obtain the TA $\mathcal{A}'''$ that accepts complementation of $\mathcal{A}''$ by replacing the final state.

$\mathcal{L}(\mathcal{A}'')$ is the set of terms having redex at a μ-replacing position. Thus we can obtain the deterministic, complete, and reduced TACBB $\mathcal{A}_{\mathrm{NF}}$ recognizing NF_R^μ by the algorithm. □

Here, we show an example of $\mathcal{A}_{\mathrm{NF}}$

Example 80 Let the TRS R and the replacement map μ be as

$$R = \{a \to b, a \to c, f(x,b) \to g(x,a), g(x,x) \to h(x,x)\} \text{ and}$$
$$\mu(f) = \emptyset, \mu(g) = \{1,2\}, \mu(h) = \{1,2\}.$$

We construct the TACBB $\mathcal{A}_{\mathrm{NF}}$ such that $\mathcal{L}(\mathcal{A}_{\mathrm{NF}}) = \mathrm{NF}_R^\mu$ by the algorithm shown in this section.

First, we construct the deterministic TACBB $\mathcal{A}_a$, $\mathcal{A}_{f(x,a)}$, and $\mathcal{A}_{g(x,x)}$ at the first step of the algorithm.

The set of final states of $\mathcal{A}_a$ is $Q_a^f = \{U^\circ\}$ and the set of transition rules is:

$$\Delta_a = \left\{ \begin{array}{rclrclrcl} a & \xrightarrow{\top} & U^\circ, & b & \xrightarrow{\top} & U_\perp, & c & \xrightarrow{\top} & U_\perp, \\ f(U,U) & \xrightarrow{\top} & U_\perp, & g(U_\perp,U_\perp) & \xrightarrow{\top} & U_\perp, & g(U_1,U_2) & \xrightarrow{\top} & U^\circ, \\ h(U_\perp,U_\perp) & \xrightarrow{\top} & U_\perp, & h(U_1,U_2) & \xrightarrow{\top} & U^\circ & & & \end{array} \right\}$$

where $U, U_1, U_2 \in \{U_\perp, U^\circ\}$ and one of U_1 and U_2 is U°.

The set of final states of $\mathcal{A}_{f(x,b)}$ is $Q_{f(x,b)}^f = \{U^\circ\}$ and the set of transition rules is:

$$\Delta_{f(x,b)} = \left\{ \begin{array}{rclrclrcl} a & \xrightarrow{\top} & U_\perp, & b & \xrightarrow{\top} & U_b, & c & \xrightarrow{\top} & U_\perp, \\ f(U,U_b) & \xrightarrow{\top} & U^\circ, & f(U,U') & \xrightarrow{\top} & U_\perp, & g(U_1',U_2') & \xrightarrow{\top} & U_\perp, \\ g(U_1,U_2) & \xrightarrow{\top} & U^\circ, & g(U_\perp,U_\perp) & \xrightarrow{\top} & U_\perp, & h(U_1',U_2') & \xrightarrow{\top} & U_\perp, \\ h(U_1,U_2) & \xrightarrow{\top} & U^\circ & & & & & & \end{array} \right\}$$

where $U, U_1, U_2 \in \{U_\perp, U^\circ, U_b\}$, $U', U'_1, U'_2 \in \{U_\perp, U_b\}$, and one U_1 or U_2 is U°.

The set of final states of $\mathcal{A}_{g(x,x)}$ is $Q^f_{g(x,x)} = \{U^\circ\}$ and the set of transition rules is:

$$\Delta_{g(x,x)} = \left\{ \begin{array}{rclrcl} a & \xrightarrow{\top} & U_\perp, & b & \xrightarrow{\top} & U_b, \\ c & \xrightarrow{\top} & U_\perp, & f(U) & \xrightarrow{\top} & U_\perp, \\ g(U_\perp, U_\perp) & \xrightarrow{1=2} & U^\circ, & g(U_1, U_2) & \xrightarrow{\top} & U^\circ, \\ g(U_\perp, U_\perp) & \xrightarrow{\top} & U_\perp, & h(U_\perp, U_\perp) & \xrightarrow{\top} & U_\perp, \\ h(U_1, U_2) & \xrightarrow{\top} & U^\circ & & & \end{array} \right\}$$

where $U, U_1, U_2 \in \{U_\perp, U^\circ\}$ and one U_1 or U_2 is U°.

At the second step, we construct the TACBB $\mathcal{A}'$ accepting all unions of $\mathcal{A}_a$, $\mathcal{A}_{f(x,a)}$, and $\mathcal{A}_{g(x,x)}$.

The set of final states of $\mathcal{A}'$ is $Q'^f = \{\langle U_1, U_2, U_3\rangle\}$ where $U_1, U_2, U_3 \in \{U_\perp, U^\circ\}$ and one U_1, U_2, or U_3 is U°.

The set of transition rules of $\mathcal{A}'$ is:

$$\Delta' = \left\{ \begin{array}{rcl} a & \xrightarrow{\top} & \langle U^\circ, U_\perp, U_\perp\rangle, \\ b & \xrightarrow{\top} & \langle U_\perp, U_b, U_\perp\rangle, \\ c & \xrightarrow{\top} & \langle U_\perp, U_\perp, U_\perp\rangle, \\ f(U, U_b) & \xrightarrow{\top} & \langle U_\perp, U^\circ, U_\perp\rangle, \\ f(U, U') & \xrightarrow{\top} & \langle U_\perp, U_\perp, U_\perp\rangle, \\ g(\langle U_\perp, U'_1, U_\perp\rangle, \langle U_\perp, U'_2, U_\perp\rangle) & \xrightarrow{1=2} & \langle U_\perp, U_\perp, U_\perp\rangle, \\ g(\langle U_\perp, U'_1, U_\perp\rangle, \langle U_\perp, U'_2, U_\perp\rangle) & \xrightarrow{1\neq 2} & \langle U_\perp, U_\perp, U^\circ\rangle, \\ g(U_1, U_2) & \rightarrow & \langle U_\perp, U_\perp, U^\circ\rangle, \\ h(U_\perp, U_\perp) & \xrightarrow{\top} & U_\perp, \\ h(U_1, U_2) & \xrightarrow{\top} & U^\circ \end{array} \right\}$$

where $U', U'_1, U'_2 \in \{U_\perp, U^\circ\}$ and one of U_1 and U_2 is U°. We omit to show complete and reduced TACBB obtained because the number of transition rules becomes huge. Let $\mathcal{A}''$ be the TACBB obtained by converting $\mathcal{A}'$ to a complete and reduced TACBB.

Finally, at the third step of the algorithm, we obtain $\mathcal{A}_{\mathrm{NF}}$ from $\mathcal{A}''$ by replacing the final state. We show the set of final states and the set of transition rules of $\mathcal{A}_{\mathrm{NF}}$ in the following. However, since the size of $\mathcal{A}_{\mathrm{NF}}$ originally obtained from the algorithm is huge, we show a minified one. If we minify TACBB obtained by the algorithm, Proposition 81 may not hold. Therefore, we should not minify the TACBB obtained by the algorithm. In the case of the following TACBB, Proposition 81 holds.

Each component of $\mathcal{A}_{\mathrm{NF}}$ is as follows:

$$
\begin{aligned}
&Q_{\mathrm{NF}} = \{u_b, u_\perp, u^\circ\},\\
&Q^f_{\mathrm{NF}} = \{u_b, u_\perp\}, \text{ and}\\
&\Delta_{\mathrm{NF}} = \left\{ \begin{array}{rclrclrcl}
a & \xrightarrow{\top} & u^\circ, & b & \xrightarrow{\top} & u_b, & c & \xrightarrow{\top} & u_\perp,\\
f(u^\circ, u^b) & \xrightarrow{\top} & u^\circ, & f(u_1, u_2) & \xrightarrow{\top} & u_\perp, & g(u_3, u_3) & \xrightarrow{1=2} & u^\circ,\\
g(u_3, u_3) & \xrightarrow{1\neq 2} & u_\perp, & g(u^\circ, u^\circ) & \xrightarrow{\top} & u^\circ, & g(u_1, u_4) & \xrightarrow{\top} & u^\circ,\\
h(u_1, u_1) & \xrightarrow{\top} & u_\perp, & h(u_1, u_4) & \xrightarrow{\top} & u_\perp & & &
\end{array} \right\}
\end{aligned}
$$

where $u_1, u_4 \in Q_{\mathrm{NF}}$, $u_2 \in \{u^\circ, u_\perp\}$, $u_3 \in Q^f_{\mathrm{NF}}$, and $u_1 \neq u_4$. □

For the constructed TA $\mathcal{A}_{\mathrm{NF}}$, the following proposition holds from Lemmas 78 and 79. These lemmas are used in the next subsection.

Proposition 81 *Let $t \in \mathcal{T}(F)$, $u \in Q_{\mathrm{NF}}$, and $t \xrightarrow[\Delta_{\mathrm{NF}}]{*} u$. If t is a proper subterm of some l of $l \to r \in R$ where R is a shallow TRS, then u accepts no term other than t.*

Proof: Let $\mathcal{A}^d_l$ be the deterministic TACBB obtained by determinizing $\mathcal{A}_l$ in the algorithm. From Lemma 76, Lemma 78, and shallowness of R, we have $t \xrightarrow[\Delta_l]{*} S \in Q^d_l$ where S contains $u_t \in Q_l$ and there exists no term other than t accepted by u_t. Thus, from the construction of Δ_{NF}, there exists no term accepted by u other than t. □

Proposition 82 *If $f(u_1, \ldots, u_n) \xrightarrow{c} u \in \Delta_{\mathrm{NF}}$ and $u \in Q^f_{\mathrm{NF}}$, then $i \in \mu(f)$ implies $u_i \in Q^f_{\mathrm{NF}}$.*

Proof: Let $f(u_1, \ldots, u_n) \xrightarrow{c} u \in \Delta_{\mathrm{NF}}$, $u \in Q^f_{\mathrm{NF}}$, and assume $u_i \notin Q^f_{\mathrm{NF}}$ for some $i \in \mu(f)$. Since $\mathcal{A}_{\mathrm{NF}}$ is a reduced TACBB from Lemma 79, there exist ground terms $t_1, \ldots, t_n$ such that $t_j \xrightarrow[\Delta_{\mathrm{NF}}]{*} u_j$ for each j ($1 \leq j \leq n$). Hence we have $f(t_1, \ldots, t_n) \xrightarrow[\Delta_{\mathrm{NF}}]{*} f(u_1, \ldots, u_n) \xrightarrow[\Delta_{\mathrm{NF}}]{} u$. Here $f(t_1, \ldots, t_n) \in \mathrm{NF}^\mu_{\mathcal{R}}$ and $t_i \notin \mathrm{NF}^\mu_R$ follows from Lemma 79. Since t_i is not a context-sensitive normal form and

$i \in \mu(f)$, the term $f(t_1, \ldots, t_n)$ is not a context-sensitive normal form, which contradicts $f(t_1, \ldots, t_n) \in \mathrm{NF}^{\mu}_{R}$. □

5.1.3 Tree Automata Accepting Reachable Terms

In this subsection, we show a concrete definition of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ to construct a TACBB that recognizes the set of reachable terms by the context-sensitive innermost reduction of a shallow and non-erasing TRS. $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ is a modification of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$. The main difference between $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ and $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ is the number of components of each state of output automata. States of output TA of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ have an extra component that is a state of $\mathcal{A}_{\mathrm{NF}}$. Since $\mathcal{A}_{\mathrm{NF}}$ is TACBB, the output automata of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ are also TACBB.

Algorithm $\mathbf{P}^{\mathrm{s}}_{\mathrm{csin}}$:

Input A term t and a shallow and non-erasing TRS R and a replacement map μ.

Output A TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_* \rangle$ such that $\mathcal{L}(\mathcal{A}_*) = \underset{R}{\hookrightarrow}{}^{\mu}_{\mathrm{in}}[\mathcal{L}(\mathcal{A})]$.

Step 1 (initialize) 1. Prepare a TACBB $\mathcal{A}_{\mathrm{NF}}$ obtained by the algorithm in the previous section and a TA $\mathcal{A} = \langle Q, Q^f, \Delta \rangle$ similarly in $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$.

2. Let
   - $k := 0$,
   - $Q_* = \{P \mid P \in 2^Q, P \neq \emptyset\} \times \{\mathtt{a}\} \times Q_{\mathrm{NF}} \cup \{\{q\} \mid q \in Q\} \times \{\mathtt{a}\} \times Q_{\mathrm{NF}}$,
   - $Q^f_* = \{P \mid P \in 2^Q, P \cap Q^f \neq \emptyset\} \times \{\mathtt{a}\} \times Q_{\mathrm{NF}}$, and
   - Δ_0 as follows:
     (a) $f(\langle \{q_1\}, \mathtt{i}, u_1 \rangle, \ldots, \langle \{q_n\}, \mathtt{i}, u_n \rangle) \overset{c}{\to} \langle \{q\}, \mathtt{i}, u \rangle \in \Delta_0$ where $f(q_1, \ldots, q_n) \to q \in \Delta$ and $f(u_1, \ldots, u_n) \overset{c}{\to} u \in \Delta_{\mathrm{NF}}$, and
     (b) $f(\langle \{q_1\}, \mathtt{x}_1, u_1 \rangle, \ldots, \langle \{q_n\}, \mathtt{x}_n, u_n \rangle) \overset{c}{\to} \langle \{q\}, \mathtt{a}, u \rangle \in \Delta_0$ where $f(q_1, \ldots, q_n) \to q \in \Delta$, $f(u_1, \ldots, u_n) \overset{c}{\to} u \in \Delta_{\mathrm{NF}}$, and if $i \in \mu(f)$ then $\mathtt{x}_i = \mathtt{a}$, otherwise $\mathtt{x}_i = \mathtt{i}$.

Step 2 Let Δ_{k+1} be the set of transition rules produced by augmenting transition rules of Δ_k If there exists $\sigma : X \to \mathcal{T}(F)$ such that $f(l_1, \ldots, l_n)\sigma \xrightarrow[\Delta_k]{*} f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow[\Delta_k]{} \langle \{q\}, \mathtt{a}, u\rangle$ and we have $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, then we apply the following inference rules:

$$\frac{f(l_1, \ldots, l_n) \to g(r_1, ..., r_m) \in R, \quad f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle \{q\}, \mathtt{a}, u\rangle \in \Delta_k}{g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow{c'} \langle \{q\}, \mathtt{a}, u'\rangle \in \Delta_{k+1}}$$

Let $I_j = \{i \mid l_i = r_j\}$. Each P'_j, $\mathtt{x}'_j$, u'_j, c', and u' is determined as follows:

- – $P'_j = \begin{cases} \{q^{r_j}\} & \cdots \text{ if } r_j \notin X, \\ P_i & \cdots \text{ if } r_j \in X \wedge \exists i \in I_j.\mathtt{x}_i = \mathtt{i}, \text{ and} \\ \bigcup_{i \in I_j} P_i & \cdots \text{ if } r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a}. \end{cases}$
  – $\mathtt{x}'_j = \begin{cases} \mathtt{i} \cdots \text{ if } j \notin \mu(g) \wedge (r_j \in X \Rightarrow \exists i \in I_j.\mathtt{x}_i = \mathtt{i})), \text{ and} \\ \mathtt{a} \cdots \text{ if } j \in \mu(g) \vee (r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{a})). \end{cases}$
  – $u'_j = \begin{cases} u_i & \cdots \text{ if } r_j \in X \wedge (j \in \mu(g) \Rightarrow \forall i \in I_j.\mathtt{x}_i = \mathtt{a}), \text{ and} \\ v \in Q_{\mathrm{NF}} & \cdots \text{ if } r_j \notin X \vee (j \notin \mu(g) \wedge \exists i \in I_j.\mathtt{x}_i = \mathtt{i}) \end{cases}$
- $c' = c_1 \wedge c_2 \wedge c_3$ that is a satisfiable constraint, where
  – $c_1 = \displaystyle\bigwedge_{r_i = r_j \in X, \neg\exists k \in I_j.\mathtt{x}_k = \mathtt{i}} i = j$
  – c_2 is obtained from c by replacing equality and disequality between i and j in c as follows. Let i' and j' be as $l_i = r_{i'}$ and $l_j = l_{j'}$.
    * If $u_i \neq u_j$, we replace $i = j$ in c by $\bot$ and $i \neq j$ by $\top$.
    * If $u_i = u_j \in Q_{\mathrm{NF}} \backslash Q^f_{\mathrm{NF}}$, we consider the following two cases:
      · In the subcase of $P_i = P_j$, we replace $i = j$ in c by $\top$ and $i \neq j$ by $\bot$.
      · In the subcase of $P_i \neq P_j$, we replace $i = j$ in c by $\bot$ and $i \neq j$ by $\top$.
    * If $u_i = u_j \in Q^f_{\mathrm{NF}}$, we consider the following two cases:
      · In the subcase of $l_i \neq l_j$ and $l_i, l_j \in X$, we replace $i = j$ in c by $i' = j'$ and $i \neq j$ by $i' \neq j'$.

· Otherwise, we replace $i = j$ in c by $\top$ and $i \neq j$ by $\perp$.

$*$ If $i > n$ or $j > n$, then replace $i = j$ and $i \neq j$ by $\perp$.

– $g(u'_1, \ldots, u'_m) \xrightarrow{c_3} u' \in \Delta_{\mathrm{NF}}$.

Note that c' is not unique because we may choose more than one constraint for c_3, and also that the role of c_2 is to preserve the constraints for variables in the rewrite rule applied at the inference rule.

The second inference rule is the following:

$$\frac{f(l_1, ..., l_n) \to x \in R,\ f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle\{q\}, \mathtt{a}, u\rangle \in \Delta_k}{\langle P', \mathtt{a}, u\rangle \xrightarrow{\top} \langle\{q\}, \mathtt{a}, u\rangle \in \Delta_{k+1}}$$

Let $I = \{i \mid l_i = x\}$. P' is determined as follows:

- $P' = \begin{cases} P_i & \cdots \text{ if } \exists i \in I.\mathtt{x}_i = \mathtt{i}, \text{ and} \\ \bigcup_{i \in I} P_i & \cdots \text{ if } \forall i \in I.\mathtt{x}_i = \mathtt{a}. \end{cases}$

Step 3 For all states $\langle P^1 \cup P^2, \mathtt{a}, u\rangle \in Q_*$ where $P^1 \neq P^2$, we add the new transition rules to Δ_{k+1} as follows:[1]

1. $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c'} \langle P^1 \cup P^2, \mathtt{a}\rangle \in \Delta_{k+1}$ where
   - $P_i = \begin{cases} P_i^j & \cdots \begin{array}{l} \text{if } \mathtt{x}_i^j = \mathtt{i} \text{ for some } j \in \{1,2\} \text{ and} \\ \mathcal{L}(\Delta_k, \langle P_i^1, \mathtt{x}_i^1, u_i\rangle) \cap \mathcal{L}(\Delta_k, \langle P_i^2, \mathtt{x}_i^2, u_i\rangle) \neq \emptyset \end{array} \\ P_i^1 \cup P_i^2 & \cdots \text{ if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a} \end{cases}$
   - $\mathtt{x}_i = \begin{cases} \mathtt{a} \cdots \text{if } \mathtt{x}_i^1 = \mathtt{x}_i^2 = \mathtt{a} \\ \mathtt{i} \cdots \text{otherwise} \end{cases}$
   - $c' = c^1 \wedge c^2$.

   if $f(\langle P_1^j, \mathtt{x}_1^j, u_1\rangle, \ldots, \langle P_n^j, \mathtt{x}_n^j, u_1\rangle) \xrightarrow{c^j} \langle P^j, \mathtt{a}, u\rangle \in \Delta_k$ for $j \in \{1,2\}$.
   Note that if $\mathcal{L}(\Delta_k, \langle P_i^1, \mathtt{x}_i^1, u_i\rangle) \cap \mathcal{L}(\Delta_k, \langle P_i^2, \mathtt{x}_i^2, u_i\rangle) = \emptyset$ and $\mathtt{x}_i^j = \mathtt{i}$ for some $j \in \{1,2\}$, then the transition rule is not produced.

[1]This step is almost the same as the step of $\mathrm{P}_{\mathrm{cs}}^{\mathrm{rlrs}}$ because we do not need to be concerned about third components of states in each transition rule $f(\langle P_1^j, \mathtt{x}_1^j, u_1\rangle, \ldots, \langle P_n^j, \mathtt{x}_n^j, u_n\rangle) \xrightarrow{c^j} \langle P^j, \mathtt{a}, u\rangle$.

2. $\langle P_1' \cup P_2', \mathtt{a}, u\rangle \xrightarrow{\top} \langle P_1 \cup P_2, \mathtt{a}, u\rangle \in \Delta_{k+1}$ if $\langle P_1', \mathtt{a}, u\rangle \xrightarrow{\top} \langle P_1, \mathtt{a}, u\rangle \in \Delta_k$, and, $\langle P_2', \mathtt{a}, u\rangle \xrightarrow{\top} \langle P_2, \mathtt{a}, u\rangle \in \Delta_k$ or $P_2' = P_2$.

Step 4 If $\Delta_{k+1} = \Delta_k$ then stop and set $\Delta_* = \Delta_k$. Otherwise, $k := k+1$, and go to step 2.

Here, we show an example that shows how $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ works.

Example 83 Let a TRS R and a replacement map μ be in Example 80. We input $f(a,b)$, R, and μ to $\mathrm{P}_{\mathrm{csin}}$. Here, we have:

$$\hookrightarrow^{\mu}_{R\,\mathrm{in}}[\{f(a,b)\}] = \left\{ \begin{array}{c} f(a,b), g(a,a), g(b,a), g(a,b), g(b,b), g(c,a), \\ g(c,b), g(a,c), g(b,c), g(c,c), h(b,b), h(c,c) \end{array} \right\}.$$

In the initializing step, at (1) of Step 1 of $\mathrm{P}_{\mathrm{csin}}$, we have the TA $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ where:

$$\begin{array}{l} Q = \{q^a, q^b, q^{f(a,b)}\}, \\ Q^f = \{q^{f(a,b)}\}, \text{ and} \\ \Delta = \{a \to q^a, b \to q^b, f(q^a, q^b) \to q^{f(a,b)}\}, \end{array}$$

and TACBB $\mathcal{A}_{\mathrm{NF}}$ as a previous subsection. At (2) of Step 1, we have

$$\begin{array}{l} Q_* = \{\langle P, \mathtt{a}, u\rangle, \langle \{p\}, \mathtt{i}, u\rangle \mid P \in 2^Q, P \neq \emptyset, p \in Q, u \in Q_{\mathrm{NF}}\} \\ Q_*^f = \{\langle P^f, \mathtt{a}, u\rangle \mid P^f \in 2^Q, P^f \cap Q^f \neq \emptyset, u \in Q_{\mathrm{NF}}\} \end{array}$$

$$\Delta_0 = \left\{ \begin{array}{rcl} a & \xrightarrow{\top} & \langle \{q^a\}, \mathtt{x}, u^\circ\rangle, \\ b & \xrightarrow{\top} & \langle \{q^b\}, \mathtt{x}, u_b\rangle, \\ f(\langle \{q^a\}, \mathtt{i}, u^\circ\rangle, \langle \{q^b\}, \mathtt{i}, u_b\rangle) & \xrightarrow{\top} & \langle \{q^{f(a,b)}, \mathtt{x}, u^\circ\}\rangle \\ f(\langle \{q^a\}, \mathtt{i}, u_1\rangle, \langle \{q^b\}, \mathtt{i}, u_2\rangle) & \xrightarrow{\top} & \langle \{q^{f(a,b)}, \mathtt{x}, u_\perp\}\rangle \end{array} \right\}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$, $u_1 \in Q_{\mathrm{NF}}$, and $u_2 \in \{u^\circ, u_\perp\}$.

In the saturation step, at $k = 0$, we have Δ_1 as follows:

$$\Delta_1 = \Delta_0 \cup \left\{ \begin{array}{rcl} b & \xrightarrow{\top} & \langle \{q^a\}, \mathtt{x}, u_b\rangle, \\ c & \xrightarrow{\top} & \langle \{q^a\}, \mathtt{x}, u_\perp\rangle, \\ g(\langle \{q^a\}, \mathtt{a}, u_3\rangle, \langle \{q^a\}, \mathtt{a}, u_3\rangle) & \xrightarrow{1=2} & \langle \{q^{f(a,b)}\}, \mathtt{a}, u^\circ\rangle \\ g(\langle \{q^a\}, \mathtt{a}, u_3\rangle, \langle \{q^a\}, \mathtt{a}, u_3\rangle) & \xrightarrow{1\neq 2} & \langle \{q^{f(a,b)}\}, \mathtt{a}, u_\perp\rangle \\ g(\langle \{q^a\}, \mathtt{a}, u^\circ\rangle, \langle \{q^a\}, \mathtt{a}, u^\circ\rangle) & \xrightarrow{\top} & \langle \{q^{f(a,b)}\}, \mathtt{a}, u^\circ\rangle \\ g(\langle \{q^a\}, \mathtt{a}, u_1\rangle, \langle \{q^a\}, \mathtt{a}, u_4\rangle) & \xrightarrow{\top} & \langle \{q^{f(a,b)}\}, \mathtt{a}, u^\circ\rangle \end{array} \right\}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$, $u_1, u_4 \in Q_{\mathrm{NF}}$, $u_3 \in Q^f_{\mathrm{NF}}$, and $u_1 = u_4$ at Step 2.

At $k = 1$, we have Δ_2 as follows:

$$\Delta_2 = \Delta_1 \cup \left\{ \; h(\langle \{q^a\}, \mathtt{a}, u_3\rangle, \langle \{q^a\}, \mathtt{a}, u_3\rangle) \;\xrightarrow{1=2}\; \langle \{q^{f(x,b)}\}, \mathtt{a}, u_\bot\rangle \; \right\}$$

where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$ and $u_3 \in Q^f_{\mathrm{NF}}$, and $\{b \xrightarrow{\top} \langle \{q^a, q^b\}, \mathtt{x}, u_b\rangle\}$ at Step 3 where $\mathtt{x} \in \{\mathtt{a}, \mathtt{i}\}$.

The saturation step at $k = 2$, and we have $\Delta_* = \Delta_1$. TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_*\rangle$ holds that $\mathcal{L}(\mathcal{A}_*) = \hookrightarrow^{\mu}_{R\,\mathrm{in}}[\{f(a,b)\}]$. □

This algorithm $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ eventually terminates at some k and apparently $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$ similarly to the other algorithms in this paper.

In the following, we show the correctness of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

First, we show several propositions. Since Propositions 84–87 below are similar to the case of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$ or $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{csin}}$, we omit their proofs.

Proposition 84 *Let $s \in \mathcal{T}(F)$, Then $s \xrightarrow[\Delta]{*} q^s \in Q$ iff $s \xrightarrow[\Delta_0]{*} \langle \{q^s\}, \mathtt{i}, u\rangle$ for some $u \in Q_{NF}$.*

Proof: This proposition is proved similarly to the proof of Proposition 57. □

Proposition 85 *Let $t \in \mathcal{T}(F)$. For any k, if $t \xrightarrow[\Delta_k]{*} \langle P, \mathtt{i}, u\rangle$, then $t \xrightarrow[\Delta_0]{*} \langle P, \mathtt{i}, u\rangle$. Moreover, P is of the form $\{q\}$.*

Proof: This proposition is proved similarly to the proof of Proposition 58. □

Proposition 86 *Let $t \in \mathcal{T}(F)$. Then, for any k, $t \xrightarrow[\Delta_0]{*} \langle P, \mathtt{a}, u\rangle$ iff $t \xrightarrow[\Delta_0]{*} \langle P, \mathtt{i}, u\rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 59. □

Proposition 87 *Let $t \in \mathcal{T}(F)$. Then, $t \xrightarrow[\Delta_k]{*} \langle P, \mathtt{i}, u\rangle$ implies $t \xrightarrow[\Delta_k]{*} \langle P, \mathtt{a}, u\rangle$.*

Proof: This proposition is proved similarly to the proof of Proposition 60. □

Proposition 88 *If the rule* $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle P, \mathtt{i}, u\rangle$ *is in* Δ_*, *then it is also in* Δ_0. *Moreover,* $\mathtt{x}_i = \mathtt{i}$ *for all* $1 \leq i \leq n$.

Proof: This proposition is proved similarly to the proof of Proposition 33. □

Proposition 89 *If the rule* $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle P, \mathtt{a}, u\rangle$ *is in* Δ_*, *then* $i \in \mu(f)$ *implies* $\mathtt{x}_i = \mathtt{a}$.

Proof: From the construction of the transition rule. □

Next we show several technical lemmas. These are necessary to prove the completeness and the soundness of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$. We omit the proofs of Lemma 91 and 95–97 since their proofs are similar to the case of $\mathrm{P}^{\mathrm{rlrs}}_{\mathrm{cs}}$. However, Since the output of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ is TACBB, we cannot prove some lemmas similarly in previous chapters.

Lemma 90 *For any* k, *if* $\alpha : t \xrightarrow[\Delta_k]{*} \langle P, \mathtt{x}, u\rangle$, *then* $t \xrightarrow[\Delta_{\mathrm{NF}}]{*} u$.

Proof: We show the proof by induction on $|\alpha|$. If the last transition rule applied in α is of the form $\langle P', \mathtt{x}, u\rangle \xrightarrow{\top} \langle P, \mathtt{x}, u\rangle$, then we have $t \xrightarrow[\Delta_{\mathrm{NF}}]{*} u$ from the induction hypothesis. Otherwise, let the last transition rule applied in $|\alpha|$ is $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle P, \mathtt{x}, u\rangle$.

If $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle P, \mathtt{x}, u\rangle \in \Delta_k$, then we have $f(u_1, \ldots, u_n) \xrightarrow{c'} u \in \Delta_{\mathrm{NF}}$ where c is of the form $c = c'' \wedge c'$ for some c'. Thus, if t satisfies c then c' is also satisfied. Since we have $t|_i \xrightarrow[\Delta_{\mathrm{NF}}]{*} u_i$ from the induction hypothesis, we have $t \xrightarrow[\Delta_{\mathrm{NF}}]{*} f(u_1, \ldots, u_n) \xrightarrow[\Delta_{\mathrm{NF}}]{} u$. □

Lemma 91 *Let* $s, t \in \mathcal{T}(F)$, $s \xrightarrow[\Delta_0]{*} \langle P, \mathtt{x}, u\rangle$, *and* $t \xrightarrow[\Delta_0]{*} \langle P', \mathtt{x}', u'\rangle$. *Then,* $P = P'$ *iff* $s = t$.

Proof: This lemma is proved similarly to the proof of Lemma 61. □

Lemma 92 *Let* $\alpha :\ t \xrightarrow[\Delta_*]{*} t[\langle P, \mathtt{a}, u\rangle]_p \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}, u'\rangle$ *and* $p \in Pos^{\mu}(s)$. *Then* $u' \in Q^f_{\mathrm{NF}}$ *implies* $u \in Q^f_{\mathrm{NF}}$.

Proof: We show this lemma by induction on $|\alpha|(> 0)$.

1. Consider the case where the last transition rule applied in α is (of the form) $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle P, \mathtt{a}, u'\rangle \in \Delta_*$. Then α can be represented as $t \xrightarrow[\Delta_*]{*} t[\langle P, \mathtt{a}, u\rangle]_p \xrightarrow[\Delta_*]{*} f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow[\Delta_*]{} \langle P, \mathtt{a}, u'\rangle$.

   In this case, the position p can be represented as ip' for $1 \leq i \leq m$. From the construction of the transition rule $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle P, \mathtt{a}, u'\rangle$, we have the transition rule $f(u_1, \ldots, u_n) \xrightarrow{c'} u' \in \Delta_{\mathrm{NF}}$ for some c'. Therefore, from $i \in \mu(g)$ and Proposition 82, we have $u_i \in Q^f_{\mathrm{NF}}$ and hence we also have $u \in Q^f_{\mathrm{NF}}$ from the induction hypothesis.

2. In the case where the last transition rule applied in α is (of the form) $\langle P', \mathtt{a}, u'\rangle \xrightarrow{c} \langle P, \mathtt{a}, u\rangle \in \Delta_k$, we have $u' = u$ from the construction of Δ_0 or the second inference rule of Step 2. Hence this lemma holds by the induction hypothesis. □

Lemma 93 *If $j \notin \mu(g)$ and $g(\ldots, \langle P'_j, \mathtt{x}'_j, u'_j\rangle, \ldots) \xrightarrow{c} \langle P, \mathtt{x}', u'\rangle \in \Delta_*$, then $u'_j \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}'_j = \mathtt{i}$.*

Proof: From the construction of Δ_*, we can assume that $g(\ldots, \langle P'_j, \mathtt{x}'_j, u'_j\rangle, \ldots) \xrightarrow{c} \langle P, \mathtt{x}', u'\rangle \in \Delta_k$. We show the proof of this lemma for the following two cases.

1. If $k = 0$, then $\mathtt{x}'_j = \mathtt{i}$ from the construction of Δ_0

2. Consider the case of $k > 0$. We can assume $g(\ldots, \langle P'_j, \mathtt{x}'_j, u'_j\rangle, \ldots) \xrightarrow{c} \langle P, \mathtt{x}', u'\rangle \in \Delta_k \backslash \Delta_{k-1}$ without loss of generality. This rule is introduced by (1) of Step 2 or (1) of Step 3. In the latter case, if $\mathtt{x}'_j = \mathtt{a}$, then we have $g(\ldots, \langle P''_j, \mathtt{a}, u'_j\rangle, \ldots) \xrightarrow{c'} \langle \{q\}, \mathtt{x}', u'\rangle \in \Delta_k$ for any $q \in P$ where this rule is in Δ_0 or produced by (1) of Step 2. Therefore, if we prove the former case, we can also prove the latter case. In the former case, $\mathtt{x}'_j = \mathtt{a}$ implies $\mathtt{x}' = \mathtt{a}$ from Proposition 88, and there exist $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$ and $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle P, \mathtt{a}, u\rangle \in \Delta_{k-1}$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$. If $j \notin \mu(g)$ and $\mathtt{x}'_j = \mathtt{a}$, then there exists some i such that $u_i = u'_j$ and $\mathtt{x}_{i'} = \mathtt{a}$ for all i' such that $l_{i'} = r_j$. Hence we have $u_i = u'_j \in Q^f_{\mathrm{NF}}$. □

Lemma 94 *Let* $\alpha : t[t']_p \xrightarrow[\Delta_*]{*} t[\langle P, \mathtt{a}, u\rangle]_p \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}, u'\rangle$. *If* $u \in Q_{NF} \setminus Q^f_{NF}$ *and* $p \in Pos^\mu(t)$, *then there exists* $v' \in Q_{NF}$ *such that* $t[t']_p \xrightarrow[\Delta_*]{*} t[\langle P, \mathtt{a}, v\rangle]_p \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}, v'\rangle$ *for any* $v \in Q_{NF}$.

Proof: We prove this lemma by induction on $|\alpha|$ (> 0).

1. Consider the case where the last transition rule applied in α is (of the form) $g(\langle q'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle q'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow{c} \langle P', \mathtt{x}', u'\rangle \in \Delta_*$. Then α can be represented as $t[t']_p \xrightarrow[\Delta_*]{*} t[\langle P, \mathtt{x}, u\rangle]_p \xrightarrow[\Delta_*]{*} g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow[\Delta_*]{} \langle P', \mathtt{x}', u'\rangle$. Let $p = jp'$ where $1 \leq j \leq n$.

   If the rule $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow{c} \langle P, \mathtt{x}', u\rangle$ is in Δ_0, the rule is produced at (2) of Step 1 of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$. Therefore, for any $u''_j \in Q_{\mathrm{NF}}$, there exists the constraints c'' and $u'' \in Q_{\mathrm{NF}}$ such that $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_j, \mathtt{x}'_j, u''_j\rangle, \ldots, \langle P'_n, \mathtt{x}'_n, u'_n\rangle) \xrightarrow{c''} \langle P, \mathtt{x}, u''\rangle \in \Delta_0$ where t satisfies c'' from the the completeness of $\mathcal{A}_{\mathrm{NF}}$. Hence this lemma holds from the induction hypothesis.

   Consider the case where the rule $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow{c} \langle P, \mathtt{x}', u\rangle$ is in $\Delta_k \backslash \Delta_{k-1}$ for $k > 0$. In this case, $j \in \mu(g)$ from $p \in \mathrm{Pos}^\mu(s)$, and we have $\mathtt{x}'_j = \mathtt{a}$ from $\mathtt{x} = \mathtt{a}$ and Proposition 89. For $\alpha_j : (t|_j)[t']_{p'} \xrightarrow[\Delta_*]{*} (t|_j)[\langle P, \mathtt{a}, u\rangle]_{p'} \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j, u'_j\rangle$, we have $(t|_j)[t']_{p'} \xrightarrow[\Delta_*]{*} (t|_j)[\langle P, \mathtt{a}, v\rangle]_{p'} \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{a}, v'_j\rangle$ for some $v'_j \in Q_{\mathrm{NF}}$ from the induction hypothesis. Note that we have $u'_j \notin Q^f_{\mathrm{NF}}$ from $u \notin Q^f_{\mathrm{NF}}$ and Lemma 92. Thus, we prove there exists the transition rule $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_j, \mathtt{a}, v'_j\rangle, \ldots, \langle P'_n, \mathtt{x}'_n, u'_n\rangle) \xrightarrow{c'} \langle P', \mathtt{x}, v'\rangle \in \Delta_*$.

   Here, there are two cases in which the rule $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_j, \mathtt{a}, u'_j\rangle, \ldots, \langle P'_n, \mathtt{x}'_n, u'_n\rangle) \xrightarrow{c'} \langle P', \mathtt{x}, u\rangle \in \Delta_*$ is produced in (1) of Step 2 or (1) of Step 3 of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$. In the former case, there exist $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$ and $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c'} \langle P', \mathtt{a}, u''\rangle \in \Delta_{k-1}$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \leq i \leq n$, and the constraint c is of the form $c_1 \wedge c_2$ where $g(\ldots, u'_j, \ldots) \xrightarrow{c_2} u \in \Delta_{\mathrm{NF}}$.

   In the subcase $r_j \notin X$, we have $g(\ldots, \langle P'_j, \mathtt{a}, v'_j\rangle, \ldots) \xrightarrow{c'} \langle P', \mathtt{a}, v'\rangle \in \Delta_k \setminus \Delta_{k-1}$ for any v'_j and c' is of the form $c_1 \wedge c'_2$ where $g(\ldots, v'_j, \ldots,) \xrightarrow{c'_2} v' \in$

Δ_{NF}. Moreover, from the completeness of Δ_{NF}, we have c'_2 that is satisfied by $t[t']_p$.

In the remaining subcase $r_j \in X$, we have $l_i = r_j$ such that $\mathtt{x}_i = \mathtt{i}$ for some i; otherwise we have $u'_j = u_i$ from $j \in \mu(g)$ and $\mathtt{x}_i = \mathtt{a}$ for any i such that $l_i = r_j$. Hence we have $u_i \in Q^f_{\mathrm{NF}}$. This contradicts $u'_j = u_i$ and $u'_j \notin Q^f_{\mathrm{NF}}$. Thus, we have the transition rule $g(\ldots, \langle P'_j, \mathtt{a}, v'_j\rangle, \ldots) \xrightarrow{c'} \langle P', \mathtt{a}, v'\rangle$ for any v'_j and $s[s']_p$ satisfies c' by the same reason in the case of $r_j \notin X$.

If the transition rule is produced at (1) of Step 3, we have $g(\ldots, \langle P''_j, \mathtt{a}, u_j\rangle, \ldots) \xrightarrow{c''} \langle \{q\}, a, u'\rangle$ where c is of the form $c'' \wedge c'''$ for any $q \in P$ and some c'''. From the former case, we have $g(\ldots, \langle P''_j, \mathtt{a}, v_j\rangle, \ldots) \xrightarrow{c'} \langle \{q\}, a, v'\rangle$ for each q and c' that is satisfied by $t[t']_p$. Thus, we have $g(\ldots, \langle P'_j, \mathtt{a}, v_j\rangle, \ldots) \xrightarrow{c'} \langle P, a, v'\rangle$.

2. In the case where the last transition rule applied in α is (of the form) $\langle P', \mathtt{x}', u'\rangle \to \langle P, \mathtt{x}, u\rangle \in \Delta_k$, we have $u' = u$ from the construction of Δ_0, (2) of Step 2, or (2) of Step 3. Hence this lemma holds from the induction hypothesis. □

Lemma 95 *If* $\alpha : t[t']_p \xrightarrow[\Delta_k]{*} \langle P, \mathtt{a}, u\rangle$ *and* $p \in \mathrm{Pos}^{\mu}(t)$*, then there exists* $\langle P', \mathtt{a}, u'\rangle$ *such that* $t' \xrightarrow[\Delta_k]{*} \langle P', \mathtt{a}, u'\rangle$ *and* $t[\langle P', \mathtt{a}, u'\rangle]_p \xrightarrow[\Delta_k]{*} \langle P, \mathtt{a}, u\rangle$.

Proof: This lemma is proved similarly to the proof of Lemma 62. □

Lemma 96 *Let R be a shallow and non-erasing TRS and $l \to g(r_1, \ldots, r_m) \in R$. For $r_j \notin X$ and any substitution σ, $r_j\sigma \xrightarrow[\Delta_*]{*} \langle \{q^{r_j}\}, \mathtt{i}, u\rangle$ and $r_j\sigma \xrightarrow[\Delta_*]{*} \langle \{q^{r_j}\}, \mathtt{a}, u\rangle$ for some $u \in Q_{\mathrm{NF}}$.*

Proof: By Proposition 84 and 87, and the construction of Δ_0, this lemma is proved similarly to the proof of Lemma 63. □

Proofs of the following Lemmas 97– 100 are similar to the ones of Lemma 64–67, because it is not necessary to consider the third components of the states.

Lemma 97 *If* $\langle P_1', \mathsf{a}, u\rangle \xrightarrow[\Delta_*]{*} \langle P_1, \mathsf{a}, u\rangle$ *and* $\langle P_2', \mathsf{a}, u\rangle \xrightarrow[\Delta_*]{*} \langle P_2, \mathsf{a}, u\rangle$*, then we have* $\langle P_1' \cup P_2', \mathsf{a}, u\rangle \xrightarrow[\Delta_*]{*} \langle P_1 \cup P_2, \mathsf{a}, u\rangle$.

Proof: This lemma is proved similarly to the proof of Lemma 64. □

Lemma 98 *If* $\langle P_1, \mathsf{a}, u\rangle \xrightarrow[\Delta_*]{*} \langle P, \mathsf{a}, u\rangle$*, then there exists* $P_1' \subseteq P_1$ *such that* $\langle P_1', \mathsf{a}, u\rangle \xrightarrow[\Delta_*]{*} \langle P', \mathsf{a}, u\rangle$ *for all* $P' \subseteq P$.

Proof: The proof of this lemma is similar to the proof of Lemma 98. □

Lemma 99 *If* $t \xrightarrow[\Delta_*]{*} \langle P^j, \mathsf{a}, u\rangle$ *for* $1 \leq j \leq m$*, then we have* $t \xrightarrow[\Delta_*]{*} \langle \bigcup_{1\leq j\leq m} P^j, \mathsf{a}, u\rangle$.

Proof: The proof of this lemma is similar to the proof of Lemma 65 since Lemma 97 is proved. The difference between proofs of this lemma and Lemma 65 is in the constraints. However, since the constraints of the transition rules produced at (1) of Step 3 of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$ are simple, the difference does not cause difficulty. □

Lemma 100 *If* $t \xrightarrow[\Delta_*]{*} \langle P, \mathsf{a}, u\rangle$*, then* $t \xrightarrow[\Delta_*]{*} \langle P', \mathsf{a}, u\rangle$ *for any* $P' \subseteq P$.

Proof: The proof of this lemma is similar to the proof of Lemma 67 since Lemma 98 is proved. Similarly to the proof of Lemma 99, there is the difference between proofs of this lemma and Lemma 67 but the difference does not cause difficulty. □

The following lemma is the key lemma to prove the completeness of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

Lemma 101 *If* R *be a shallow and non-erasing TRS, then* $s \xrightarrow[\Delta_*]{*} \langle P, \mathsf{a}, u\rangle$ *and* $s \overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}}\, t$ *imply* $t \xrightarrow[\Delta_*]{*} \langle P, \mathsf{a}, u'\rangle$ *for some* $u' \in Q_{NF}$.

Proof: Similarly to the proof of Lemma 14, we show the proof in the case where $s \underset{R}{\hookrightarrow}{}^{\mu}_{\mathrm{in}}\, t$ and let $s \xrightarrow[\Delta_*]{*} \langle P, \mathsf{a}, u\rangle$ and $s = s[l\sigma]_p \underset{R}{\hookrightarrow}{}^{\mu}_{\mathrm{in}} s[r\sigma]_p = t$ for some rewrite rule $l \to r \in R$, where $p \in \mathrm{Pos}^{\mu}(s)$. Then we have the transition sequence $s[l\sigma]_p \xrightarrow[\Delta_*]{*} s[\langle P', \mathsf{a}, u''\rangle]_p \xrightarrow[\Delta_*]{*} \langle P, \mathsf{a}, u\rangle$ by Lemma 95. Thanks to Lemma 94, it is sufficient to show $r\sigma \xrightarrow[\Delta_*]{*} \langle P', \mathtt{a}, v\rangle$ for some v. Moreover, similarly to the proof of Lemma 68, it is sufficient to show that $r\sigma \xrightarrow[\Delta_*]{*} \langle \{q\}, \mathtt{a}, v\rangle$ where $l\sigma \xrightarrow[\Delta_*]{*} \langle \{q\}, \mathtt{a}, u''\rangle$ and $q \in P'$ from Lemma 99 and Lemma 100 and determinacy of $\mathcal{A}_{\mathrm{NF}}$.

We assume that $l \to r$ is represented by $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m)$. Since the proof of the other case is similar, we omit it. Then we have the transition $l\sigma = f(l_1, \ldots, l_n)\sigma \xrightarrow[\Delta_*]{*} f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow[\Delta_*]{} \langle \{q\}, \mathtt{a}, u''\rangle$, and we have $i \notin \mu(g)$ or $u_i \in Q^f_{\mathrm{NF}}$ for $1 \leq i \leq n$ from the construction of Δ_* and Lemma 93.

Thus, there exists the transition rule $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow{c'} \langle \{q\}, \mathtt{a}, u'\rangle \in \Delta_*$. Each $\langle P'_j, \mathtt{x}'_j, u'_j\rangle$ and c is defined as Step 2 of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$. Since we can take u'_j as u_i where $j = i$, $r_j\sigma \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j, u'_j\rangle$ holds similarly to the proof of Lemma 68 from Proposition 87, Lemma 96, and Lemma 99.

Here, we must show that the term $g(r_1, \ldots, r_m)\sigma$ satisfies the constraint c'. We show that if $f(l_1, \ldots, l_n)\sigma$ satisfies the constraint c of the transition rule $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow[\Delta_*]{} \langle \{q\}, \mathtt{a}, u\rangle$, then there exists c' satisfied by $g(r_1, \ldots, r_m)\sigma$. In the following, we assume the constraints c_1, c_2, and c_3 are the same as the definition of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

1. $g(r_1, \ldots, r_m)\sigma$ trivially satisfies c_1 because we have $r_i\sigma = r_j\sigma$ for $r_i = r_j \in X$ obviously.

2. Here, we prove the claim that if $f(l_1, \ldots, l_n)\sigma$ satisfies c then $g(r_1, \ldots, r_m)\sigma$ satisfies c_2. We describe the constraints replaced by equality, disequality, or $\perp$.

   - For i and j such that $u_i \neq u_j$, we have $l_i\sigma \neq l_j\sigma$ from Lemma 90 and the determinacy of $\mathcal{A}_{\mathrm{NF}}$. Thus, $i = j$ is not satisfied by $f(l_1, \ldots, l_n)\sigma$, and hence there is no problem replacing $i = j$ in c by $\perp$ in c_2.

   - For i and j such that $u_i = u_j \in Q_{\mathrm{NF}} \backslash Q^f_{\mathrm{NF}}$, we have $\mathtt{x}_i = \mathtt{x}_j = \mathtt{i}$ and hence $l_i\sigma \xrightarrow[\Delta_0]{*} \langle P_i, \mathtt{x}_i, u_i\rangle$ and $l_j\sigma \xrightarrow[\Delta_0]{*} \langle P_j, \mathtt{x}_j, u_j\rangle$ from Proposition 85. Therefore, we have $P_i = P_j$ iff $l_i\sigma = l_j\sigma$ from Lemma 91. Thus, $i \neq j$ is not satisfied by $f(l_1, \ldots, l_n)\sigma$ if $P_i = P_j$ and $i = j$ is not satisfied by $f(l_1, \ldots, l_n)\sigma$ if $P_i \neq P_j$, and therefore there is no problem replacing $i \neq j$ in c by $\perp$ in c_2 if $P_i = P_j$ and $i = j$ in c by $\perp$ in c_2 if $P_i \neq P_j$.

- For i and j such that $u_i = u_j \in Q^f_{\mathrm{NF}}$, we consider the following three cases. Let i' and j' be as $l_i = r_{i'}$ and $l_j = r_{j'}$.
  - If $l_i \neq l_j$ and $l_i, l_j \in X$, then we have $l_i\sigma = l_j\sigma$ iff $r_{i'}\sigma = r_{j'}\sigma$. Thus, we have $f(l_1, \ldots, l_n)\sigma$ satisfies $i = j$ in c iff $g(r_1, \ldots, r_m)\sigma$ satisfies $i' = j'$ in c_2, and $f(l_1, \ldots, l_n)\sigma$ satisfies $i \neq j$ in c iff $g(r_1, \ldots, r_m)\sigma$ satisfies $i' \neq j'$ in c_2.
  - If $l_i = l_j \in X$, then we have $l_i\sigma = l_j\sigma$ and $r_{i'}\sigma = r_{j'}\sigma$. Thus, there is no problem to replace $i \neq j$ in c by $\perp$ in c_2.
  - If $l_i \notin X$, then we have $l_i = l_j\sigma$ from Lemma 90 and Proposition 81. Thus, there is no problem replacing $i \neq j$ in c by $\perp$ in c_2.
- For i and j such that $i > 0$ or $j > 0$, the constraints $i = j$ or $i \neq j$ is not satisfied by $f(l_1, \ldots, l_n)\sigma$ and therefore, there is no problem replacing these constraints by $\perp$.

Moreover, we have a constraint c_3 that is satisfied by $g(r_1, \ldots, r_m)\sigma$ from the completeness of Δ_{NF}. Thus, we have a constraint c' that is satisfied by $g(r_1, \ldots, r_m)\sigma$ and hence we have the transition $g(r_1, \ldots, r_m)\sigma \xrightarrow[\Delta_*]{*} g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow[\Delta_*]{} \langle\{q\}, \mathtt{a}, u'\rangle$. $\square$

The following lemma shows the completeness of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

Lemma 102 *Let R be a shallow and non-erasing TRS. Then $\mathcal{L}(\mathcal{A}_*) \supseteq \hookrightarrow^{\mu}_{R\,\mathrm{in}}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let $s \overset{*}{\hookrightarrow}{}^{\mu}_{R\,\mathrm{in}} t$ and $s \xrightarrow[\Delta]{*} q \in Q^f$. Since $s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{i}, u\rangle \in Q^f_*$ from Proposition 84, we have $s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{a}, u\rangle \in Q^f_*$ by Proposition 86. Hence $t \xrightarrow[\Delta_*]{*} \langle\{q\}, \mathtt{a}, u'\rangle \in Q^f_*$ for some $u' \in Q_{\mathrm{NF}}$ by Lemma 101. $\square$

The following lemma is the key lemma to prove the soundness of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$, and we use the measurement $[[\ ]]$ and the order $\succ$ to prove it.

Lemma 103 *If R is a shallow and non-erasing TRS, then $\alpha :\ t \xrightarrow[\Delta_*]{*} \langle P, \mathtt{x}, u'\rangle$ implies that $s \overset{*}{\hookrightarrow}{}^{\mu}_{R\,\mathrm{in}} t$ and $\beta :\ s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{x}, u\rangle$ for some term s, $q \in P$, and $u \in Q_{\mathrm{NF}}$.*

Proof: Similarly to the proof of Lemma 72, we assume that P is of the form $\{q\}$. Let $t = g(t_1, \ldots, t_m)$ and α be $g(t_1, \ldots, t_m) \xrightarrow[\Delta_*]{*} g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow[\Delta_k]{} \langle\{q\}, \mathtt{x}, u'\rangle$. In the case where the last transition rule in α is in Δ_0, then each P'_j is of the form $\{q_j\}$ and hence this lemma is proved similarly to the proof of Lemma 41 from Proposition 85.

Assume that $\Delta_k \backslash \Delta_{k-1}$ for $k > 0$. Then the last transition rule in α is $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow{c'} \langle\{q\}, \mathtt{x}, u'\rangle \in \Delta_k \backslash \Delta_{k+1}$. Since this rule is introduced at (1) of Step 2, there exist $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$, $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \ldots, \langle P_n, \mathtt{x}_n, u_n\rangle) \xrightarrow{c} \langle\{q\}, \mathtt{a}, u\rangle \in \Delta_{k-1}$ where $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all $1 \le i \le n$, $\sigma' : X \to \mathcal{T}(F)$ such that $l_i\sigma' \xrightarrow[\Delta_{k-1}]{*} \langle P_i, \mathtt{x}_i, u_i\rangle$, and $\langle P'_j, \mathtt{x}'_j, u'_j\rangle$, c', and u' are given as the definition of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

In the following, we show that there exists the substitution σ such that $g(r_1, \ldots, r_m)\sigma \xhookrightarrow[R]{*}{}^{\mu}_{\mathrm{in}} g(t_1, \ldots, t_m)$ and $\alpha' : g(r_1, \ldots, r_m)\sigma \xrightarrow[\Delta_*]{*} g(\langle P'_1, \mathtt{x}'_1, v'_1\rangle, \ldots, \langle P'_m, \mathtt{x}'_m, v'_m\rangle) \xrightarrow[\Delta_k]{} \langle\{q\}, \mathtt{a}, v'\rangle$ for some v', where $\alpha' \preceq \alpha$.

1. For j such that $r_j \notin X$, the claim holds from Lemma 96.

2. For j such that $r_j \in X(\wedge j \notin \mu(g) \exists i \in I_j.\mathtt{x}_i = \mathtt{i})$, we have $u'_j = u_i$ and hence the claim holds similarly to the proof of Lemma 72.

3. For j such that $r_j \in X \wedge (j \in \mu(g) \wedge \exists i \in I_j \mathtt{x}_i = \mathtt{i})$, we have $t_j \xrightarrow[\Delta_*]{*} \langle P'_j, \mathtt{x}'_j, u'_j\rangle = \langle P_i, \mathtt{a}, u\rangle$ where u is an arbitrary state in Q_{NF}. Since P_i is of the form $\{q_i\}$ from Proposition 85, there exists some s_j such that $s_j \xhookrightarrow[R]{*}{}^{\mu}_{\mathrm{in}} t_j$ and $s_j \xrightarrow[\Delta_0]{*} \langle P'_j, \mathtt{x}'_j, v'_j\rangle$ for some v'_j from the induction hypothesis. Let s_j be $r_j\sigma$.

4. For j such that $r_j \in X \wedge \forall i \in I_j.\mathtt{x}_i = \mathtt{i}$, we take $r_j\sigma = t_j$.

Note that the substitution σ is well-defined because for all $r_j \in X$ such that there exists i such that $l_i = r_j$ and $\mathtt{x}_i = \mathtt{i}$, we have the term $s_j(= t_j$ for $j \notin \mu(g))$ such that $s_j \xhookrightarrow[R]{*}{}^{\mu}_{\mathrm{in}} t_j$ and $s_j \xrightarrow[\Delta_0]{*} \langle P'_j, \mathtt{x}'_j, u'_j\rangle$. Since there is no term other than s_j that transits to $\langle P'_j, \mathtt{x}'_j, u'_j\rangle$, all s_j's are the same for such j. For all $r_j \in X$ such that there is no i such that $l_i = r_j$ and $\mathtt{x}_i = \mathtt{i}$, the constraint c' (c_1 in the procedure) has the equality that implies all t_j's are the same for such j.

Next we show that $g(r_1, \ldots, r_m)\sigma$ satisfies c_1 and c_2 of c' defined as the definition of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

Obviously, $g(r_1,\ldots,r_m)\sigma$ satisfies c_1 because $r_i\sigma = r_j\sigma$ for all $r_i = r_j \in X$. Moreover, it is not so difficult to show that $g(r_1,\ldots,r_m)\sigma$ satisfies c_2. This is because for all $r_j\sigma \neq t_j$, there exists i such that $l_i = r_j$ and $\mathtt{x}_i = \mathtt{i}$ and we have $u'_j = u_i$ from Lemma 90 and the determinacy of $\mathcal{A}_{\mathrm{NF}}$. In this case, we have $u_i = u'_j \notin Q^f_{\mathrm{NF}}$ and hence there is no equality or disequality that contain such j. From the completeness of Δ_{NF}, we have c'_3 that is satisfied by $g(r_1,\ldots,r_m)\sigma$. Thus we have the transition rule $g(\langle P'_1, \mathtt{x}'_1, v'_1\rangle,\ldots,\langle P'_m,\mathtt{x}'_m,v'_m\rangle) \xrightarrow{c''} \langle\{q\},\mathtt{a},v'\rangle$ where $g(r_1,\ldots,r_m)\sigma$ satisfies c''.

On the other hand, we have $f(l_1,\ldots,l_n)\sigma \xhookrightarrow[R]{\mu}_{\mathrm{in}} g(r_1,\ldots,r_m)\sigma$. Here, we show that we can construct $\beta : f(l_1,\ldots,l_n)\sigma \xrightarrow[\Delta_*]{*} f(\langle q_1,\mathtt{x}_1,u_1\rangle,\ldots,\langle q_n,\mathtt{x}_n,u_n\rangle) \xrightarrow[\Delta_{k-1}]{} \langle\{q\},\mathtt{a},v\rangle$ and hence $\beta \prec \alpha$. For $l_i \notin X$, $l_i\sigma = l_i \xrightarrow[\Delta_{k-1}]{*} \langle P_i,\mathtt{x}_i,u_i\rangle$ from Step 2 of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$. For $l_i \in X$ and there exists h such that $l_h = l_i$ and $\mathtt{x}_h = \mathtt{i}$, since there is no term other than $l_h\sigma$ that transits to $\langle P_h,\mathtt{x}_h,u_h\rangle$ from Lemma 91, we have $l_i\sigma = l_i\sigma'$ and hence $l_i\sigma \xrightarrow[\Delta_{k-1}]{*} \langle P_i,\mathtt{x}_i,u_i\rangle$. For $l_i \in X$ such that there is no k such that $l_h = l_i$ and $\mathtt{x}_h = \mathtt{i}$, we have $l_i\sigma \xrightarrow[\Delta_*]{*} \langle P_i,\mathtt{x}_i,u_i\rangle$ from Lemma 99. In the following, we show that if $g(r_1,\ldots,r_m)\sigma$ satisfies c'' then $f(l_1,\ldots,l_n)\sigma$ satisfies c.

For an $\top$ in c_2, the constraint c has an equality or disequality. We consider the following three cases:

- Consider the case where $\top$ in c_2 is obtained by replacing $i \neq j$ in c where $u_i \neq u_j$. In this case, we have $l_i \neq l_i\sigma$ and hence $f(l_1,\ldots,l_n)\sigma$ satisfies $i \neq j$.

- Consider the case where $\top$ in c_2 is obtained by replacing $i = j$ or $i \neq j$ in c where $u_i = u_j \in Q_{\mathrm{NF}} \backslash Q^f_{\mathrm{NF}}$. Then, we have $\mathtt{x}_i = \mathtt{x}_j = \mathtt{i}$. In this case, if $P_i = P_j$ then we have $i = j$ but $f(l_1,\ldots,l_n)\sigma$ satisfies it from Proposition 85 and Lemma 91, and if $P_i \neq P_j$ then we have $i \neq j$ but $f(l_1,\ldots,l_n)\sigma$ satisfies it.

- Consider the case where $\top$ in c_2 is obtained by replacing $i = j$ in c where $u_i = u_j \in Q^f_{\mathrm{NF}}$.

  - If $l_i = l_j \in X$, we have $i = j$ in c but $f(l_1,\ldots,l_n)\sigma$ satisfies it trivially.

– If $l_i \neq l_j$ and $l_i, l_j \in X$, then c does not have equality or disequality replaced by $\top$ in c_2.

– If $l_i \notin X$, we have $i = j$ in c but we have $l_i = l_j\sigma$ from Lemma 90 and Proposition 81.

Moreover, we have $i = j$ or $i \neq j$ in c for $i' = j'$ or $i' \neq j'$ in c_2. These kinds of constraints are satisfied by $f(l_1, \dots, l_n)\sigma$ similarly to the statement in Lemma 101.

Since $u_i \in Q^f_{\mathrm{NF}}$ or $\mathtt{x}_i = \mathtt{i}$ for all i, $l_i\sigma$ is a normal form or $i \notin \mu(f)$ for each i from Lemma 90 and the procedure. Hence we have $f(l_1, \dots, l_n)\sigma \xhookrightarrow[R]{\mu}_{\mathrm{in}}$ $g(r_1, \dots, r_m)\sigma$. Here $\alpha \sqsupseteq \alpha' \sqsupset \beta$ follows. Thus, we have $s \xhookrightarrow[R]{*}{}^{\mu}_{\mathrm{in}} f(l_1, \dots, l_n)\sigma \xhookrightarrow[R]{}{}^{\mu}_{\mathrm{in}}$ $g(r_1, \dots, r_m)\sigma \xhookrightarrow[R]{*}{}^{\mu}_{\mathrm{in}} g(t_1, \dots, t_m) = t$ and $s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{a}, u\rangle$ for some u by the induction hypothesis. $\Box$

If a TRS has an erasing variable, we cannot prove Lemma 103 by the above proof. Assume that the transition rule $g(\langle P'_1, \mathtt{x}'_1, u'_1\rangle, \dots, \langle P'_m, \mathtt{x}'_m, u'_m\rangle) \xrightarrow{c'}$ $\langle\{q\}, \mathtt{a}, u'\rangle \in \Delta_{k+1}$ is produced from $f(l_1, \dots, l_n) \to g(r_1, \dots, r_m) \in R$ and $f(\langle P_1, \mathtt{x}_1, u_1\rangle, \dots, \langle P_m, \mathtt{x}_m, u_m\rangle) \xhookrightarrow{c} \langle\{q\}, \mathtt{a}, u\rangle \in \Delta_{k-1}$. If we have the equality or the disequality between i and j such that $l_i, l_j \in X$, l_i is the erasing variable, and there exists j' such that $l_j = r_{j'}$, then the equality or the disequality is not preserved to the produced rule.

The following lemma shows the soundness of $\mathrm{P}^{\mathrm{s}}_{\mathrm{csin}}$.

Lemma 104 *If R is a shallow and non-erasing TRS, then $\mathcal{L}(\mathcal{A}_*) \subseteq \xhookrightarrow[R]{}{}^{\mu}_{\mathrm{in}}[\mathcal{L}(\mathcal{A})]$.*

Proof: Let $t \xrightarrow[\Delta_*]{*} \langle P, \mathtt{x}, u'\rangle \in Q^f_*$ then we have $s \xhookrightarrow[R]{*}{}^{\mu}_{\mathrm{in}} t$ and $s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{x}, u\rangle \in Q^f_*$ for some $q \in P$ from Lemma 103. Since $s \xrightarrow[\Delta_0]{*} \langle\{q\}, \mathtt{i}, u\rangle$ from Proposition 86, we have $s \xrightarrow[\Delta]{*} q \in Q^f$ from Proposition 84. $\Box$

Finally we obtain the following theorems from Lemma 102 and 104.

Theorem 105 *Reachability is decidable for a shallow and non-erasing TRSs with respect to the context-sensitive innermost reduction.*

However, in general, we cannot always construct a TACBB recognizing the set of terms reachable from a regular set of terms by the context-sensitive innermost reduction of a shallow and non-erasing TRS, while we can construct a TACBB in the case of the ordinary reduction [10].

Theorem 106 *There exists a regular set L and a shallow and non-erasing TRS R and a replacement map μ such that $\overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}}[L]$ cannot be recognized by any TACBB.*

Proof: Let

$$\begin{aligned}
&\mathrm{ar}(a) = 0, \mathrm{ar}(f) = \mathrm{ar}(h) = \mathrm{ar}(i) = 1, \mathrm{ar}(g) = 2,\\
&L = \{f(t) \mid t \in \mathcal{T}(\{h, a\})\},\\
&R = \{f(x) \to g(x, x), h(x) \to i(x)\}, \text{and}\\
&\mu(f) = \mu(h) = \mu(i) = \emptyset, \mu(g) = \{1, 2\}.
\end{aligned}$$

Then, we have $\overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}}[L] \cap \mathcal{T}(g, h, i, a) = \{g(t_1, t_2) \mid t_1, t_2 \in \mathcal{T}(\{h, i, a\}), |t_1| = |t_2|\}$. Since $\overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}}[L] \cap \mathcal{T}(g, h, i, a)$ cannot be recognized by any TACBB, $\mathcal{T}(g, h, i, a)$ is regular, and TACBB is closed under intersection, there exists no TACBB that recognizes $\overset{*}{\underset{R}{\hookrightarrow}}{}^{\mu}_{\mathrm{in}}[L]$. $\square$

Chapter 6

Decidability and Undecidability for Right-Linear and Left-Shallow TRSs

In this chapter, we show that right-linear, left-shallow, and non-erasing properties define a class in which the reachability problem is decidable for the ordinary reduction and the innermost reduction, and undecidable for the outermost reduction and the context-sensitive reduction. It has been known that the reachability problem is undecidable for right-linear and left-shallow TRSs since it was shown in [18] and [35]. However, for the subclass with non-erasingness, the reachability problem becomes decidable. Moreover, we show that left-linear and right-shallow, which is the almost symmetric class of right-linear, left-shallow, and non-erasing, is a class such that reachability is decidable for the ordinary reduction, and is undecidable for the context-sensitive reduction and the innermost reduction. Decidability and undecidability of the reachability problem for the class aimed in this chapter is arranged in Table 1 in Chapter 1.

We show the results on decidability in this paper by presenting algorithms that construct a TA recognizing the set of terms reachable to or reachable from a given term by the reduction of a given TRS. Results for undecidability is proved by constructing a TRS in which reachability problem corresponds to an instance of the post correspondence problem (PCP), which is well-known as a undecidable problem.

6.1 On Right-Linear and Left-Shallow TRSs

In this section, we show that for right-linear, left-shallow, and non-erasing TRSs, reachability is decidable with respect to the ordinary reduction and the innermost reduction, and undecidable for the context-sensitive reduction and the outermost reduction. Remark that it was shown that reachability for right-linear and left-shallow TRSs are undecidable in [18] and [35], but erasing variables are allowed to occur in the class.

Undecidability properties in this paper are shown by reducing post correspondence problem (PCP), which is well-known as undecidable problem, to a reachability problem.

Definition 107 ((PCP) [29]) An instance of PCP is a set of pairs of strings $\{(u_i, v_i) \in \Sigma^* \times \Sigma^* \mid 1 \leq i \leq n\}$ where Σ is the set of alphabets.

PCP is the problem whether there exists a string w such that $w = u_{k_1} u_{k_2} \cdots u_{k_l} = v_{k_1} v_{k_2} \cdots v_{k_l}$ where $k_j \in \{1, \ldots, n\}$ for $1 \leq j \leq l$.

Theorem 108 ([29]) *PCP is undecidable.*

For readability, we sometimes denote a context $a_1(a_2(\cdots(a_n(\Box))))$ as $a_1 a_2 \cdots a_n(\Box)$ constructed from a string $a_1 a_2 \cdots a_n$.

6.1.1 On the Ordinary Reduction

In the case of the ordinary reduction, it was shown that reachability is undecidable for right-linear and left-shallow TRSs in [18] and [35] as the following Theorem.

Theorem 109 ([18, 35]) *Reachability is undecidable for the ordinary reduction of right-linear and left-shallow TRSs.*

Proof: We show the proof in [18] here. From a PCP $P = \{(u_i, v_i) \mid 1 \leq i \leq n\}$, we construct $R = \bigcup_{1 \leq i \leq n} \{\varepsilon_1 \to f(u_i(\varepsilon_2), v_i(\varepsilon_2))\} \cup \bigcup_{1 \leq i \leq n} \{f(x, y) \to$

$f(u_i(x), v_i(y))\} \cup \{f(x,x) \to \mathtt{e}\}$ for all $1 \leq i \leq n$ where ε_1, ε_2, f, and $\mathtt{e}$ are not the alphabets used in P.

Then w is a solution of P iff $\varepsilon_1 \xrightarrow[R]{*} f(w(\varepsilon_2), w(\varepsilon_2)) \xrightarrow[R]{} \mathtt{e}$. □

However, the reachability problem become decidable by restricting erased variables. In the proof of Theorem 109, the rewrite rule $f(x,x) \to \mathtt{e}$ in R has the erased variable x. If the occurrence of non-erasing variable is restricted, then we cannot construct a TRS in which reachability problem can be reduced from PCP.

In the following, we five an algorithm that constructs a TA $\mathcal{A}_*$ that recognizes the set of terms reachable to an input term similarly in the previous chapters. The algorithm is almost same as the algorithm in [18]. The only deference is the inputs of the algorithm.

The algorithm in [18] is equivalent to $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ except for augmenting parameter to each state of output automata and the language recognized by output automata that is the set of terms reachable to an input term, while output automata of $\mathrm{P}^{\mathrm{lrs}}_{\mathrm{cs}}$ recognizes the set of terms reachable from an input term. In the algorithm $\mathrm{P}^{\mathrm{rlls}}$ in this paper, a TA $\mathcal{A}$ produced from an input term has a condition that each state q^s has a tag s such that $\mathcal{L}(\mathcal{A}, q^s) = \{s\}$ for a state q^s. This condition is the key to the proof of decidability of reachability for the ordinary reduction of right-linear, left-shallow, and non-erasing TRSs. The algorithm $\mathrm{P}^{\mathrm{rlls}}$ allows EV-TRSs, which are the TRSs with the rewrite rules such that variables occur only in a right-hand-side of a rewrite rule or a left-hand-side of a rewrite rule is a variable, as input. This makes it easy to prove decidability for left-linear and right-shallow TRSs (at the next section).

Algorithm $\mathrm{P}^{\mathrm{rlls}}$

Input A term t and a right-linear, left-shallow, and non-erasing EV-TRS R.

Output TA $\mathcal{A}_* = \langle Q, Q^f, \Delta_* \rangle$ such that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}[\{t\}]$.

Step 1 (initialize) 1. Prepare a TA $\mathcal{A} = \langle F, Q, Q^f, \Delta_0 \rangle$ such that $\mathcal{L}(\mathcal{A}) = \{t\}$. $Q = \{q^{s'} \mid s' \trianglelefteq s, s \in \{t\} \cup \mathrm{LS}(R)\}$, $Q^f = \{q^t\}$, and $\Delta_0 = \{f(q^{t_1}, \ldots, q^{t_n}) \to q^{f(t_1,\ldots,t_n)} \mid f(t_1, \ldots, t_n) \trianglelefteq s, s \in \{t\} \cup \mathrm{LS}(R)\}$ where

LS(R) is the set of ground proper subterms in right-hand-sides of the rules in EV-TRS R.

2. Let $k := 0$.

Step 2 Let Δ_{k+1} be the set of transition rules produced by augmenting transition rules of Δ_k by the following inference rules:

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

if there exists $\theta : X \to Q$ such that $r_i\theta \xrightarrow[\Delta_k]{*} q_i$ for all $1 \le i \le m$. Each q'_j is determined as

- $q'_j = \begin{cases} l_j\theta & \cdots\ l_j \in X \\ q^{l_j} & \cdots\ l_j \notin X \end{cases}$

for all $1 \le j \le n$,

$$\frac{f(l_1, ..., l_n) \to x \in R,\ q \in Q}{f(q'_1, \ldots, q'_n) \to q \in \Delta_{k+1}}$$

Each q'_j is determined as

- $q'_j = \begin{cases} q & \cdots\ l_j \in X \\ q^{l_j} & \cdots\ l_j \notin X \end{cases}$

for all $1 \le j \le n$, and

$$\frac{x \to g(r_1, \ldots, r_m) \in R,\ g(q_1, \ldots q_m) \to q \in \Delta_k}{q' \to q \in \Delta_{k+1}}$$

if there exists $\theta : X \to Q$ such that $r_i\theta \xrightarrow[\Delta_k]{*} q_i$ for all $1 \le i \le m$. q' is $x\theta$.

Note that each q'_j in the first and the second inference rules is well-defined since R is non-erasing.

Step 3 If $\Delta_{k+1} = \Delta_k$ then stop and set $\Delta_* = \Delta_k$. Otherwise, $k := k + 1$, and go to step 2.

Here, we show an example that shows how $\mathrm{P}^{\mathrm{rlls}}$ works.

Example 110 We input the term $g(b)$ and the TRS $R = \{a \to b, f(x,x) \to g(x)\}$. P$^{\mathrm{rlls}}$ output the TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}[\{g(b)\}] = \{g(b), g(a), f(b,b), f(a,b), f(b,a), f(a,a)\}$.

In Step 1, we obtain the TA $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ as follows:

$$\begin{aligned} Q &= \{q^t \mid t \trianglelefteq g(b)\}, \\ Q^f &= \{q^{g(b)}\} \\ \Delta &= \left\{b \to q^b, g(q^b) \to q^{g(b)}\right\} \end{aligned}$$

and set $k = 0$.

In step 2, at $k = 0$, we have Δ_1 as follows:

$$\Delta_1 = \Delta_0 \cup \left\{f(q^b, q^b) \to q^{g(b)}, a \to q^b\right\}.$$

At $k = 1$, this step stop and we have $\Delta_* = \Delta_1$.

Finally, TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_*\rangle$ holds that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}[\{g(b)\}]$. □

If TA $\mathcal{A}$ produced at 1 of Step 1 has no condition, there is the case that P$^{\mathrm{rlls}}$ does not output the TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}[\mathcal{L}(\mathcal{A})]$. It is because if $\mathcal{A}$ in the example 110 has the extra transition rule $c \to q^a$, then the terms that is not reachable to $g(h(a))$ (e.g. $f(c,b)$) is accepted by $\mathcal{A}_*$.

The algorithm eventually terminates at some k similarly to the other algorithms in this paper, and apparently $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$.

The following proposition holds for constructed TAs by the above algorithm.

Proposition 111 *Let $s \xrightarrow[\Delta_0]{*} q$ and $t \xrightarrow[\Delta_0]{*} q'$. Then, $q = q'$ implies $s = t$.*

Proof: From the construction of Δ_0. □

Following Lemma 113 shows the completeness of P$^{\mathrm{rlls}}$ and Lemma 112 is the key to prove Lemma 113.

Lemma 112 *Let R be a right-linear, left-shallow, and non-erasing EV-TRS. Then, $t \xrightarrow[\Delta_*]{*} q$ and $s \xrightarrow[R]{} t$ implies $s \xrightarrow[\Delta_*]{*} q$.*

Proof: Assume that s is rewritten to t by the rewrite rule $l \to r \in R$. Then, s and t can be represented as $C[l\sigma]$ and $C[r\sigma]$, respectively.

1. Assume that the rewrite rule $l \to r$ is represented as $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m)$. In this case, the transition $t = C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_*]{*} q$ is represented as $C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_*]{*} C[g(q_1, \ldots, q_m)] \xrightarrow[\Delta_*]{} C[q'] \xrightarrow[\Delta_*]{*} q$. Here, from right-linearity of R, we have the substitution $\theta : X \to Q$ such that $r_i\sigma \xrightarrow[\Delta_*]{*} r_i\theta \xrightarrow[\Delta_*]{*} q_i$. Therefore, from the first inference rule of the algorithm, we have the transition rule $f(q'_1, \ldots, q'_n) \to q'$ where $q'_j = q^{l_j}$ for $l_j \notin X$ and $q'_j = l_j\theta$ for $l_j \in X$. Note that every q'_j is well-defined since R is non-erasing. Thus, we have $l_j\sigma \xrightarrow[\Delta_*]{*} q'_j$ for all $1 \leq j \leq n$ from Proposition 111 and hence $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[\Delta_*]{*} C[f(q'_1, \ldots, q'_n)] \xrightarrow[\Delta_*]{} C[q'] \xrightarrow[\Delta_*]{*} q$.

2. In the case where the rewrite rule $l \to r \in R$ is represented as $f(l_1, \ldots, l_n) \to x$, the proof is similar to the previous case.

3. In the case where the rewrite rule $l \to r$ is represented as $x \to g(r_1, \ldots, r_m)$, the transition $t = C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_*]{*} q$ is represented as $C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_*]{*} C[g(q_1, \ldots, q_m)] \xrightarrow[\Delta_*]{} C[q'] \xrightarrow[\Delta_*]{*} q$ and we have the substitution $\theta : X \to Q$ such that $r_i\sigma \xrightarrow[\Delta_*]{*} r_i\theta \xrightarrow[\Delta_*]{*} q_i$. From the third inference rule of the algorithm, we have the transition rule $q'' \to q'$ where $q'' = q_j$ where $l_j = x$. Note that q'' is well-defined since R is non-erasing. Thus, we have $x\sigma \xrightarrow[\Delta_*]{*} q''$ and hence $C[x\sigma] \xrightarrow[\Delta_*]{*} C[q''] \xrightarrow[\Delta_*]{} C[q'] \xrightarrow[\Delta_*]{*} q$.

4. If $l \to r$ is represented as $x \to y$, then we have $x = y$ since R is non-erasing and hence $s = t$. □

Lemma 113 *Let R be a right-linear, left-shallow, and non-erasing EV-TRS. Then, $\xleftarrow[R]{}[\{t\}] \subseteq \mathcal{L}(\mathcal{A}_*)$.*

Proof: Since $t \xrightarrow[\Delta_0]{*} q^f \in Q^f$ and $\Delta_0 \subseteq \Delta_*$, we have $t \xrightarrow[\Delta_*]{*} q^f$. If $s \xrightarrow[R]{*} t$, we have $s \xrightarrow[\Delta_*]{*} q^f$ by applying Lemma 112 repeatedly. □

Following Lemma 115 shows the soundness of $\mathrm{P}^{\mathrm{rlls}}$ and Lemma 114 is the key to prove Lemma 115.

Lemma 114 *Let R be a right-linear and left-shallow non-erasing EV-TRS. Then, $s \xrightarrow[\Delta_k]{*} q$ for some k implies that there exists t such that $s \xrightarrow[R]{*} t$ and $t \xrightarrow[\Delta_0]{*} q$.*

Proof: We prove this lemma by induction on k.

1. If $k = 0$, then this lemma holds trivially.

2. If $k > 0$, we show this lemma by induction on the number of the transition rules in $\Delta_k \backslash \Delta_{k-1}$ that occur in $s \xrightarrow[\Delta_k]{*} q$

   If no transition rule in $\Delta_k \backslash \Delta_{k-1}$ occurs in $s \xrightarrow[\Delta_k]{*} q$, then we also have $s \xrightarrow[\Delta_{k-1}]{*} q$ and hence this lemma holds by the induction hypothesis.

   In the following, consider the case where a transition rule in $\Delta_k \backslash \Delta_{k-1}$ occurs in $s \xrightarrow[\Delta_k]{*} q$.

   (a) Consider the case where $s \xrightarrow[\Delta_k]{*} q$ is represented as $s = C[f(s_1, \ldots, s_n)] \xrightarrow[\Delta_k]{*} C[f(q'_1, \ldots, q'_n)] \xrightarrow[\Delta_k \backslash \Delta_{k-1}]{} C[q'] \xrightarrow[\Delta_{k-1}]{*} q$. Then, from the induction hypothesis, we have t_j such that $s_j \xrightarrow[R]{*} t_j$ and $t_j \xrightarrow[\Delta_0]{*} q'_j$ for all $1 \leq j \leq n$. Since the transition rule $f(q'_1, \ldots, q'_n) \to q' \in \Delta_k \backslash \Delta_{k-1}$ is produced by the first inference rule or second inference rule of the algorithm, we should consider the two cases. Here, we show the proof in the former case, since the proof in the latter case is similar and easier.

   If the transition rule is produced by the first inference rule, then there exist a rewrite rule $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m) \in R$, a transition rule $g(q_1, \ldots, q_m) \to q' \in \Delta_{k-1}$, and a substitution $\theta : X \to Q$ such that $r_i\theta \xrightarrow[\Delta_{k-1}]{*} q_i$ for $1 \leq i \leq m$. Moreover, q'_j is q^{l_j} if $l_j \notin X$, otherwise $l_j\theta$. Note that there is no $l_j \in X$ that is not a domain of θ since the rewrite rule $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m)$ has no erasing variable.

   For j such that $l_j \notin X$, we have $t_j = l_j$ from shallowness of R and the construction of Δ_0. For j such that $l_j \in X$, we have $q'_j = q'_{j'} = l_j\theta$ for all j' such that $l_{j'} = l_j$ and hence we have $t_j = t_{j'}$ from Proposition 111. Thus, we have the substitution $\sigma : X \to \mathcal{T}(F)$ such that $r_i\sigma \xrightarrow[\Delta_0]{*} q_i$ by taking $l_j\sigma$ as t_j for $l_j \in \mathrm{Var}(g(r_1, \ldots, r_m))$ and $x\sigma \xrightarrow[\Delta_0]{*} x\theta$ for $x \in \mathrm{Var}(g(r_1, \ldots, r_m)) \backslash \mathrm{Var}(f(l_1, \ldots, l_n))$.

   Finally, we have $C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_0]{*} C[g(r_1, \ldots, r_m)\theta] \xrightarrow[\Delta_{k-1}]{*} C[g(q_1, \ldots, q_m)] \xrightarrow[\Delta_{k-1}]{} C[q'] \xrightarrow[\Delta_{k-1}]{*} q$ and hence there exists the term t such that $t \xrightarrow[R]{*} C[g(r_1,\ldots,r_m)\sigma] \xrightarrow[R]{} C[f(l_1,\ldots,l_n)\sigma] \xrightarrow[R]{*} C[f(l_1,\ldots,l_n)] = s$ and $t \xrightarrow[\Delta_0]{*} q$ from the induction hypothesis.

(b) Assume that $s \xrightarrow[\Delta_k]{*} q$ is represented as $s = C[s'] \xrightarrow[\Delta_k]{*} C[q''] \xrightarrow[\Delta_k\backslash\Delta_{k-1}]{} C[q'] \xrightarrow[\Delta_{k-1}]{*} q$. Since the transition rule $q'' \to q' \in \Delta_k\backslash\Delta_{k-1}$ is produced by the third inference rule, then there exist a rewrite rule $x \to g(r_1,\ldots,r_n) \in R$, a transition rule $q'' \to q' \in \Delta_{k-1}$, and a substitution $\theta : X \to Q$ such that $r_i\theta \xrightarrow[\Delta_{k-1}]{*} q_i$. Moreover, q'' is $l_j\theta$ where $l_j = x$.

Here, let the substitution $\sigma : X \to \mathcal{T}(F)$ such that $x\sigma = s'$ and we have $x\sigma \xrightarrow[\Delta_0]{*} x\theta$. Then, we have $C[x\sigma] \xrightarrow[\Delta_0]{*} C[x\theta] = C[q''] \xrightarrow[\Delta_{k-1}]{} C[q'] \xrightarrow[\Delta_{k-1}]{*} q$ and hence there exists the term t such that $t \xrightarrow[R]{*} C[x\sigma] \xrightarrow[R]{} C[f(l_1,\ldots,l_n)\sigma] \xrightarrow[R]{*} C[f(l_1,\ldots,l_n)] = s$ and $t \xrightarrow[\Delta_0]{*} q$ from the induction hypothesis. $\square$

Lemma 115 *Let R be a right-linear, left-shallow, and non-erasing TRS. Then, $\xleftarrow[R]{}[\{t\}] \supseteq \mathcal{L}(\mathcal{A}_*)$.*

Proof: Let $s \in \mathcal{L}(\mathcal{A}_*)$, then we have the transition $s \xrightarrow[\Delta_*]{*} q^f \in Q^f$, and hence there exists s' such that $s \xrightarrow[R]{*} s'$ and $s' \xrightarrow[\Delta_0]{*} q^f$ from Lemma 114. Since $\mathcal{A}$ accepts only t from the construction, we have $t = s'$ and hence this lemma holds. $\square$

The following theorem is proved by Lemma 113, 115 and Theorem 2.

Theorem 116 *Reachability is decidable for left-linear, right-shallow, and non-erasing EV-TRSs with respect to the ordinary reduction.*

Following corollary is obtained from Theorem 116.

Corollary 117 *Reachability is decidable for left-linear, right-shallow, and non-erasing TRSs with respect to the ordinary reduction.*

6.1.2 On the Innermost Reduction

In the case of the innermost reduction, similarly to the ordinary reduction, reachability for right-linear and left-shallow TRSs is undecidable but decidable for right-linear, left-shallow, and non-erasing TRSs.

The following theorem shows undecidability.

Theorem 118 *Reachability is undecidable for right-linear and left-shallow TRSs with respect to the innermost reduction.*

Proof: For the TRS R in the proof of Theorem 109, we have $\varepsilon_1 \xrightarrow[R]{*}_{\text{in}} \mathtt{e}$ iff PCP has a solution. $\square$

Next, we show an algorithm $\mathrm{P}_{\text{in}}^{\text{rlls}}$ that constructs a TA $\mathcal{A}_*$ from a right-linear, left-shallow, and non-erasing TRS R and a term t that accepts the set of terms reachable to t by the innermost reductions of R.

For an innermost reduction $C[l\sigma] \xrightarrow[R]{}_{\text{in}} C[r\sigma]$, $x\sigma$ for every $x \in \text{Var}(l)$ and each ground proper subterm of l are normal forms. Therefore, first we remove the rewrite rules in which the left-hand-side has redexes in its ground proper subterm, and we add transition rules for substitutions $\theta : X \to Q$ such that each $x\theta$ accepts a normal form for $x \in \text{Var}(l)$.

Remark In the case of context-sensitive innermost in the previous chapters, we introduced the idea that we construct a deterministic complete reduced TA and add its states as new parameter of the states of output TA of the algorithms. However, in $\mathrm{P}_{\text{in}}^{\text{rlls}}$, we do not add new parameter to each state of an output TA. It is because, for a TRS R and a term t, normal forms of R that occurs in $\xleftarrow[R]{}_{\text{in}}[\{t\}]$ are finite. We describe the reason here. In an innermost rewriting $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[R]{}_{\text{in}} C[r\sigma]$ by a TRS R, normal forms that occurs in $C[f(l_1, \ldots, l_n)\sigma]$ and does not occur in $C[r\sigma]$ must be some l_i's. Therefore, normal forms occurs in $\xleftarrow[R]{}_{\text{in}}[\{t\}]$ are l_i's such that $l_i \notin X$ or subterms in t.

Thus, it is sufficient that we produce the states q^s's that correspond to a term s and produce the set $Q_{\text{NF}} = \{q^s \mid s \in NF_R\}$. Then, we can check the state q^s accepts a normal form or not.

Algorithm $\mathbf{P}_{\text{in}}^{\text{rlls}}$

Input A term t and a right-linear, left-shallow, and non-erasing TRS R.

Output A TA $\mathcal{A}_* = \langle F, Q, Q^f, \Delta_* \rangle$ such that $\mathcal{L}(\mathcal{A}_*) = \xrightarrow[R]{}_{\text{in}}[\{t\}]$.

Step 1 (initialize)

1. Let $R' = \{f(l_1, \ldots, l_n) \to r \mid f(l_1, \ldots, l_n) \to r \in R, l_i \in NF_R \vee l_i \in X \text{ for all } 1 \leq i \leq n\}$
2. Prepare a TA $\mathcal{A} = \langle F, Q, Q^f, \Delta_0 \rangle$ such that $\mathcal{L}(\mathcal{A}) = \{t\}$. $Q = \{q^{s'} \mid s' \trianglelefteq s, s \in \{t\} \cup \mathrm{LS}(R')\}$, $Q^f = \{q^t\}$, and $\Delta_0 = \{f(q^{t_1}, \ldots, q^{t_n}) \to q^{f(t_1,\ldots,t_n)} \mid f(t_1, \ldots, t_n) \trianglelefteq s, s \in \{t\} \cup \mathrm{LS}(R')\}$ where $\mathrm{LS}(R')$ is the set of ground proper subterms in right-hand-sides of the rules in R'.
3. Construct the set of states $Q_{\mathrm{NF}} = \{q^s \mid s \in NF_{R'}, q^s \in Q\}$.
4. Let $k := 0$.

Step 2 Let Δ_{k+1} be the set of transition rules produced by augmenting transition rules of Δ_k by the following inference rules:

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

if there exists $\theta : X \to Q$ such that $r_i\theta \xrightarrow[\Delta_k]{*} q_i$ for all $1 \leq i \leq m$ and $x\theta \in Q_{\mathrm{NF}}$ for all $x \in \mathrm{Var}(g(r_1, \ldots, r_m))$. Each q'_j is determined as

- $q'_j = \begin{cases} l_j\theta & \cdots\ l_j \in X \\ q^{l_j} & \cdots\ l_j \notin X \end{cases}$

for all $1 \leq j \leq n$, and

$$\frac{f(l_1, ..., l_n) \to x \in R',\ q \in Q_{\mathrm{NF}}}{f(q'_1, \ldots, q'_n) \to q \in \Delta_{k+1}}$$

Each q'_j is determined as

- $q'_j = \begin{cases} q & \cdots\ l_j \in X \\ q^{l_j} & \cdots\ l_j \notin X \end{cases}$

Note that each q'_j is well-defined in the both cases since R is non-erasing.

Step 3 If $\Delta_{k+1} = \Delta_k$ then stop and set $\Delta_* = \Delta_k$. Otherwise, $k := k + 1$, and go to step 2.

Here, we show two examples that shows how $\mathrm{P}^{\mathrm{rlls}}_{\mathrm{in}}$ works.

Example 119 We input the term $g(b)$ and the TRS $R = \{a \to b, f(x,x) \to g(x)\}$. $\mathrm{P}^{\mathrm{rlls}}$ output the TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}_{\mathrm{in}}[\{g(b)\}] = \{g(b), g(a), f(b,b), f(a,b), f(b,a), f(a,a)\}$.

In Step 1, we obtain the TA $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ as follows:

$$\begin{aligned}Q &= \{q^t \mid t \trianglelefteq g(b)\},\\ Q^f &= \{q^{g(b)}\}\\ \Delta &= \left\{b \to q^b, g(q^b) \to q^{g(b)}\right\}\end{aligned}$$

and set $k = 0$.

In step 2, at $k = 0$, we have Δ_1 as follows:

$$\Delta_1 = \Delta_0 \cup \left\{f(q^b, q^b) \to q^{g(b)}, a \to q^b\right\}.$$

At $k = 1$, this step stop and we have $\Delta_* = \Delta_1$.

Finally, TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_*\rangle$ holds that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}_{\mathrm{in}}[\{g(b)\}]$. □

Example 120 We input the term $g(a)$ and the TRS $R = \{a \to b, f(x,x) \to g(x)\}$. $\mathrm{P}^{\mathrm{rlls}}$ output the TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}_{\mathrm{in}}[\{g(a)\}] = \{g(a)\}$.

In Step 1, we obtain the TA similarly to Example 119. $\mathcal{A} = \langle Q, Q^f, \Delta\rangle$ as follows:

In step 2, at $k = 0$, there is no transition rule added to Δ_1 since a is not a normal form and hence $q^a \notin Q_{\mathrm{NF}}$. Thus, this step stop at $k = 0$.

Finally, TA $\mathcal{A}_* = \langle Q_*, Q^f_*, \Delta_*\rangle$ holds that $\mathcal{L}(\mathcal{A}_*) = \xleftarrow[R]{}_{\mathrm{in}}[\{g(a)\}]$. □

The output TA in the Example 119 is same as the one in the case of the ordinary reduction from Example 110. However, the output TA in the Example 120 is not same as the one in the case of the ordinary reduction since $\xleftarrow[R]{}[\{g(a)\}]$ for the TRS R in Example 120 is $\{g(a), f(a,a)\} \neq \xleftarrow[R]{}_{\mathrm{in}}[\{g(a)\}]$ and $\mathrm{P}^{\mathrm{rlls}}$ output the TA whose transition rules are $\Delta \cup \{f(q^a, q^a) \to q^{g(a)}\}$ for R and $g(a)$ as inputs.

The algorithm eventually terminates at some k similarly in the other algorithms in this paper, apparently $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$.

For the constructed TA $\mathcal{A}_*$, the following proposition and lemmas holds. These are used to prove the completeness and the soundness of $\mathrm{P}^{\mathrm{rlls}}_{\mathrm{in}}$.

Proposition 121 *Let* $s \xrightarrow[\Delta_0]{*} q$ *and* $t \xrightarrow[\Delta_0]{*} q'$. *Then,* $q = q'$ *implies* $s = t$.

Proof: From the construction of Δ_0 □

Lemma 122 $NF_R = NF_{R'}$.

Proof: Since $R \supseteq R'$, if a term s is not reducible by R then it is not reducible by R', either. Thus, $NF_R \subseteq NF_{R'}$ holds.

Let $s \in NF_{R'}$. We show that $s \in NF_R$ by induction on $|s|$. Let $s = f(s_1, \ldots, s_n)$ for $n \geq 0$. Since $s \in NF_{R'}$, s cannot be reduced by R'. Here, we suppose that s can be reduced by R. Therefore, there exists a $l \to r \in R \backslash R'$ such that $s = C[l\sigma]_p$ for some substitution σ.

If $p = \varepsilon$, then s can be represented as $f(l_1, \ldots, l_n)\sigma$ where $f(l_1, \ldots, l_n) \to r \in R \backslash R'$. From the construction of R', some l_i for $l_i \notin X$ is not a normal form of R. Therefore, s is not a normal form and hence this is a contradiction.

If $p \neq \varepsilon$, then some s_i can be reduced by $R \backslash R'$. However, $s_i \in NF_{R'}$ and hence $s_i \in NF_R$ from the induction hypothesis. This is a contradiction.

Thus, $s \in NF_{R'}$ implies $s \in NF_R$ and hence $NF_R \supseteq NF_{R'}$. □

Proposition 123 $s \xrightarrow[R]{*}_{\text{in}} t$ *iff* $s \xrightarrow[R']{*}_{\text{in}} t$.

Proof:

(if part) Assume that $s \xrightarrow[R']{}_{\text{in}} t$ is represented as $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[R']{}_{\text{in}} C[r\sigma]$ where $f(l_1, \ldots, l_n) \to r \in R'$. Then we have $l_i\sigma \in NF_{R'}$ and hence $l_i\sigma \in NF_R$ from Lemma 122 for $1 \leq i \leq n$. Since the rewrite rule $f(l_1, \ldots, l_n) \to r$ is also in R, we have $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[R]{}_{\text{in}} C[r\sigma]$.

(only if part) Assume that $s \xrightarrow[R]{}_{\text{in}} t$ is represented as $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[R]{}_{\text{in}} C[r\sigma]$. Then, since this reduction is innermost and the rewrite rule $f(l_1, \ldots, l_n) \to r$ is left-shallow, all l_i's are variable or normal form. Thus, $f(l_1, \ldots, l_n) \to r$ is also in R' and hence we have $s \xrightarrow[R']{}_{\text{in}} t$. By repeating this process, only if part is proved. □

Lemma 124 *If* $t \xrightarrow[\Delta_k]{*} q$, *then,* $k = 0 \wedge q \in Q_{\text{NF}}$ *iff* $t \in NF_R$.

Proof: If $k = 0$, then $q = q^t$ from the construction of $\mathcal{A}$. Thus, from the construction of Q_{NF}, $q \in Q_{\mathrm{NF}}$ iff $t \in NF_R$.

If $k > 0$, then $t \xrightarrow[\Delta_k]{*} q$ is represented as $t = C[f(t_1, \ldots, t_n)] \xrightarrow[\Delta_0]{*} C[f(q'_1, \ldots, q'_n)] \xrightarrow[\Delta_k \setminus \Delta_0]{} C[q'] \xrightarrow[\Delta_k]{*} q$ without loss of generality.

From the algorithm, we have the rewrite rule $f(l_1, \ldots, l_n) \to r \in R$ where $l_j \notin X$ implies $l_j \in NF_R \wedge q'_j = q^{l_j}$ and $l_j \in X$ implies $q'_j \in Q_{\mathrm{NF}}$.

Since we have $t_j \xrightarrow[\Delta_0]{*} q'_j$, $t_j = l_j$ holds for $l_j \notin X$, and $t_j \in NF_R$ and $t_j = t_{j'}$ for all j' such that $l_{j'} = l_j \in X$ from the construction of $\mathcal{A}$.

Thus, we have a substitution $\sigma : X \to \mathcal{T}(F)$ such that $f(t_1, \ldots, t_n) = f(l_1, \ldots, l_n)\sigma$ and each $l_j\sigma$ is a normal form. Hence, t is not a normal form.

□

Following Lemma 126 shows the completeness of $\mathrm{P}_{\mathrm{in}}^{\mathrm{rlls}}$, and Lemma 125 is the key to prove Lemma 126.

Lemma 125 *If R is a right-linear, left-shallow, and non-erasing TRS, then, $t \xrightarrow[\Delta_*]{*} q$ and $s \xrightarrow[R]{}_{\mathrm{in}} t$ implies $s \xrightarrow[\Delta_*]{*} q$.*

Proof: From Proposition 123, we have $s \xrightarrow[R']{}_{\mathrm{in}} t$. Assume that t is rewritten from s by an innermost reduction of the rewrite rule $l \to r \in R'$. Then, s and t is represented as $C[l\sigma]$ and $C[r\sigma]$, respectively.

1. Consider the case that the rewrite rule $l \to r$ is represented as $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_m)$. In this case, the transition $t = C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_*]{*} q$ is represented as $C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_*]{*} C[g(q_1, \ldots, q_m)] \xrightarrow[\Delta_*]{} C[q'] \xrightarrow[\Delta_*]{*} q$. Note that $x\sigma \in NF_{R'}$ for all $x \in \mathrm{Var}(g(r_1, \ldots, r_m))$. Here, from right-linearity of R' and Lemma 124, we have the substitution $\theta : X \to Q$ such that $g(r_1, \ldots, r_m)\sigma \xrightarrow[\Delta_0]{*} g(r_1, \ldots, r_m)\theta \xrightarrow[\Delta_*]{*} g(q_1, \ldots, q_m)$ where $x\theta \in Q_{\mathrm{NF}}$ for all $x \in \mathrm{Var}(g(r_1, \ldots, r_m))$. Therefore, from the first inference rule of the algorithm, we have the transition rule $f(q'_1, \ldots, q'_n) \to q'$, and for all $1 \leq j \leq n$, we have $q'_j = q^{l_j}$ for $l_j \notin X$ and $q'_j = l_j\theta$ for $l_j \in X$ since R' is non-erasing. Thus, we have $l_j\sigma \xrightarrow[\Delta_*]{*} q_j$ for all $1 \leq j \leq n$ from left-shallowness of R and hence $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[\Delta_*]{*} C[f(q'_1, \ldots, q'_n)] \xrightarrow[\Delta_*]{} C[q'] \xrightarrow[\Delta_*]{*} q$.

2. Otherwise, the rewrite rule $l \to r \in R'$ is represented as $f(l_1, \ldots, l_n) \to x$. In this case, the transition $t = C[x\sigma] \xrightarrow[\Delta_*]{*} q$ is represented as $C[x\sigma] \xrightarrow[\Delta_*]{*} C[q'] \xrightarrow[\Delta_*]{*} q$. Note that $x\sigma \in NF_R$. Here, let the substitution $\theta = \{x \mapsto q'\}$ and from the second inference rule of the algorithm, we have the transition rule $f(q'_1, \ldots, q'_n) \to q'$, and for all $1 \leq j \leq n$, we have $q'_j = q^{l_j}$ for $l_j \notin X$ and $q'_j = q' = l_j\theta$ for $l_j \in X$ since R' is non-erasing. Thus, we have $l_j\sigma \xrightarrow[\Delta_*]{*} q_j$ for all $1 \leq j \leq n$ from left-shallowness of R and hence $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[\Delta_*]{*} C[f(q'_1, \ldots, q'_n)] \xrightarrow[\Delta_*]{} C[q'] \xrightarrow[\Delta_*]{*} q$. □

Lemma 126 *If R is a right-linear, left-shallow, and non-erasing TRS. Then, $\xrightarrow[R]{}_{\mathrm{in}}[\{t\}] \subseteq \mathcal{L}(\mathcal{A}_*)$ where $\mathcal{A}_*$ is constructed from $\mathcal{A}$ that is constructed from $\{t\}$.*

Proof: Since $s \xrightarrow[\Delta_0]{*} q^s \in Q^f$ and $\Delta_0 \subseteq \Delta_*$, we have $s \xrightarrow[\Delta_*]{*} q^s$. If $s \xrightarrow[R]{*}_{\mathrm{in}} t$, we have $t \xrightarrow[\Delta_0]{*} q^s$ by applying Lemma 125 repeatedly. □

Following Lemma 128 shows the completeness of $\mathrm{P}^{\mathrm{rlls}}_{\mathrm{in}}$, and Lemma 127 is the key to prove Lemma 126.

Lemma 127 *If R is a right-linear, left-shallow, and non-erasing TRS, then, $s \xrightarrow[\Delta_k]{*} q$ implies that there exists t such that $s \xrightarrow[R]{*}_{\mathrm{in}} t$ and $t \xrightarrow[\Delta_0]{*} q$.*

Proof: We prove this lemma by induction on k.

1. If $k = 0$, then this lemma holds trivially.

2. Consider the case where $k > 0$. We show this lemma by induction on the number of the transition rules in $\Delta_k \backslash \Delta_{k-1}$ that occurs in $s \xrightarrow[\Delta_k]{*} q$.

   If no transition rule in $\Delta_k \backslash \Delta_{k-1}$ occurs in $s \xrightarrow[\Delta_k]{*} q$, then we also have $s \xrightarrow[\Delta_{k-1}]{*} q$ and hence this lemma holds by the induction hypothesis.

   If a transition rule in $\Delta_k \backslash \Delta_{k-1}$ occurs in $s \xrightarrow[\Delta_k]{*} q$, it is represented as $s = C[f(s_1, \ldots, s_m)] \xrightarrow[\Delta_k]{*} C[f(q'_1, \ldots, q'_n)] \xrightarrow[\Delta_k \backslash \Delta_{k-1}]{} C[q'] \xrightarrow[\Delta_{k-1}]{*} q$ from the inference rules in the algorithm. Then, from the induction hypothesis, we have t_j such that $s_j \xrightarrow[R]{*}_{\mathrm{in}} t_j$ and $t_j \xrightarrow[\Delta_0]{*} q'_j$ for all $1 \leq j \leq n$. Here, since the transition rule $f(q'_1, \ldots, q'_n) \to q' \in \Delta_k \backslash \Delta_{k-1}$ is produced by the inference rule in the algorithm, we have to consider two cases.

(a) If the transition rule is produced by the first inference rule, then there exist the rewrite rule $f(l_1, \ldots, l_n) \to g(r_1, \ldots, r_n) \in R'$, the transition rule $g(q_1, \ldots, q_m) \to q' \in \Delta_{k-1}$, and the substitution $\theta : X \to Q$ such that $r_i\theta \xrightarrow[\Delta_{k-1}]{*} q_i$ and $x\theta \in Q_{\mathrm{NF}}$ for all $x \in \mathrm{Var}(g(r_1, \ldots, r_m))$. Note that each $l_j \notin X$ is a normal form from the construction of R'. Moreover, $q_j = q^{l_j}$ if $l_j \notin X$, otherwise $q_j = l_j\theta$. Note that there is no $l_j \in X$ that is not a domain of θ since R' is non-erasing.

For j such that $l_j \notin X$, we have $t_j = l_j$ from left-shallowness of R' and the construction of $\mathcal{A}$. For j such that $l_j \in X$, we have $q'_j = q'_{j'} = l_j\theta$ for all j' such that $l_{j'} = l_j$ and hence we have $t_j = t_{j'}$ from $q'_j = q'_{j'}$ and the construction of $\mathcal{A}$. Thus, we have a substitution $\sigma : X \to \mathcal{T}(F)$ such that $l_j\sigma \xrightarrow[\Delta_0]{*} q'_j$ by taking t_j as $l_j\sigma$. Moreover, we have $t_j = l_j\sigma \in NF_R$ from Lemma 124 since $t_j \xrightarrow[\Delta_0]{*} q'_j$ and $q'_j = l_j\theta \in Q_{\mathrm{NF}}$. Since there is no $x \in \mathrm{Var}(g(r_1, \ldots, r_m))$ such that x is not a domain of θ from $\mathrm{Var}(f(l_1, \ldots, l_n)) \supseteq \mathrm{Var}(g(r_1, \ldots, r_m))$.

Thus, we have $C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[\Delta_0]{*} C[g(r_1, \ldots, r_m)\theta] \xrightarrow[\Delta_{k-1}]{*} C[g(q_1, \ldots, q_m)] \xrightarrow[\Delta_{k-1}]{} C[q'] \xrightarrow[\Delta_{k-1}]{*} q$ and $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[R']{}_{\mathrm{in}} C[g(r_1, \ldots, r_m)\sigma]$. Hence, $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[R]{}_{\mathrm{in}} C[g(r_1, \ldots, r_m)\sigma]$ from Proposition 123 and there exists the term t such that $s = C[f(s_1, \ldots, s_n)\sigma] \xrightarrow[R]{*}_{\mathrm{in}} C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[R]{}_{\mathrm{in}} C[g(r_1, \ldots, r_m)\sigma] \xrightarrow[R]{*}_{\mathrm{in}} t$ and $t \xrightarrow[\Delta_0]{*} q$ from the induction hypothesis.

(b) If the transition rule is produced by the second inference rule, this lemma holds similarly to the previous case. □

Lemma 128 *If R is a right-linear, left-shallow, and non-erasing TRS, then, $\xrightarrow[R]{}_{\mathrm{in}}[\{t\}] \supseteq \mathcal{L}(\mathcal{A}_*)$ where $\mathcal{A}_*$ is constructed from $\mathcal{A}$ that is constructed from $\{t\}$.*

Proof: Let $s \in \mathcal{L}(\mathcal{A}_*)$, then we have the transition $s \xrightarrow[\Delta_*]{*} q^f \in Q^f$. Hence, there exists s' such that $s \xrightarrow[R]{*} s'$ and $s' \xrightarrow[\Delta_0]{*} q^f$ from Lemma 127. Since $\mathcal{A}$ accepts only t from the construction, we have $s' = t$ and hence this lemma holds. □

The following theorem is proved by Lemma 126 and 128.

Theorem 129 *Reachability is decidable for right-linear, left-shallow, and non-erasing TRSs with respect to the ordinary reduction.*

6.1.3 On the Context-Sensitive Reduction

In the case of the context-sensitive reduction, reachability is undecidable.

Theorem 130 *Reachability is undecidable for right-linear, left-shallow, and non-erasing TRSs with respect to the context-sensitive reduction.*

Proof: Let the PCP $P = \{(u_i, v_i) \in \Sigma^* \times \Sigma^* \mid 1 \ \leq i \leq n\}$ and the TRS R be the one in the proof of Theorem 109. We prepare the TRS $R' = (R \backslash \{f(x,x) \to \mathsf{e}\}) \bigcup_{a \in \Sigma} \{a(\varepsilon_2) \to \varepsilon_2\} \cup \{f(x,x) \to g(x), g(\varepsilon_2) \to \mathsf{e}\}$ and the replacement map μ such that $\mu(h) = \{1\}$ for $h \in \Sigma \cup \{g\}$ and $\mu(f) = \emptyset$.

Then, $\varepsilon_1 \xleftarrow[R']{*}{}^{\mu} f(w(\varepsilon_2), w(\varepsilon_2)) \xhookrightarrow[R']{}{}^{\mu} g(w(\varepsilon_2)) \xleftarrow[R']{*}{}^{\mu} \mathsf{e}$ iff w is a solution of P. □

6.1.4 On the Outermost Reduction

Similarly to the case of the context-sensitive reduction, reachability is undecidable for the outermost reduction.

Theorem 131 *Reachability is undecidable for right-linear, left-shallow, and non-erasing TRSs with respect to the outermost reduction.*

Proof: For PCP P and the TRS R' in the proof of Theorem 130, we have $\varepsilon_1 \xrightarrow[R']{*}{}_{\mathrm{out}} \mathsf{e}$ iff P has a solution. □

6.2 On Left-Linear and Right-Shallow TRSs

For the ordinary reduction and the context-sensitive reduction, their inverse are also the ordinary reduction and the context-sensitive reduction, respectively. Left-linear and right-shallow TRSs are not an exactly but almost a symmetric class of right-linear, left-shallow, and non-erasing TRS, because if a TRS R has collapsing rules or rules that have erasing variable, its inverse is not a TRS from the definition of TRSs. Therefore, decidability or undecidability of reachability for them is not proved directly from the results for right-linear, left-shallow, and non-erasing TRSs.

In this section, we show decidability and undecidability of reachability for left-linear and right-shallow TRSs. We show that decidability and undecidability results are same for the ordinary reduction and the context-sensitive reduction of right-linear, left-shallow, and non-erasing TRSs. In the case of the innermost reduction, we show that reachability is undecidable for left-linear and right-shallow TRSs while it is undecidable for right-linear, left-shallow, and non-erasing TRSs.

6.2.1 On the Ordinary Reduction

Reachability is also decidable for the ordinary reduction of left-linear and right-shallow TRSs. This result is proved as a corollary of Theorem 116.

Corollary 132 *Reachability is decidable for left-linear and right-shallow TRSs with respect to the ordinary reduction.*

Proof: Let R be a left-linear and right-shallow TRSs and $R^{-1} = \{(r,l) \mid (l,r) \in R\}$. Then, R^{-1} is a right-linear, left-shallow, and non-erasing EV-TRS. From Theorem 116, a problem whether $s \xrightarrow[R^{-1}]{*} t$ is decidable for any terms s, t. Since $s \xrightarrow[R^{-1}]{*} t$ iff $t \xrightarrow[R]{*} s$, reachability is also decidable for left-linear and right-shallow TRSs. □

6.2.2 On the Context-Sensitive Reduction

In the case of context-sensitive, if a TRS R is non-erasing, then the reductions by the inverse R^{-1} of R and replacement map μ is also context-sensitive reductions of another TRS. Therefore, we can prove that reachability for the context-sensitive reduction is also undecidable for left-linear and right-shallow TRSs similarly to the case of right-linear, left-shallow, and non-erasing TRSs.

Theorem 133 *Reachability is undecidable for left-linear and right-shallow TRSs with respect to the context-sensitive reduction.*

Proof: The context-sensitive reductions of the TRS $R'^{-1} = \{r \to l \mid l \to r \in R'\}$ obtained from the TRS R' and replacement map μ in the proof of Theorem 130 is also the context-sensitive reduction. Thus, by simulating the reduction $\varepsilon_1 \xhookrightarrow[R]{*}{}^{\mu}\ \mathsf{e}$ inversely, we have $\mathsf{e} \xhookrightarrow[R^{-1}]{*}{}^{\mu}\ \varepsilon_1$ iff a PCP has a solution. □

6.2.3 On the Innermost Reduction

Reachability for the innermost reduction of left-linear and right-shallow TRSs is undecidable, while it is decidable for right-linear and left-shallow TRSs. It is mainly because for a innermost reduction $s \xrightarrow[R]{}{}^{\text{in}} t$, its inverse $t \xrightarrow[R^{-1}]{} s$ does not always become innermost reduction as the following example.

Example 134 For a TRS $R = \{a \to b, f(x) \to g(x, a)\}$, the reduction $f(b) \xrightarrow[R]{}$ $g(b, a)$ is innermost but $g(b, a) \xrightarrow[R^{-1}]{} f(b)$ is not an innermost reduction since the proper subterm a in $g(b, a)$ is not a normal form.

The following theorem shows undecidability of reachability.

Theorem 135 *Reachability is undecidable for left-linear and right-shallow TRSs with respect to the innermost reduction.*

Proof: Let the TRS R'^{-1} in the proof of Theorem 133. Then a PCP has a solution iff $\mathsf{e} \xrightarrow[R'^{-1}]{*}{}^{\text{in}} \varepsilon_1$. □

6.2.4 On the Outermost Reduction

Similarly to the innermost reduction, for a outermost reduction $s \xrightarrow[R]{}{}^{\text{out}} t$, its inverse $t \xrightarrow[R^{-1}]{} s$ does not always become outermost reduction as the following example.

Example 136 For a TRS $R = \{f(x) \to g(x))\}$, the reduction $g(f(b)) \xrightarrow[R]{}$ $g(g(b))$ is outermost but $g(g(b)) \xrightarrow[R^{-1}]{} g(f(b))$ is not an outermost reduction since the outermost redex position of $g(g(b))$ is the root position.

We conjecture that reachability is decidable for the outermost reduction of left-linear, right-shallow, and non-erasing TRSs, that is the symmetric class of right-linear and left-shallow TRSs similarly to the innermost case. However, it is still open.

Conjecture 137 *Reachability is decidable for left-linear, right-shallow, and non-erasing TRSs with respect to the outermost reduction.*

Since we could not find terms and a left-linear and right-shallow R in which reachability problem corresponds to PCP even if R has erasing variables, we may remove non-erasing from the above conjecture.

Chapter 7

Controlled Term Rewriting

Motivated by the problem of verification of imperative tree transformation programs (e.g. programs of XSLT or XQuery update), we formalize controlled term rewriting systems (CntTRSs), which are the combinations of term rewriting rules with constraints selecting the possible positions. These constraints are specified, for each rewrite rule, by a selection automaton which defines a set of positions in a term based on tree automata computations. Thanks to selection automaton, CntTRS can express explicitly some context condition to be applied a rewrite rule while the context-sensitive reduction can specify rewritable arguments. Therefore, CntTRS is a model of XSLT or XQuery update, which have the feature to specify the context condition of rewritten position.

In Section 1, we describe formalization and some examples of CntTRSs and prefix controlled TRSs (pCntTRSs), which are the restricted forms of CntTRSs. However, it turns out quickly (Examples at the end of Section 7.1) that even the restricted pCntTRS are actually too powerful for preserving regularity, even for very simple rewrite rules.

Therefore, we consider in Section 7.2 the classes of *context-free* (CF) and *context-sensitive* (CS) tree languages, which are both strictly larger than the class of TA languages (also called *regular* tree languages). We also define the so called CF and *monotonic* classes of rewrite systems (without control). A rewrite rule is CF if its *lhs* is of the form $f(x_1, \ldots, x_n)$, where $x_1, \ldots, x_n$ are distinct variables, and monotonic if the size of its *lhs* is larger or equal to the size of its *rhs*. We show that CF and monotonic TRS respectively preserve CF tree languages and CS tree languages.

The monotonic uncontrolled rewrite rules already have the full power of CS tree grammars. Adding control with SA does not improve their expressiveness, and it follows that reachability is PSPACE-complete and model checking undecidable for monotonic CntTRSs (Section 7.3.1). Similar results also hold for CF CntTRSs without collapsing rules ()even when restricting to prefix control (Section 7.3.2)).

When allowing depth-reducing rules (Section 7.3.3), reachability becomes undecidable, even for flat CntTRS (*lhs* and *rhs* of rewrite rules are of depth at most one) and in the case of words (*i.e.* function symbols of arity at most one). Similarly, reachability is undecidable for ground CntTRSs. When restricting to words, prefix control and flat rewrite rules, reachability is decidable in PSPACE.

Finally, we consider in Section 7.3.4 a relaxed form of the prefix control of a rewrite rule, where the selection is done by considering the term in input modulo the rewrite relation. We obtain a regularity preservation result for this recursive form of prefix controlled rewriting, using alternating tree automata with ε-transitions.

7.1 Formalization of Controlled Term Rewriting

7.1.1 Selection automata

Besides being able to recognize terms, tree automata can also be used to select positions in a term [9, 25]. We propose here a definition of position selection by TA very close to [9].

A *selection automaton* (SA) $\mathcal{A}$ is a quadruple $\langle Q, Q^f, S, \Delta\rangle$ where $\langle Q, Q^f, \Delta\rangle$ is a tree automaton denoted $ta(\mathcal{A})$ and S is a set of states of Q called selection states. Given a SA $\mathcal{A}$ and a term $t \in \mathcal{T}(\mathcal{F})$, the set of positions of t selected by $\mathcal{A}$ is defined as

$$\mathrm{sel}(\mathcal{A}, t) = \{p \in \mathrm{Pos}(t) \mid \exists \rho \in sruns(ta(\mathcal{A}), t), \rho(p) \in S\}.$$

Note that it is required that t is recognized by $\mathcal{A}$ in order to select positions.

We shall consider below a restricted kind of selection by TA, where a position p in a term t is selected only according to its strict prefix (*i.e.* the sequence of symbols labeling the path from the root of t down to the immediate ancestor of p), which is tested for membership to a regular (word) language. More precisely, a selection automaton $\mathcal{A} = \langle Q, Q^f, S, \Delta \rangle$ is called *prefix* if Q contains two special states: q_0 (universal state) and q_s (selection state), $Q^f \subseteq Q \setminus \{q_0\}$, $S = \{q_s\}$, and Δ contains $f(q_0, \ldots, q_0) \to q_0$ and $f(q_0, \ldots, q_0) \to q_s$ for all $f \in F$, and Δ contains some other transition rules of the form $f(q_1, \ldots, q_n) \to q$ where $q \in Q \setminus \{q_0, q_s\}$ and there exists exactly one $i \leq n$ such that $q_i \neq q_0$. Intuitively, assume that we are given a finite automaton $\mathcal{B}$ defining the strict prefixes of selected positions. Then q_s is the initial state of $\mathcal{B}$, F is the set of final states of $\mathcal{B}$, and for all $f \in F$ where $\mathrm{ar}(f) = n$, Δ contains n rules $f(q_0, \ldots, q_0, q', q_0, \ldots, q_0) \to q$ for each transition $q' \xrightarrow{a} q$ of $\mathcal{B}$. Note that with this definition, the root position is always selected by a prefix selection automaton.

7.1.2 Controlled Term Rewriting Systems

We propose a formalism that strictly extends standard term rewriting systems by the restricting the rewritable positions to positions selected by a given SA. Formally, a *controlled term rewriting system* (CntTRS) $\mathcal{R}$ is a finite set of *controlled rewrite rules* of the form $\mathcal{A} : l \to r$, made of a SA $\mathcal{A}$ and a *rewrite rule* $l \to r$ such that $l \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}$ (the left-hand side of the rule), and $r \in \mathcal{T}(\mathcal{F}, \mathrm{Var}(l))$ (the right-hand side). The size of $\mathcal{R}$ is the number of the rewrite rules in $\mathcal{R}$.

A term s rewrites to t in one step by an CntTRS $\mathcal{R}$, denoted by $s \xrightarrow[\mathcal{R}]{} t$, if there exists a controlled rewrite rule $\mathcal{A} : l \to r \in \mathcal{R}$, a position $p \in \mathrm{sel}(\mathcal{A}, s)$, and a substitution σ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$. In this case, s is said to be $\mathcal{R}$-*reducible*, and otherwise s is called an $\mathcal{R}$-*normal form*. The reflexive and transitive closure of $\xrightarrow[\mathcal{R}]{}$ is denoted as $\xrightarrow[\mathcal{R}]{*}$.

Example 138 Let us consider the CntTRS $\mathcal{R} = \{(1)\, \mathcal{A}_1 : a \to c, (2)\, \mathcal{A}_2 : b \to c, (3)\, \mathcal{A}_3 : f(x, y) \to g(x, y)\}$ where each SA is as follows ($Q = \{q_1, q_2, q_f\}$):

$$\begin{array}{rcl}\mathcal{A}_1 & = & \langle Q, \{q_f\}, \{q_1\}, \{a \to q_1, b \to q_2, f(q_1, q_2) \to q_f\}\rangle \\ \mathcal{A}_2 & = & \langle Q, \{q_f\}, \{q_2\}, \{c \to q_1, b \to q_2, g(q_1, q_2) \to q_f\}\rangle \\ \mathcal{A}_3 & = & \langle Q, \{q_f\}, \{q_f\}, \{c \to q_1, b \to q_2, f(q_1, q_2) \to q_f\}\rangle\end{array}$$

Then, the following rewriting is possible with $\mathcal{R}$. $f(a,b) \xrightarrow[(1)]{} f(c,b)$ since $sel(\mathcal{A}_1, f(a,b)) = \{1\}$ and the subterm of $f(a,b)$ at the position 1 is a. Similarly, we have $f(c,b) \xrightarrow[(3)]{} g(c,b) \xrightarrow[(2)]{} g(c,c)$ where $sel(\mathcal{A}_3, f(c,b)) = \{\varepsilon\}$, and $sel(\mathcal{A}_2, g(c,b)) = \{2\}$. □

The following example shows that CntTRSs are too powerful to preserve regularity.

Example 139 Let us consider the following CntTRS $\mathcal{R}$ over the unary signature F with $\mathcal{F}_1 = \{a, b, c, d\}$ and $\mathcal{F}_0 = \{\bot\}$. Let $\mathcal{R}$ be the CntTRS containing the following controlled rewrite rules. The SA of these rules select one position per term, and they are represented by a regular expression where the selected letter is underlined.

$$\begin{array}{llcccllcccl}(1) & c^*a^*d^*\underline{d}\,b^* & : & d(x) & \to & b'(x) & (2) & c^*\underline{c}\,a^*d^*b'b^* & : & c(x) & \to & a'(x) \\ (3) & c^*a'a^*d^*\underline{b'}\,b^* & : & b'(x) & \to & b(x) & (4) & c^*\underline{a'}\,a^*d^*b^* & : & a'(x) & \to & a(x)\end{array}$$

More precisely, the SA for the above rules are respectively (Q is the state set $\{q_a, q_b, q_c, q_d, q\}$)

$$\begin{array}{rcl}\mathcal{A}_1 & = & \big\langle Q, \{q_c, q_a, q_d, q\}, \{q\}, \{\bot \to q_b, b(q_b) \to q_b, d(q_b) \to q, d(q|q_d) \to q_d, \\ & & a(q|q_d|q_a) \to q_a, c(q_a|q_c|q) \to q_c\}\big\rangle \\ \mathcal{A}_2 & = & \big\langle Q, \{q_c, q\}, \{q\}, \{\bot \to q_b, b(q_b) \to q_b, b'(q_b) \to q_d, d(q_d) \to q_d, \\ & & a(q_d|q_a) \to q_a, c(q_a) \to q, c(q|q_c) \to q_c\}\big\rangle \\ \mathcal{A}_3 & = & \big\langle Q, \{q_c\}, \{q\}, \{\bot \to q_b, b(q_b) \to q_b, b'(q_b) \to q, d(q|q_d) \to q_d, a(q|q_d) \to q_a, \\ & & a'(q|q_d|q_a) \to q_c, c(q_c) \to q_c\}\big\rangle \\ \mathcal{A}_4 & = & \big\langle Q, \{q_c, q\}, \{q\}, \{\bot \to q_b, b(q_b) \to q_b, d(q_b|q_d) \to q_d, a(q_b|q_d|q_a) \to q_a, \\ & & a'(q_b|q_d|q_a) \to q, c(q|q_c) \to q_c\}\big\rangle\end{array}$$

Note that these SA are all deterministic. The SA $\mathcal{A}_1$ selects the last d (starting from the top), $\mathcal{A}_2$ selects the last c when there is a b', $\mathcal{A}_3$ selects the b' when there is a a', and $\mathcal{A}_4$ selects the a' when there is no b'. The closure of the regular tree language $L = c^+(d^+(\perp))$ by the CntTRS is such that $\xrightarrow[\mathcal{R}]{}[L] \cap a^*(b^*(\perp)) = \{a^n b^n \mid n \geq 0\}$. Therefore, $\xrightarrow[\mathcal{R}]{}[L]$ is a non regular tree language (it is a context-free tree language). □

We call *prefix controlled term rewriting systems* (pCntTRSs), the special cases of CntTRS such that every SA in a controlled rewrite rule are a prefix SA,

A controlled rewrite rule $\mathcal{A} : l \to r$ is ground, *flat*, *linear*, *shallow* if l and r are so. It is *collapsing* if r is a variable. A CntTRS is *flat*, *linear*, etc if all its rules are so.

Example 140 Using the same signature as in Example 139, we can obtain a context-free set of descendants with a flat pCntTRS. Indeed, intersection of a^*b^* and the closure of c^*d^* by the following set of rewrite rules is $\{a^n b^m \mid n \geq m\}$ which is CF and not regular. In the rewrite rules, we use an informal description of the languages of the prefix allowed, instead of giving explicitly the prefix SA.

$$\begin{array}{lclcl@{\qquad}lclcl}
\text{no } a', \text{ no } a & : & c(x) & \to & a'(x), & \text{exactly one } a' & : & d(x) & \to & b'(x), \\
\text{no } a', \text{ no } a & : & a'(x) & \to & a(x), & \text{no } a', \text{ no } b', \text{ no } b & : & b'(x) & \to & b(x).
\end{array}$$

□

It is not difficult to generalize the construction of Example 140 in order to obtain a context sensitive rewrite closure of the form $\{a^n b^m c^p \mid n \geq m \geq p\}$, starting from a regular set of the form $c^*d^*e^*$ and using a flat pCntTRS.

7.2 Context-Free and Context-Sensitive Tree Languages

In this section, we define the context-free and context-sensitive sets of terms, and give properties of their closure under term rewriting.

A rewrite rule over Σ is called *context-free* (CF) if it is of the form $f(x_1, \ldots, x_n) \to r$ where $r \in \mathcal{T}\big(\Sigma, \{x_1, \ldots, x_n\}\big)$, $x_1, \ldots, x_n$ are distinct variables and $f \in \Sigma_n$. Recall that when $r = x_i$ for some $i \leq n$, then the rule is called collapsing, and is called *monotonic* if it is of the form $C[x_1, \ldots, x_n] \to D[x_1, \ldots, x_n]$ where C and D are two contexts over Σ and such that $\|C\| \leq \|D\|$ and $x_1, \ldots x_n$ are distinct variables (note that it implies that the rule is linear).

A *tree grammar* (TG, see *e.g.* [5]) is a tuple $\mathcal{G} = \langle N, S, \Sigma, P\rangle$ where N is a finite set of *non-terminal* symbols with arities, $S \in N$ has arity 0, it is called the *axiom* of $\mathcal{G}$, Σ is a signature disjoint from N, (its elements are also called *terminal* symbols) and P is a set of (uncontrolled) rewrite rules, called *production rules*, of the form $l \to r$ where l, r are terms of $\mathcal{T}(\Sigma \cup N, X)$ such that l contains at least one non-terminal. The tree grammar $\mathcal{G}$ is *regular* if all non-terminal symbols of N have arity 0 and all production rules of P have the form $A \to r$, with $A \in N$ and $r \in \mathcal{T}(\Sigma \cup N)$. It is *context-free* (CFTG), resp. *context-sensitive* (CSTG) if all production rules are CF, resp. monotonic. In the two later cases, we assume from now on without loss of generality that every production rule either has the form $A(x_1, \ldots, x_n) \to a(x_1, \ldots, x_n)$ where $A \in N$ and $a \in \Sigma_n$, or it does not contain terminal symbols of Σ, by introducing the non-terminal symbol $\langle b\rangle$, the production rule $\langle b\rangle \to b$, and replace all b in the other production rules by $\langle b\rangle$.

The *language* generated by $\mathcal{G}$, denoted by $\mathcal{L}(\mathcal{G})$, is the set of terms of $\mathcal{T}(\Sigma)$ which can be reached by successive applications of the production rules, starting from the axiom, *i.e.* $\mathcal{L}(\mathcal{G}) = \{t \in \mathcal{T}(\Sigma) \mid S \xrightarrow[P]{*} t\}$. A tree language is called regular (resp. CF, CS) if it is the language of a regular (resp. CF, CS) grammar.

Note that the classical cases of word languages are particular cases of the above, if the symbols of $N \cup \Sigma$ are unary symbols of a unary signature.

The regular tree grammars are equivalent in expressiveness to TA. There exists a model of pushdown TA equivalent to the CF tree grammars [15]. The *membership* problem is, given a tree grammar $\mathcal{G}$ over Σ and a term $t \in \mathcal{T}(\Sigma)$, to decide whether $t \in \mathcal{L}(\mathcal{G})$. The *emptiness* problem is, given a tree grammar $\mathcal{G}$, to decide whether $\mathcal{L}(\mathcal{G}) = \emptyset$.

The following proposition is already known.

Proposition 141 *Membership and emptiness are decidable in PTIME for CFTG.*

The following result (perhaps a folklore knowledge) is almost immediate from the above definitions.

Proposition 142 *Given a CFTG $\mathcal{G}$ and a CF TRS R, one can construct in PTIME a CFTG generating the closure of $\mathcal{L}(\mathcal{G})$ by R, and whose size is polynomial in the size of $\mathcal{G}$ and R.*

Proof: Let $\mathcal{G} = \langle N, S, \Sigma, P \rangle$ be a CFTG and R a CF TRS over Σ. For all $a \in \Sigma$, we create a new non-terminal N_a with the same arity as a. Let $N' = N \cup \{N_a \mid a \in \Sigma\}$, and let P' be obtained from P by replacing every production rule $A(x_1, \ldots, x_n) \to a(x_1, \ldots, x_n)$, with $A \in N$ and $a \in \Sigma$, by $A(x_1, \ldots, x_n) \to N_a(x_1, \ldots, x_n)$. Moreover, we add to P' the rules obtained from the rules of R by replacing every symbol $a \in \Sigma$ by N_a. The CFTG $\mathcal{G}' = \langle N', S, \Sigma, P' \rangle$ generates the closure of $\mathcal{L}(\mathcal{G})$ by R. $\Box$

Corollary 143 *Reachability and RMC are decidable in PTIME for CF TRS.*

Proof: For the RMC, we use the fact that the intersection of the languages of a CF tree grammar $\mathcal{G}$ and a TA $\mathcal{A}$ is the language of a CF tree grammar whose size is the product of the respective sizes of $\mathcal{G}$ and $\mathcal{A}$. $\Box$

Note however that *joinability*, is undecidable for CF TRS [3], because the emptiness of intersection is undecidable for CF tree languages.

We can also observe that the CF TRS are left linear, but in general not right linear. They are the symmetric the so called right-linear, right-shallow and non-collapsing TRS, whose rules have the form $f(x_1, \ldots, x_n) \to r$, where $r \in \mathcal{T}(\Sigma, \{x_1, \ldots, x_n\}) \setminus X$. It has been shown that these TRS preserve regularity [24]: the closure of a regular tree language by such a TRS is a regular tree language. The decidability of reachability for CF TRS is already a consequence of this former result. It has been observed, see *e.g.* [16], that in several cases, one class of word rewrite system preserves regularity and its symmetric class preserves CF languages.

To our knowledge, the case of CSTG and monotonic TRS was not studied before but it is not very surprising.

Proposition 144 *Membership is PSPACE-complete for CSTG.*

Proof: The hardness is an immediate consequence of the same result for CS (word) grammars [22], which are a particular case of CSTG. The decision algorithm is shown in the below and PSPACE follows from Lemmas 149 and 150.

$\Box$

We give here a non-deterministic linear space decision algorithm of the membership problem for CSTG, by simulating CSTG by linear bounded automata.

Let α be the transformation of terms of $\mathcal{T}(\mathcal{F})$ into strings defined as follows: $\alpha(x) =$ "(x)" for $x \in X$ and $\alpha(f(t_1, \ldots, t_n)) =$ "$(f($"$\alpha(t_1) \cdots \alpha(t_n)$"$))$" for $n \geq 0$.

Lemma 145 *$\alpha(C[t_1, \ldots, t_n]) = \alpha(C)[\alpha(t_1), \ldots, \alpha(t_n)]$ (where $\alpha(C) = \alpha(C[x_1, \ldots, x_n])$).*

Proof: We show this lemma by induction on the size of C.

If $|C| = 0$, then this lemma holds trivially.

Otherwise, let $p_1, \ldots, p_n$ be the positions where t_i's occur in $C[t_1, \ldots, t_n]$. We take $C[t_1, \ldots, t_n]$ as $f(t'_1, \ldots, t'_m)$ and then we have $\alpha(C[t_1, \ldots, t_n]) = \alpha(f(t'_1, \ldots, t'_m)) =$ "$(f($"$\alpha(t'_1) \cdots \alpha(t'_m)$"$))$". For j such that there exists p_i where $jp' = p_i$ for some p', t'_j is represented as $C_j[t_i]$ by some context C_j and hence $\alpha(t'_j) = \alpha(C_j[t_i]) = \alpha(C_j)[\alpha(t_i)]$ from the induction hypothesis. By applying the above claim for all p_i's, we have $\alpha(C[t_1, \ldots, t_n]) = \alpha(C)[\alpha(t_1), \ldots, \alpha(t_n)]$. □

Lemma 146 *$|\alpha(t)| = 5|t|_\Sigma + 3|t|_X$ where $|\alpha(t)|$ is the length of $\alpha(t)$, $|t|_\Sigma$ is the number of the occurrence of signatures in Σ, and $|t|_X$ is the number of the occurrence of variables in X.*

Proof: We show this lemma by induction on the structure of term.

If $t = x$, then $|\alpha(t)| = |$"(x)"$| = 3 = 5|t|_\Sigma + 3|t|_X$.

If $t = f(t_1, \ldots, t_n)$, then $|\alpha(t)| = |$"$(f($"$\alpha(t_1) \cdots \alpha(t_n)$"$))$"$| = 5 + |\alpha(t_1) \cdots \alpha(t_n)|$. From the induction hypothesis, we have $|\alpha(t_1) \cdots \alpha(t_n)| = 5\Sigma_{i=1}^n |t_i|_\Sigma + 3\Sigma_{i=1}^n |t_i|_X$.

Thus, we have $|\alpha(t)| = 5 + 5\Sigma_{i=1}^n |t_i|_\Sigma + 3\Sigma_{i=1}^n |t_i|_X = 5|t|_\Sigma + 3|t|_X$. □

From Lemma 145, a term $C[l\sigma]$ is transformed to $\alpha(C)[\alpha(l)\alpha(\sigma)]$ where $\alpha(\sigma) = \alpha(x\sigma)$ for all domain x of σ. Hence, we can simulate the term rewriting $C[l\sigma] \to C[r\sigma]$ by the string rewriting $\alpha(C)[\alpha(l)\alpha(\sigma)] \to \alpha(C)[\alpha(r)\alpha(\sigma)]$.

Since each production rule of CSTG is monotonic, we can simulate inverse of production rules by LBA.

In the following, we show the simulation of one-step of inverse production of CSTG by LBA. For the production rule $l \to r$, let $\alpha(l\sigma) =$

$l_1(x_1\sigma')l_2(x_2\sigma')\cdots(x_m\sigma')l_n$, $\alpha(r\sigma') = r_1(x_{m_1}\sigma')r_2(x_{m_2}\sigma')\cdots(x_{m_m}\sigma')r_{n'}$ where $x_{m_j} \in \{x_1, \dots x_n\}$, and $(x\sigma') = \alpha(x\sigma)$. We simulate $\alpha(l\sigma) \to \alpha(r\sigma)$.

In the following, we use different notation for LBA from the one in the proof for Proposition 162 for readability. We denote the configuration such that the head of LBA reads u in $u_1 \cdots u \cdots u_n$ and whose state is q as $\|{:}u_1 \cdots uq \cdots u_n{:}\|$, while we denote $\|{:}u_1 \cdots u^q \cdots u_n{:}\|$ in the proof for Proposition 162.

Simulation of one-step of inverse production. LBA $\mathcal{M}^l$ can simulate the string rewriting $\alpha(l) = l_1(x_1\sigma)l_2(x_2\sigma)\cdots(x_m\sigma)l_n\sigma \to \alpha(r) = r_1(x_{m_1}\sigma)r_2(x_{m_2}\sigma)\cdots(x_{m_m}\sigma)r_{n'}\sigma$ where $l \to r \in R$ of the inverse monotonic TRS R, *i.e.* $|l| \geq |r|$, $l_i, r_j \in (\Sigma \cup \{(,)\})^*$ for $1 \leq i \leq n$ and $1 \leq j \leq n'$, $\sigma : X \to (\Sigma \cup \{(,)\})^*$ is a substitution, $x_i \in X$ for $1 \leq i \leq m$, and $x_{m_j} \in \{x_1, \dots x_m\}$ and no x_i and x_j for $i \neq j$ are same.

$\mathcal{M}^l$ is constructed by connecting the following automata.

1. LBA $\mathcal{M}^l_{\mathtt{in}}$ such that

   Initial state $q^l_{\mathtt{in}}$

   Final state $q^{l_1}_{\mathtt{in}}$

   Behavior Head of the automaton moves right and can change state to $q^{l_1}_{\mathtt{in}}$ in any time.

2. LBA $\mathcal{M}^{l_i}$ for $1 \leq i \leq n$ such that

   Initial state $q^{l_i}_{\mathtt{in}}$

   Final state $q^{l_i}_f$

   Behavior $u_1 q^{l_i}_{\mathtt{in}} \cdots u_k v \xrightarrow[\mathcal{M}^{l_i}]{*} \flat^* v^{q^{l_i}_f}$ if $u_1 \cdots u_n = l_i$. Otherwise, the automaton terminate and input does not accepted. The final state appears at v *i.e.* the next symbol of rewritten l_i.

3. LBA $\mathcal{M}^{i()}$ for $1 \leq i \leq n-1$ and $n > 1$ such that

   Initial state $q^{l_i}_f$

   Final state $q^{l_{i+1}}_{\mathtt{in}}$

Behavior If first symbol is "(", then move head to next symbol of corresponding ")" and change state to $q_{\mathtt{in}}^{l_{i+1}}$.

4. LBA $\mathcal{M}^{\leftarrow}$ such that

   Initial state $q_f^{l_n}$

   Final state $q_{\mathtt{in}}^{m\sigma}$

   Behavior Move left until reading ")".

5. LBA $\mathcal{M}^{i\sigma}$ for $1 \leq i \leq m$such that

   Initial state $q_{\mathtt{in}}^{i\sigma}$

   Final state $q_f^{i\sigma}$

   Behavior Move "$(x_i\sigma)$" to right until the next symbol become other than $\flat$. Final state appears at the first position of "(" in "$(x_i\sigma)$".

6. LBA $\mathcal{M}^{i\sigma\leftarrow}$ for $2 \leq i \leq m$ such that

   Initial state $q_f^{i\sigma}$

   Final state $q_{\mathtt{in}}^{(i-1)\sigma}$

   Behavior Move left until reading ")".

7. LBA $\mathcal{M}^{sh}$ such that

   Initial state $q_f^{1\sigma}$

   Final state q_f^{sh}

   Behavior Shuffle "$(x_1\sigma)\cdots(x_m\sigma)$" to "$(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)$". Final state appears at first "(" in "$(x_{m_1}\sigma)$".

8. LBA $\mathcal{M}'^{\leftarrow}$ such that

   Initial state q_f^{sh}

   Final state $q_{\mathtt{in}}^{r_1}$

   Behavior Move left until reading the symbol other than $\flat$.

9. LBA $\mathcal{M}^{r_i}$ for $1 \leq i \leq n'$ such that

   Initial state $q^{r_i}_{\mathtt{in}}$

   Final state $q^{r_i}_f$

   Behavior Rewrite $\flat^*$ to r_i. The final state appears at the next symbol of r_i.

10. LBA $\mathcal{M}'^{i()}$ for $1 \leq i \leq n'$ such that

    Initial state $q^{r_i}_f$

    Final state $q^{r_{i+1}}_{\mathtt{in}}$

    Behavior Move "(u)" that is most near and at right side of head to left until no symbol $\flat$ is there at left of "(u)". Final state appears at next symbol of ")" of "(u)".

11. LBA $\mathcal{M}^\flat$ such that

    Initial state $q^{r_{n'}}_f$

    Final state q^l_f

    Behavior Move all of $\flat$ to right-end and finally head move to left-end.

    Move $\flat$ to right-end.

In the following, we prove that $C[l\sigma] \underset{R}{\rightarrow} C[r\sigma]$ iff $\alpha(C)[\alpha(l\sigma)] \overset{*}{\underset{\mathcal{A}}{\rightarrow}} \alpha(C)[\alpha(r\sigma)]$ for inverse monotonic TRSs.

Lemma 147 *If* $C[l\sigma] \underset{R}{\rightarrow} C[r\sigma]$ *for inverse monotonic TRS* $\mathcal{R}$*, then* $| : q^l_{\mathtt{in}}\alpha(C)[\alpha(l\sigma)]:\| \overset{*}{\underset{\mathcal{A}}{\rightarrow}} \flat_l q^l_f \alpha(C)[\alpha(r\sigma)]\flat^* : |$.

Proof: From Lemma 145, we can take $\alpha(C) = u_1 \cdots u_k \Box v_1 \cdots , v_h$, $\alpha(l\sigma) = l_1(x_1\sigma)\cdots(x_m\sigma)l_n$, and $\alpha(r\sigma) = r_1(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)r_n$. Thus, this lemma holds from the following transition sequence.

$$
\begin{array}{rl}
 & \|{:}q^l_{\mathtt{in}} u l_1(x_1\sigma)\cdots(x_m\sigma)l_n v{:}\| \\
\xrightarrow[\mathcal{M}^l]{*} & \|{:}u q^{l_1}_{\mathtt{in}} l_1(x_1\sigma)\cdots(x_m\sigma)l_n v{:}\| \\
\xrightarrow[\mathcal{M}^{l_1}]{*} & \|{:}u\flat^* q^{l_1}_{\mathtt{in}}(x_1\sigma)\cdots(x_m\sigma)l_n v{:}\| \\
\xrightarrow[\mathcal{A}^{1()}]{*} & \|{:}u\flat^*(x_1\sigma)q^{l_2}_{\mathtt{in}}\cdots(x_m\sigma)l_n v{:}\| \\
\xrightarrow[\mathcal{A}^{l_i},\mathcal{A}^{i()}]{*} & \|{:}u\flat^*(x_1\sigma)\cdots(x_m\sigma)\flat^* q^{l_n}_f v{:}\| \\
\xrightarrow[\mathcal{A}^{\leftarrow}]{*} & \|{:}u\flat^*(x_1\sigma)\cdots(x_m\sigma q^{m\sigma}_{\mathtt{in}})\flat^* v{:}\| \\
\xrightarrow[\mathcal{A}^{m\sigma}]{*} & \|{:}u\flat^*(x_{m_1}\sigma)\sigma)\cdots\flat^* q^{m\sigma}_f(x_m\sigma)v{:}\| \\
\xrightarrow[\mathcal{A}^{i\sigma},\mathcal{A}^{i\sigma\leftarrow}]{*} & \|{:}u\flat^* q^{1\sigma}_f(x_1\sigma)\cdots(x_m\sigma)v{:}\| \\
\xrightarrow[\mathcal{A}^{sh}]{*} & \|{:}u\flat^* q^{sh}_f(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)v{:}\| \\
\xrightarrow[\mathcal{A}'^{\leftarrow}]{*} & \|{:}u q^{r_1}_{\mathtt{in}}\flat^*(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)v{:}\| \\
\xrightarrow[\mathcal{A}^{r_1}]{*} & \|{:}u r_1 q^{r_1}_f\flat^*(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)v{:}\| \\
\xrightarrow[\mathcal{A}'^{i()}]{*} & \|{:}u r_1(x_{m_1}\sigma)q^{r_2}_{\mathtt{in}}\flat^*\cdots(x_{m_m}\sigma)v{:}\| \\
\xrightarrow[\mathcal{A}^{r_i},\mathcal{A}'^{i()}]{*} & \|{:}u r_1(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)r_{n'}q^{r_{n'}}_f\flat^* v{:}\| \\
\xrightarrow[\mathcal{A}^{\flat}]{*} & \|{:}q^l_f u r_1(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)r_{n'}v\flat^*{:}\|
\end{array}
$$

Since $|l| \geq |r|$, we can generate the all r_i's because $\Sigma^n_{i=1}|l_i| < \Sigma^{n'}_{i=1}|r_i|$ and there are sufficient $\flat$'s to be rewritten to each r_i. $\square$

Lemma 148 *If* $\|{:}q^l_{\mathtt{in}}\alpha(C)[\alpha(l'\sigma)]{:}\| \xrightarrow[\mathcal{A}^l]{*} \|{:}q^l_f\alpha(C)[\alpha(r'\sigma)]\flat^*{:}\|$*, then* $C[l'\sigma] \xrightarrow[\mathcal{R}]{} C[r'\sigma]$ *for monotonic TRS* $\mathcal{R}$*.*

Proof: From Lemma 145, we can represent $\alpha(C[l'\sigma])$ and $\alpha(C[r'\sigma])$ as $ul'_1(x_1\sigma)\cdots(x_m\sigma)l'_n v$ and $ur'_1(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)r'_{n'}v$ where $\alpha(C) = u\Box v$, $\alpha(l'\sigma) = l'_1(x_1\sigma)\cdots(x_m\sigma)l'_n$, and $\alpha(r'\sigma) = r'_1(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)r'_{n'}$. From the construction of $\mathcal{A}^l$, the transition $\|{:}q^l_{\mathtt{in}}ul'_1(x_1\sigma)\cdots(x_m\sigma)l'_n v{:}\| \xrightarrow[\mathcal{A}^l]{*} \|{:}q^l_f r'_1(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)r'_{n'}\flat^*{:}\|$ is as of the form the transition in Lemma 147. Transition $\|{:}uq^{l_1}_{\mathtt{in}}l_1(x_1\sigma)\cdots(x_m\sigma)l_n v{:}\| \xrightarrow[\mathcal{A}^l]{*} \|{:}u\flat^*(x_1\sigma)\cdots(x_m\sigma)\flat^* q^{l_n}_f v{:}\|$ and $\|{:}uq^{r_1}_{\mathtt{in}}\flat^*(x_{m_1}\sigma)\cdots(x_{m_m}\sigma)r_{n'}v{:}\| \xrightarrow[\mathcal{A}^l]{*} \|{:}ur_1(x_{m_1}\ \sigma)\cdots(x_{m_m}\sigma)r_{n'}q^{r_{n'}}_f v{:}\|$ implies that $l_1(x_1\sigma)\cdots(x_m\sigma)l_n = \alpha(l\sigma)$ and $r_1(x_{m_1}\sigma)\ \cdots(x_{m_m}\sigma)r_{n'} = \alpha(r\sigma)$ for some $l \to r \in R$. Thus, we have $C[l'\sigma] \xrightarrow[R]{} C[r'\sigma]$. $\square$

We construct the LBA $\mathcal{A} = \langle Q, \Gamma, q_{\mathtt{in}}, \{q_f\}, \Theta\rangle$ that simulates context-sensitive grammar $\mathcal{G} = \langle N, T, P, S\rangle$ from $\mathcal{A}^l$'s for $l \to r \in P^{-1}$. Q is disjoint union of all Q^l's. Γ are same for all $\mathcal{A}^l$'s. Θ is consists of union of all Θ^l's and the rules $\langle q_{\mathtt{in}}, x, q_{\mathtt{in}}, x, \text{left}\rangle$ for $x \neq \|{:}$, $\langle q_{\mathtt{in}}, \|{:}, q^l_{\mathtt{in}}, \|{:}, \text{right}\rangle$, $\langle q^l_f, \|{:}, q_{\mathtt{in}}, \|{:}, \text{stay}\rangle$, $\langle q_{\mathtt{in}}, \|{:}, q_{S_1}, \|{:}, \text{right}\rangle$, $\langle q_{S_1}, \text{“(”}, q_{S_2}, \text{“(”}, \text{right}\rangle$, $\langle q_{S_2}, S, q_{S_3}, S, \text{right}\rangle$, $\langle q_{S_3}, \text{“)”}, q_\flat, \text{“)”}, \text{right}\rangle$, $\langle q_\flat, \flat, q_\flat, \flat, \text{right}\rangle$, and $\langle q_\flat, {:}\|, q_f, {:}\|, \text{stay}\rangle$.

The following lemmas state completeness and soundness of LBA $\mathcal{A}$.

Lemma 149 *If $t \in \mathcal{L}(\mathcal{G})$, then $\alpha(t) \in \mathcal{A}$.*

Proof: We prove that $\|{:}q_{\mathtt{in}}\alpha(t){:}\| \xrightarrow[\mathcal{A}]{*} {:}\|(S)\flat^* q_f{:}\|$ for the start symbol S of $\mathcal{G}$. If $t = S$, then this lemma holds trivially. Otherwise, let $t \xrightarrow[P]{} t' \xrightarrow[P]{*} S$. We have $\|{:}q_{\mathtt{in}}\alpha(t){:}\| \xrightarrow[\mathcal{A}]{*} q^l_{\mathtt{in}}\|{:}\alpha(t){:}\| \xrightarrow[\mathcal{A}]{*} q^l_f\|{:}\alpha(t'){:}\| \xrightarrow[\mathcal{A}]{} q_{\mathtt{in}}\|{:}\alpha(t'){:}\|$ from Lemma 147. Thus, we have $q_{\mathtt{in}}\|{:}\alpha(t){:}\| \xrightarrow[\mathcal{A}]{*} q_f\|{:}(S)\flat^*{:}\|$ by applying lemma 147 as the above transition. □

Lemma 150 *If $\alpha(t) \in \mathcal{A}$, then $t \in \mathcal{L}(\mathcal{G})$.*

Proof: $\alpha(t) \in \mathcal{A}$ implies $\|{:}q_{\mathtt{in}}\alpha(t){:}\| \xrightarrow[\mathcal{A}]{*} \|{:}(S)\flat^* q_f{:}\|$.

If $t = S$, then this lemma holds trivially.

Otherwise, we have the transition $\|{:}q_{\mathtt{in}}\alpha(t){:}\| \xrightarrow[\mathcal{A}]{*} q^l_{\mathtt{in}}\|{:}\alpha(t){:}\| \xrightarrow[\mathcal{A}]{} q^l_f\|{:}\alpha(t'){:}\| \xrightarrow[\mathcal{A}]{*} \|{:}(S)\flat^* q_f{:}\|$ for some l and t'. From lemma 148, we have $t \xrightarrow[R]{} t'$. Thus, we have $t \xrightarrow[R]{} S$ by applying lemma 148 repeatedly. □

The Proposition 144 follows from Lemmas 149 and 150.

The emptiness problem of CSTG is undecidable since the emptiness problem of CS word grammar is undecidable.

Proposition 151 *Emptiness is undecidable for CSTG.*

Proof: It is a consequence of the same result for CS (word) grammars. □

Proposition 152 *Given a CSTG $\mathcal{G}$ and a monotonic TRS R, one can construct in PTIME a CSTG generating the closure of $\mathcal{L}(\mathcal{G})$ by R, and whose size is polynomial in the size of $\mathcal{G}$ and R.*

Proof: The construction is similar as the one in the proof of Proposition 142. □

7.3 Decidable or Undecidable Properties

7.3.1 Monotonic CntTRS

The result of Proposition 152 can be extended from uncontrolled to controlled monotonic TRS. Intuitively, monotonic TRS are powerful enough to be able to simulate a control with uncontrolled rewrite rules.

Theorem 153 *Given a CSTG $\mathcal{G}$ and a monotonic CntTRS $\mathcal{R}$, one can construct in PTIME a CSTG generating the closure of $L(\mathcal{G})$ by $\mathcal{R}$, and whose size is linear in the size of $\mathcal{G}$ and $\mathcal{R}$.*

In order to prove this theorem, we show how to construct a CSTG $\mathcal{G}_*$ that recognizes the set of terms reachable by $\mathcal{R}$ from the terms in $\mathcal{L}(\mathcal{G})$. The set of production rules P_* of $\mathcal{G}_*$ consists of two sets: the rules P of $\mathcal{G}$ and P_1, that simulate rewriting by $\mathcal{R}$. The basic idea for the construction of $\mathcal{G}_*$ is the introduction of non-terminals of the form $\langle f, q\rangle$ where f is a symbol and q is a state of some SA.

First, we produce the term $\langle t\rangle$ where $t \in \mathcal{L}(\mathcal{G})$ and $\langle t\rangle$ is the term obtained by replacing each symbol f by the non-terminal $\langle f\rangle$. Next, we simulate the rewriting of $\mathcal{R}$ by P_1. We simulate a transition $f(q_1, \ldots, q_n) \to q$ of a SA by some production rules in P_1 of the form $\langle f\rangle(\langle f_1, q_1\rangle(\overline{x_1}), \ldots, \langle f_n, q_n\rangle(\overline{x_n})) \to \langle f, q\rangle(\langle f_1, q_1\rangle(\overline{x_1}), \ldots, \langle f_n, q_n\rangle(\overline{x_n}))$. Finally, if a final state occurs at the root position of a term and a rewrite rule matches the subterm where a selection state appears, then we rewrite the term.

The following example shows an construction of CSTG.

Example 154 Consider the CntTRS $\mathcal{R}$ in Example 138 and the CSTG $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G}) = \{f(a, b)\}$. We construct the CSTG $\mathcal{G}_*$ such that $\mathcal{L}(\mathcal{G}_*) = \{f(a, b), f(c, b),\ g(c, b), g(c, c)\}$. Let the set of production rules P of $\mathcal{G}$ be $\{S \to \langle f\rangle(\langle a\rangle, \langle b\rangle), \langle a\rangle \to a, \langle b\rangle \to b, \langle f\rangle(x_1, x_2) \to f(x_1, x_2)\}$, and mark i to each component of SA to distinguish each SA $\mathcal{A}_i$. Let the axiom of $\mathcal{G}_*$ be S^λ.

We define the set of production rules P_* of $\mathcal{G}_*$ as $P_* = P \cup P' \cup P_{\mathcal{A}} \cup P_{\mathsf{fin}} \cup P_{\mathcal{R}} \cup P_{\mathsf{re}}$ where

$$
\begin{array}{rcl}
P' & = & \{S^\lambda \to \langle f\rangle^\lambda(\langle a\rangle, \langle b\rangle), \langle f\rangle^\lambda(x_1, x_2) \to f(x_1, x_2)\} \\
P_{\mathcal{A}} & = & \{\langle c_1\rangle \to \langle c_1, q^i\rangle \mid c_1 \to q^i \in \Delta_i \text{ for some } i\} \cup \\
& & \left\{ \begin{array}{l|l} \langle f_1\rangle^\lambda(\langle c_1, q_1^i\rangle, \langle c_2, q_2^i\rangle) & c_1, c_2 \in \{a, b, c\}, \\ \to \langle f_1, q^i\rangle^\lambda(\langle c_1, q_1^i\rangle, \langle c_2, q_2^i\rangle) & f_1(q_1^i, q_2^i) \to q^i \in \Delta_i \text{ for some } i \end{array} \right\} \\
P_{\mathsf{fin}} & = & \{\langle f_1, q^i\rangle^\lambda(x_1, x_2) \to \langle f_1, q^i\rangle^\lambda_{\mathsf{fin}}(x_1, x_2) \mid f_1 \in \{f, g\}, q^i \in F_i \text{ for some } i\} \cup \\
& & \left\{ \begin{array}{l|l} & f_1 \in \{f, g\}, \\ \langle f_1, q^i\rangle^\lambda_{\mathsf{fin}}(\langle c_1, q_1^i\rangle, \langle c_2, q_2^i\rangle) & c_1 \in \{a, b, c\}, \\ \to \langle f_1\rangle^\lambda(\langle c_1, q_1^i\rangle_{\mathsf{fin}}, \langle c_2, q_2^i\rangle_{\mathsf{fin}}) & f(q_1^i, q_2^i) \to q \in \Delta_i \text{ for some } i \end{array} \right\} \\
P_{\mathcal{R}} & = & \{\langle a, q_1^1\rangle_{\mathsf{fin}} \to \langle c\rangle\} \cup \{\langle b, q_2^2\rangle_{\mathsf{fin}} \to \langle c\rangle\} \cup \{\langle f, q_f^3\rangle^\lambda_{\mathsf{fin}}(x_1, x_2) \to \langle g\rangle^\lambda(x_1, x_2)\} \\
P_{\mathsf{re}} & = & \{\langle c_1, q^i\rangle \to \langle c_1\rangle, \langle c_1, q^i\rangle_{\mathsf{fin}} \to \langle c_1\rangle \mid c_1 \in \{a, b, c\}, q^i \in Q_i \text{ for some } i\} \cup \\
& & \left\{ \begin{array}{l|l} \langle f_1, q^i\rangle^\lambda(x_1, x_2) \to \langle f_1\rangle^\lambda, & f_1 \in \{f, g\}, \\ \langle f_1, q^i\rangle^\lambda_{\mathsf{fin}}(x_1, x_2) \to \langle f_1\rangle^\lambda(x_1, x_2) & q^i \in Q_i \text{ for some } i \end{array} \right\}
\end{array}
$$

The term $f(c, b)$ is produced by $\mathcal{G}_*$ with the production $S^\lambda \xrightarrow[P']{} \langle f\rangle^\lambda(\langle a\rangle, \langle b\rangle) \xrightarrow[P_{\mathcal{A}}]{*} \langle f, q_f\rangle^\lambda(\langle a, q_1\rangle, \langle b, q_2\rangle) \xrightarrow[P_{\mathsf{fin}}]{*} \langle f\rangle^\lambda(\langle a, q_1\rangle_{\mathsf{fin}}, \langle b, q_2\rangle_{\mathsf{fin}}) \xrightarrow[P_{\mathcal{R}}]{} \langle f\rangle^\lambda(\langle c\rangle, \langle b, q_2\rangle_{\mathsf{fin}}) \xrightarrow[P_{\mathsf{re}}]{} \langle f\rangle^\lambda(\langle c\rangle, \langle b\rangle) \xrightarrow[P \cup P']{*} f(c, b)$.

Here, we show the detailed proof of Theorem 153. At first, we show the detailed construction of CSTG.

We denote the ith rewrite rule of CntTRS by $\mathcal{A}_i : l_i \to r_i$ where $l_i, r_i \in \mathcal{T}(F, X)$ and $\mathcal{A}_i = \langle Q_i, Q_i^f, S_i, \Delta_i\rangle$ is a selection automaton. We assume that Q_i's are disjoint each other. In the sequel, we use large character A for non-terminal, and C, C' for contexts that has no terminal, $\langle t\rangle$ for the term obtained by replacing every signature a in t by the non-terminal $\langle a\rangle$, $\langle t\rangle^\lambda$ for the term obtained by replacing root symbol f of $\langle t\rangle$ by $\langle f\rangle^\lambda$, and let $Q := \bigcup_i Q_i$, $F := \bigcup_i F_i$, and $\Delta := \bigcup_i \Delta_i$.

We sometimes denote the sequence of terms $t_1, \ldots, t_n$ by $\bar{t}$ for readability.

CSTG construction

Input CSTG $\mathcal{G} = \langle N, S, \Sigma, P\rangle$ such that arbitrary terminal a is only produced from the non-terminal $\langle a\rangle$ by the rule $\langle a\rangle(x_1,\ldots,x_n) \to a(x_1,\ldots,x_n)$, and monotonic CntTRS $\mathcal{R}$

Output Context-sensitive tree grammar $\mathcal{G}_* = \langle N_*, S^\lambda, \Sigma, P_*\rangle$ that recognizes $\xrightarrow[\mathcal{R}]{}[\mathcal{L}(\mathcal{G})]$.

Step1(initialize) Let N_* be as follows:

- $N_* := N \cup \{A^\lambda, \langle A, q\rangle, \langle A, q\rangle^\lambda, \langle A, q\rangle_{fin}, \langle A, q\rangle^\lambda_{fin} \mid A \in N, q \in Q, 1 \leq i \leq n\}$ where n is the number of rewrite rules.

Before constructing P_*, we construct P' to mark λ at root symbol. P' consists of the rules in P, $A^\lambda(C[x_1,\ldots,x_n]) \to A'^\lambda(C'[x_1,\ldots,x_n])$ for every $A(C[x_1,\ldots,x_n]) \to A'(C'[x_1,\ldots,x_n]) \in P$, and $\langle a\rangle^\lambda(x_1,\ldots,x_n) \to a(x_1,\ldots,x_n)$ for every $\langle a\rangle(x_1,\ldots,x_n) \to a(x_1,\ldots,x_n) \in P$.

P_* is composed by P' and the following sets of production rules. These are used to simulate a selection automaton.

$$P_{\mathcal{A}} := \left\{ \begin{array}{l|l} \langle f\rangle(\langle f_1, q_1\rangle(\overline{x_1}),\ldots,\langle f_n, q_n\rangle(\overline{x_n})) & \\ \to \langle f, q\rangle(\langle f_1, q_1\rangle(\overline{x_1}),\ldots,\langle f_n, q_n\rangle(\overline{x_n})), & f, f_1,\ldots,f_n \in \Sigma, \\ \langle f\rangle^\lambda(\langle f_1, q_1\rangle(\overline{x_1}),\ldots,\langle f_n, q_n\rangle(\overline{x_n})) & f(q_1,\ldots,q_n) \to q \in \Delta. \\ \to \langle f, q\rangle^\lambda(\langle f_1, q_1\rangle(\overline{x_1}),\ldots,\langle f_n, q_n\rangle(\overline{x_n})). & \end{array} \right\}$$

$$P_{fin} := \left\{ \begin{array}{l|l} \langle f, q_f\rangle^\lambda(\overline{x}) \to \langle f, q_f\rangle^\lambda_{fin}(\overline{x}), & \\ \langle f, q\rangle_{fin}(\langle f_1, q_1\rangle(\overline{x_1}),\ldots,\langle f_n, q_n\rangle(\overline{x_n})) & f, f_1,\ldots,f_n \in \Sigma, \\ \to \langle f\rangle(\langle f_1, q_1\rangle_{fin}(\overline{x_1}),\ldots,\langle f_n, q_n\rangle_{fin}(\overline{x_n})), & f(q_1,\ldots,q_n) \to q \in \Delta. \\ \langle f, q\rangle^\lambda_{fin}(\langle f_1, q_1\rangle(\overline{x_1}),\ldots,\langle f_n, q_n\rangle(\overline{x_n})) & q \in F \\ \to \langle f\rangle^\lambda(\langle f_1, q_1\rangle_{fin}(\overline{x_1}),\ldots,\langle f_n, q_n\rangle_{fin}(\overline{x_n})). & \end{array} \right\}$$

$$P_{\mathcal{R}} := \left\{ \begin{array}{l|l} \langle f, q\rangle_{fin}(\langle l_1\rangle,\ldots,\langle l_n\rangle) \to \langle r_i\rangle, & q \in S_i, \\ \langle f, q\rangle^\lambda_{fin}(\langle l_1\rangle,\ldots,\langle l_n\rangle) \to \langle r_i\rangle^\lambda. & \mathcal{A}_i : f(l_1,\ldots,l_n) = l_i \to r_i \in \mathcal{R}. \end{array} \right\}$$

$$P_{re} := \left\{ \begin{array}{l|l} \langle f, q\rangle^\lambda(\overline{x}) \to \langle f\rangle^\lambda(\overline{x}), & \\ \langle f, q\rangle(\overline{x}) \to \langle f\rangle(\overline{x}), & f \in \Sigma, \\ \langle f, q\rangle^\lambda_{fin}(\overline{x}) \to \langle f\rangle^\lambda(\overline{x}), & q \in Q. \\ \langle f, q\rangle_{fin}(\overline{x}) \to \langle f\rangle(\overline{x}). & \end{array} \right\}$$

$P_{\mathcal{A}}$ is used to simulate transition of SA, P_{fin} is used to propagate information that final state occurs at the root position by marking fin, $P_{\mathcal{R}}$ is used to simulate rewriting by CntTRS $\mathcal{R}$, and P_{re} is used to clear states in each non-terminal of the form $\langle f, q\rangle$.

Here, we show several lemmas in the following. These are necessary to prove Theorem 153.

Lemma 155 *$S \xrightarrow[P]{*} A(t_1, \ldots, t_n)$ iff $S^\lambda \xrightarrow[P']{*} A^\lambda(t_1, \ldots, t_n)$.*

Proof: We can prove this lemma by induction on the length of $\xrightarrow[P]{*}$ for only if part and on $\xrightarrow[P']{*}$ for if part. □

Lemma 156 *$f(t_1, \ldots, t_n) \xrightarrow[\mathcal{A}_i]{*} q$ for all i iff $\langle f\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle) \xrightarrow[\mathcal{G}_*]{*} \langle f, q\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle)$ and $\langle f\rangle^\lambda(\langle t_1\rangle, \ldots, \langle t_n\rangle) \xrightarrow[\mathcal{G}_*]{*} \langle f, q\rangle^\lambda(\langle t_1\rangle, \ldots, \langle t_n\rangle)$.*

Proof: Since Q_i's are disjoint each other, we can easily prove only if part of this Lemma by the rules in $P_{\mathcal{A}}$ and P_{re}. In the following, we prove if part by induction on $||f(t_1, \ldots, t_n)||$.

Let $f(t_1, \ldots, t_n)$ be $f(f_1(\overline{t_1}), \ldots, f_n(\overline{t_n}))$ for $n \geq 0$. Since $\langle f, q\rangle$ is produced by the rule $\langle f\rangle(\langle f_1, q_1\rangle(\overline{x_1}), \ldots, \langle f_n, q_n\rangle(\overline{x_n})) \rightarrow \langle f, q\rangle(\langle f_1, q_1\rangle(\overline{x_1}), \ldots, \langle f_n, q_n\rangle(\overline{x_n}))$ where $f(q_1, \ldots, q_n) \rightarrow q \in \Delta_i$ for some i, we have $f_i(\overline{t_i}) \xrightarrow[\Delta_i]{*} q_i$ from the induction hypothesis. Thus, $f(f_1(\overline{t_1}), \ldots, f_n(\overline{t_n})) \rightarrow q$ follows.

In the case where the root position has mark λ, we can prove similarly. □

Lemma 157 *1. For $q \in Q_i$, $C[f(t_1, \ldots, t_n)] \xrightarrow[\mathcal{A}_i]{*} C[q] \xrightarrow[\mathcal{A}_i]{*} q_f$ for some $q_f \in F_i$ iff $\langle C\rangle^\lambda[\langle f\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle)] \xrightarrow[\mathcal{G}_*]{*} \langle C\rangle^\lambda[\langle f, q\rangle_{fin}(\langle t_1\rangle, \ldots, \langle t_n\rangle)]$ for $||C|| > 0$, and*

2. $f(t_1, \ldots, t_n) \xrightarrow[\mathcal{A}_i]{*} q_f \in F_i$ iff $\langle f\rangle^\lambda(\langle t_1\rangle, \ldots, \langle t_n\rangle) \xrightarrow[\mathcal{G}_*]{*} \langle f, q\rangle^\lambda_{fin}(\langle t_1\rangle, \ldots, \langle t_n\rangle)$.

Proof: 2 of this lemma follows from Lemma 156 and rules in P_{fin}.

We prove both direction of 1 of this lemma by induction on the length of the position p that $f(t_1, \ldots, t_n)$ occur in $C[f(t_1, \ldots, t_n)]$. Let $C[x] = C'[f'(t'_1, \ldots, x, \ldots, t'_n)]$ and hence $C[f(t_1, \ldots, t_n)]$ is represented as $C'[f'(t'_1, \ldots, f(t_1, \ldots, t_n), \ldots, t'_n)]$.

(only if part) Assume that we have the transition $C'[f'(f_1(\overline{t'_1}), \ldots, f(t_1, \ldots, t_n), \ldots, f_n(\overline{t'_n}))] \xrightarrow[\mathcal{A}_i]{*} C'[f'(q'_1, \ldots, q, \ldots, q'_n)] \xrightarrow[\mathcal{A}_i]{} C'[q'] \xrightarrow[\mathcal{A}_i]{*} q_f \in F_i$. Then, from Lemma 156, we have $\langle f_i\rangle(\overline{\langle t'_i\rangle}) \xrightarrow[\mathcal{G}_*]{*} \langle f_i, q'_i\rangle(\overline{\langle t'_i\rangle})$ and $\langle f\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle) \xrightarrow[\mathcal{G}_*]{*} \langle f, q\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle)$, and from the induction hypothesis or 1 of this lemma, we have the production: $\langle C'\rangle[\langle f'\rangle(\langle f_1\rangle(\overline{\langle t'_1\rangle}), \ldots, \langle f\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n\rangle(\overline{\langle t'_n\rangle}))] \xrightarrow[\mathcal{G}_*]{*} \langle C'\rangle[\langle f', q'\rangle_{fin}(\langle f_1\rangle(\overline{\langle t'_1\rangle}), \ldots, \langle f\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n\rangle(\overline{\langle t'_n\rangle}))]$

Thus, we obtaine the production

$$
\begin{aligned}
& \langle C'\rangle[\langle f'\rangle(\langle f_1\rangle(\overline{\langle t'_1\rangle}), \ldots, \langle f\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n\rangle(\overline{\langle t'_n\rangle}))] \\
\xrightarrow[\mathcal{G}_*]{*} \; & \langle C'\rangle[\langle f', q'\rangle_{fin}(\langle f_1, q'_1\rangle(\overline{\langle t'_1\rangle}), \ldots, \langle f, q\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n, q'_n\rangle(\overline{\langle t'_n\rangle}))] \\
\xrightarrow[\mathcal{G}_*]{} \; & \langle C'\rangle[\langle f'\rangle(\langle f_1, q'_1\rangle_{fin}(\overline{\langle t'_1\rangle}), \ldots, \langle f, q\rangle_{fin}(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n, q'_n\rangle_{fin}(\overline{\langle t'_n\rangle}))] \\
\xrightarrow[\mathcal{G}_*]{} \; & \langle C'\rangle[\langle f'\rangle(\langle f_1\rangle(\overline{\langle t'_1\rangle}), \ldots, \langle f, q\rangle_{fin}(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n\rangle(\overline{\langle t'_n\rangle}))]
\end{aligned}
$$

from the rule in P_{fin} and P_{re}, and $f(q'_1, \ldots, q, \ldots, q'_n) \to q' \in \Delta_i$.

(if part) Since the non-terminal $\langle f, q\rangle_{fin}$ must be produced by the rule $\langle f', q'\rangle_{fin}(\langle f_1, q'_1\rangle(\overline{x_1}), \ldots, \langle f, q\rangle(\overline{x}), \ldots, \langle f_n, q'_n\rangle(\overline{x_n})) \to \langle f'\rangle(\langle f_1, q'_1\rangle_{fin}(\overline{x_1}), \ldots, \langle f, q\rangle_{fin}(\overline{x}), \ldots, \langle f_n, q'_n\rangle_{fin}(\overline{x_n}))$, we have the production $\langle C'\rangle[\langle f'\rangle(\langle f_1\rangle(\overline{\langle t'_1\rangle}), \ldots, \langle f\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n\rangle(\overline{\langle t'_n\rangle}))] \xrightarrow[\mathcal{G}_*]{*} \langle C'\rangle[\langle f', q'\rangle_{fin}(\langle f_1, q'_1\rangle(\overline{\langle t'_1\rangle}), \ldots, \langle f, q\rangle(\langle t_1\rangle, \ldots, \langle t_n\rangle), \ldots, \langle f_n, q'_n\rangle(\overline{\langle t'_n\rangle}))]$. Hence, from the induction hypothesis, we have $C'[f'(f_1(\overline{t_1}), \ldots, f(t_1, \ldots, t_n), \ldots, f(\overline{t_n}))] \xrightarrow[\Delta_i]{*} C'[q'] \xrightarrow[\Delta_i]{*} q_f$ for some $q_f \in F_i$. Moreover, we have $f(q_1, \ldots, q, \ldots, q_n) \to q \in \Delta_i$ since states are disjoint for each $\mathcal{A}_i$, and from Lemma 156, $f_i(\overline{t_i}) \xrightarrow[\mathcal{A}_i]{*} q_i$ for all i and $f(t_1, \ldots, t_n) \xrightarrow[\mathcal{A}_i]{*} q$. Therefore, we have $C[f(t_1, \ldots, t_n)] \xrightarrow[\mathcal{A}_i]{*} C[q] \xrightarrow[\mathcal{A}_i]{*} q_f \in F_i$.

□

Lemma 158 *If $C[l\sigma] \xrightarrow[\mathcal{R}]{} C[r\sigma]$ and $C[l\sigma] \in L(\mathcal{G}_*)$, then $C[r\sigma] \in L(\mathcal{G}_*)$.*

Proof: Consider the case where $||C|| > 0$. We suppose that $C[l\sigma]$ is rewritten to $C[r\sigma]$ by the rule $\mathcal{A}_i : f(l_1, \ldots, l_n) \to r$. Then, we have the transition $C[f(l_1, \ldots, l_n)\sigma] \xrightarrow[\mathcal{A}_i]{*} C[q] \xrightarrow[\mathcal{A}_i]{*} q_f$ where $q \in S_i$ and $q_f \in Q_i^f$. Since $S^\lambda \xrightarrow[\mathcal{G}_*]{*} \langle C\rangle[\langle f(l_1, \ldots, l_n)\sigma\rangle]$, we have $S^\lambda \xrightarrow[\mathcal{G}_*]{*} \langle C\rangle^\lambda[\langle f, q\rangle(\langle l_1\rangle, \ldots, \langle l_n\rangle)\langle\sigma\rangle] \xrightarrow[\mathcal{G}_*]{*} \langle C\rangle^\lambda[\langle f, q\rangle_{fin}(\langle l_1\rangle, \ldots, \langle l_n\rangle)\langle\sigma\rangle]$ from 1 of Lemma 157. From the production rule $P_{\mathcal{R}}$, we have $\langle C\rangle[\langle f,q\rangle_{fin}(\langle l_1\rangle,\ldots,\langle l_n\rangle)\langle\sigma\rangle] \xrightarrow[\mathcal{G}_*]{*} \langle C\rangle[\langle r\rangle\langle\sigma\rangle]$. Thus, we have $S^\lambda \xrightarrow[\mathcal{G}_*]{*} C[r\sigma]$. In the case of $||C|| = 0$, we can prove similary to the previous case. $\square$

Lemma 159 *If $s \in L(\mathcal{G})$ and $s \xrightarrow[\mathcal{R}]{*} t$, then $t \in L(\mathcal{G}_*)$.*

Proof: From Lemma 155, we have $S^\lambda \xrightarrow[\mathcal{G}_*]{*} \langle s\rangle^\lambda$, and since P_* has the rules in P and the rule $\langle f\rangle^\lambda(x_1,\ldots,x_n) \to f(x_1,\ldots,x_n)$, we have $S^\lambda \xrightarrow[\mathcal{G}_*]{*} s$. By applying Lemma 158 repeatedly, we also have $t \in L(\mathcal{G}_*)$. $\square$

Lemma 160 *If $t \in L(\mathcal{G}_*)$, then there exists s such that $s \xrightarrow[\mathcal{R}]{*} t$ and $s \in L(\mathcal{G})$.*

Proof: We show this lemma by induction on the number of the rules in $P_{\mathcal{R}}$ that occur in the production $S \xrightarrow[\mathcal{G}_*]{*} t$.

Consider the case that there is no rule in $P_{\mathcal{R}}$ in the production $S^\lambda \xrightarrow[G_*]{*} t$. We have $S^\lambda \xrightarrow[G_*]{*} \langle t\rangle \xrightarrow[G_*]{*} t$. Here we aim at one symbol $\langle a\rangle$ in $\langle t\rangle$. Suppose that $\langle a\rangle$ is produced by the rule in $P_*\backslash(P \cup P_{\mathcal{R}})$. In this case, we can produce $\langle a\rangle$ by the rule of the form $\langle a,q\rangle \to \langle a\rangle$ or $\langle a,q\rangle_{fin} \to \langle a\rangle$. Since the symbols $\langle a,q\rangle$ or $\langle a,q\rangle_{fin}$ is produced from $\langle a\rangle$ by the rules in $P_{\mathcal{A}}$ and P_{fin}, the symbol $\langle a\rangle$ must be produced by P. Moreover, this claim also holds for the symbols of the form $\langle a\rangle^\lambda$. By applying this claim to all symbols in $\langle t\rangle$, we have $S^\lambda \xrightarrow[P']{*} \langle t\rangle$, and hence $S \xrightarrow[P]{*} \langle t\rangle$ from Lemma 155.

Otherwise, let $t = C[C_1[t_1,\ldots,t_n]]$ where $||C|| > 1$, and t is produced as $S^\lambda \xrightarrow[\mathcal{G}_*]{*} \langle C\rangle^\lambda[\langle C_2\rangle[\langle t_1\rangle,\ldots,\langle t_n\rangle]] \xrightarrow[P_{\mathcal{R}}]{} \langle C\rangle^\lambda[\langle C_1\rangle[\langle t_1\rangle,\ldots,\langle t_n\rangle]] \xrightarrow[\mathcal{G}_*]{*} t$. From the construction of $P_{\mathcal{R}}$, we have the rewrite rule $\mathcal{A}_i : C_1[x_1,\ldots,x_m] \to C_2[x_1,\ldots,x_m]$ where $C_1[x_1,\ldots,x_m]\sigma = C_1[t_1,\ldots,t_n]$ and $C_2[x_1,\ldots,x_m]\sigma = C_2[t_1,\ldots,t_n]$ for some σ. Since root symbol in $\langle C_2\rangle$ is of the form $\langle f,q\rangle_{fin}$, we have the transition $C[C_2[t_1,\ldots,t_n]] \xrightarrow[\mathcal{A}_i]{*} C[q] \xrightarrow[\mathcal{A}_i]{*} q_f$ for some $q \in S_i$ and $q_f \in Q_i^f$ from Lemma 157. Hence we have $C[C_2[t_1,\ldots,t_n]] \xrightarrow[\mathcal{R}]{} C[C_1[t_1,\ldots,t_n]]$. Here, we can easily obtain the production $\langle C\rangle^\lambda[\langle C_2\rangle[\langle t_1\rangle,\ldots,\langle t_n\rangle]] \xrightarrow[G_*]{*} C[C_2[t_1,\ldots,t_n]]$. Thus, there exists s such that $s \xrightarrow[\mathcal{R}]{*} C[C_2[t_1,\ldots,t_n]]$ from the induction hypothesis and we have $s \xrightarrow[\mathcal{R}]{*} t$.

If $||C|| = 0$, then root symbol in $\langle C_2\rangle$ is of the form $\langle f,q\rangle_f^\lambda$. In this case, we can prove similary to the previous case. $\square$

Theorem 153 follows from Lemma 159 and Lemma 160.

The following corollary is derived from Theorem 153.

Corollary 161 *Reachability is PSPACE-complete for monotonic CntTRS.*

From Proposition 151, it immediately holds that regular model checking is undecidable for monotonic CntTRS. Moreover, the following lower bounds already hold in the very restricted case of controlled rewrite rules over words, and where each side of every rule has depth exactly one.

Proposition 162 *Reachability is NLINSPACE-complete and regular model checking is undecidable for monotonic and flat CntTRS over unary signatures.*

Proof: We reduce the acceptance (for reachability) and emptiness (for regular model checking) problems for a linear bounded automaton (LBA) $\mathcal{M}$ [22].

Let $\mathcal{M} = \langle P, \Gamma, p_0, G, \Theta\rangle$ be a non-deterministic LBA where P is the set of states, Γ is an input alphabet containing in particular the left and right endmarkers $\|\!:$ and $:\!\|$ are $p_0 \in P$ is the initial state, $G \subseteq P$ is the set of accepting states and Θ is the transition relation in $P \times \Gamma \times P \times \Gamma \times \{\mathsf{left}, \mathsf{right}\}$. The transition relation Θ is such that $\mathcal{M}$ cannot move left from $\|\!:$, cannot move right from $:\!\|$ or print another symbols over $\|\!:$ or $:\!\|$.

Let $\Gamma' = \Gamma \cup \{a^p \mid a \in \Gamma, p \in P\}$ and let us define the unary signature $\mathcal{F} = \Gamma' \cup (\Gamma' \times \Theta) \cup \{\bot\}$, where $\bot$ has arity 0 and every other symbol has arity 1. For the sake of simplicity, a term $a_1(a_2(\ldots a_n(\bot)))$ will be denoted by the string $a_1\, a_2 \ldots a_n$.

Every configuration of $\mathcal{M}$ will be represented by a term of $\mathcal{T}(\mathcal{F})$ of the form $\|\!:x_1 \ldots x_{j-1}x_j^p x_{j+1} \ldots x_n\!:\!\|$, where, for each $1 \le i \le n$, $x_i \in \Gamma \setminus \{\|\!:, :\!\|\}$ is the content of the i^{th} cell of the tape of $\mathcal{M}$, j is the current position of the head of $\mathcal{M}$ and $p \in P$ is the current state. In particular, given an input word $x_1 \ldots x_n \in \big(\Gamma \setminus \{\|\!:, :\!\|\}\big)^*$, the initial configuration of $\mathcal{M}$ is represented by $\|\!:x_1^{p_0} x_2 \ldots x_n\!:\!\|$. We assume indeed without loss of generality that the computation space of $\mathcal{M}$ is limited to the exact size of the input word. We moreover assume without loss of generality that when $\mathcal{M}$ reaches an accepting state, it writes a special blank symbol $\flat \in \Gamma \setminus \{\|\!:, :\!\|\}$ in all the cells of the tape (except the left and right ends), then moves to the leftmost cell, and finally changes its state to a special state p_1 which does not occur in the left hand of any transition of Θ.

To every transition $\theta = \langle p, a, p', b, \mathsf{left}\rangle$ in Θ, we associate the four following monotonic and flat controlled rules

$$\begin{array}{lcrcl}
\|\!:\Gamma^* c\, a^p \Gamma^*\!:\!\| & : & a^p(x) & \to & \langle a^p, \theta\rangle(x)\\
\|\!:\Gamma^* c\, \langle a^p, \theta\rangle \Gamma^*\!:\!\| & : & c(x) & \to & \langle c, \theta\rangle(x)\\
\|\!:\Gamma^* \langle c, \theta\rangle\, \langle a^p, \theta\rangle \Gamma^*\!:\!\| & : & \langle a^p, \theta\rangle(x) & \to & b(x)\\
\|\!:\Gamma^* \langle c, \theta\rangle\, b \Gamma^*\!:\!\| & : & \langle c, \theta\rangle(x) & \to & c^{p'}(x)
\end{array}$$

We use here for SA a simplified notation with regular expressions, assuming that the SA selects the position of the only rewritable letter in the expression.

Similarly, to every transition $\theta = \langle p, a, p', b, \mathsf{right}\rangle$ in Θ, we associate the following controlled rules

$$\begin{array}{lcrcl}
\|\!:\Gamma^* a^p c\, \Gamma^*\!:\!\| & : & a^p(x) & \to & \langle a^p, \theta\rangle(x)\\
\|\!:\Gamma^* \langle a^p, \theta\rangle\, c\, \Gamma^*\!:\!\| & : & c(x) & \to & \langle c, \theta\rangle(x)\\
\|\!:\Gamma^* \langle a^p, \theta\rangle\, \langle c, \theta\rangle \Gamma^*\!:\!\| & : & \langle a^p, \theta\rangle(x) & \to & b(x)\\
\|\!:\Gamma^*\, b\, \langle c, \theta\rangle \Gamma^*\!:\!\| & : & \langle c, \theta\rangle(x) & \to & c^{p'}(x)
\end{array}$$

Let $\mathcal{R}$ be the CntTRS containing all the controlled rewrite rules associated to the transitions of Θ. It is easy to show that $\mathcal{R}$ simulates the moves of $\mathcal{M}$, and that only the correct moves of $\mathcal{M}$ are simulated by $\mathcal{R}$.

Therefore, with the above hypotheses, $\mathcal{M}$ will accept the initial word $x_1 \dots x_n$ iff $\|\!: x_1^{p_0} x_2 \dots x_n\!:\!\| \xrightarrow[\mathcal{R}]{*} \|\!: \flat^{p_1} \underbrace{\flat \dots \flat}_{n-1}\!:\!\|$ and the language of $\mathcal{M}$ is empty iff

$\xrightarrow[\mathcal{R}]{}[\big(\|\!:((\Gamma \setminus \{\|\!:, :\!\|\}) \times P)\, (\Gamma \setminus \{\|\!:, :\!\|\})^* :\!\|\big)] \cap \|\!: \flat^{p_1}\, \flat^* \!:\!\| = \emptyset.$ ☐

Some remarks about the above result. In the above construction, the selection of the rewrite position by the SA is not necessary. Only the selection of the rewritable terms by TA is needed (a weaker condition). Note also that linear and flat TRS preserve regularity, with a PTIME construction of the TA recognizing the closure (see *e.g.* [31]). Hence reachability is decidable in PTIME in the uncontrolled case.

The *conditional grammars* of [6] can be redefined in our settings as (word) grammars whose production rules are CF controlled rewrite rules (and derivations are defined using the controlled rewrite relation). It is shown in [6] that the class of languages of conditional grammars without collapsing rules coincide with CS (word) languages. Hence, it also holds that reachability is PSPACE-complete and regular model checking is undecidable for CF non-collapsing CntTRS over unary signatures.

7.3.2 Prefix control

Some other former results in the case of words imply that the above lower bounds still hold when control is limited to prefix SA. It is shown in [28] that every CS word language can be generated by a CS grammar with production rules of the form $AB \to AC$, $A \to BC$, $A \to a$ (where A, B, C are non-terminal and a is a terminal). It follows that every CS word language is the closure of a constant symbol under a CF non-collapsing pCntTRS (over a unary signature).

Proposition 163 *For all CS tree language L over a unary signature F, there exists a CF non-collapsing pCntTRS $\mathcal{R}$ over $F' \supset F$ such that $L = \mathcal{R}^*(\{c\}) \cap \mathcal{T}(F)$ for some constant $c \in F'_0 \setminus F$.*

Proof: Since L is the language over unary symbols, L can be regarded as a word language. Moreover, we can easily construct a pCntTRS that has the rule $c \to S(\perp)$ where S is the start symbol of the grammar for L and inverse of every production rule. □

Corollary 164 *Reachability is PSPACE-complete and regular model checking undecidable for CF non-collapsing pCntTRS over unary signatures.*

Another consequence of Proposition 163 is that deterministic top-down SA (which are incomparable with prefix SA in general but more general than prefix SA in unary signatures) already capture CS languages, for unary signatures.

To add a final remark, we can observe that in Example 140, there is no hope of regularity preservation even for very simple CF CntTRS containing only flat and monotonic rules, and even restricting to prefix control.

7.3.3 Non-monotonic rewrite rules

It is also shown in [6] that the class of languages of conditional grammars with collapsing rules coincide with recursively enumerable languages. As a consequence, reachability is undecidable for CF CntTRS (with collapsing rules) already in the case of unary signatures. Actually, the following propositions shows that flat (but not monotonic) controlled rewrite rules are sufficient for the simulation of Turing machines.

Proposition 165 *Reachability is undecidable for flat CntTRS over unary signatures.*

Proof: Flat CntTRS is a super class of monotonic and flat CntTRS. We can extend the flat CntTRS $\mathcal{R}$ that simulates the moves of LBA in the proof for Proposition 162, by adding some flat rules of the form $:\| \to \flat:\|$ and $\flat:\| \to :\|$ to $\mathcal{R}$ ($:\|$ denotes the right endmarker), in order to simulate all the moves of a TM. □

Note that when the signature is unary, all the rewrite rules are necessarily linear. Again, this result is in contrast with the case of uncontrolled rewriting, because linear and flat TRS preserves regularity, and hence have a decidable reachability problem. Restricting the control to prefix permits to obtain a decidability result for non-monotonic rewrite rules, as long as they are not collapsing.

Theorem 166 *Reachability is decidable in PSPACE for flat non-collapsing pCntTRS over unary signatures.*

Proof: We show the following claim: u rewrites to v iff $u = u_0 \xrightarrow[\mathcal{R}]{} u_1 \xrightarrow[\mathcal{R}]{} \cdots \xrightarrow[\mathcal{R}]{} u_k = v$ with $\|u_0\|, \ldots, \|u_k\| \leq max(\|u\|, \|v\|)$.

The "if" direction is trivial. For the "only if" direction, assume given a reduction $u = w_1 \xrightarrow[\mathcal{R}]{*} w_n = v$, and let $max(\|u\|, \|v\|) = M$. We make an induction on the number of strings w_i longer than M. Suppose that the reduction contains one w_i such that $||w_i|| > M$. Then there exists a sub-sequence $w_k \xrightarrow[\mathcal{R}]{+} w_m$ such that $\|w_k\| = \|w_m\| = M$, with $k < m$. It holds that $w_k = w_k[\perp]_M \xrightarrow[\mathcal{R}]{*} w_m[\perp]_M = w_m$ because we consider only prefix control. This reduces the number of string w_i longer than M. □

Non-monotonicity is also a source of undecidability of reachability for CntTRS even in the case of ground controlled rewrite rules.

Proposition 167 *Reachability is undecidable for ground CntTRS.*

Proof: By representing the words $a_1, \ldots, a_n$ as right combs $f(a_1, f(\cdots f(a_n, \bot)))$, we can construct a ground CntTRS that simulates the moves of Turing Machine. Like in the proof of Propositions 165 and 162, one move of the of the TM is simulated by several rewrite steps, controlling the context left or right of the current position of the TM's head. Here, the controlled rewrite rule will have the form $\mathcal{A} : a \to a'$ were a and a' are constant symbols, and $\mathcal{A}$ controls c and d in a configuration $f(\ldots f(c, f(a, f(d(, \ldots)))))$, where a is at the rewrite position. $\Box$

This result is in contrast to uncontrolled ground TRS, for which reachability is decidable in PTIME.

7.3.4 Recursive Prefix Control

We propose a relaxed form of control, where, in order to select the positions of application of a controlled rewrite rule, the term to be rewritten is tested for membership in the closure of a regular language L, instead of membership to L directly. The idea is somehow similar to conditional rewriting (see *e.g.* [1]) where the conditions are equations that have to be solved by the rewrite system.

The definition is restricted to control with prefix SA, and a recursive pCntTRS $\mathcal{R}$ is defined as a pCntTRS. In order to define formally the rewrite relation, let us recall first that in the computations of a prefix SA $\mathcal{A}$, the states below a selection state q_s are universal (q_0), *i.e.* we can have any subterm at a selected position (only the part of the term *above* the selected position matters). Following this observation, we say that the variable position p in a context $C[x_1]$ over $\mathcal{F}$ is selected by the prefix SA $\mathcal{A}$ if p is selected in $C[c]$ where c is an arbitrary symbol of $\mathcal{F}_0$. A term s rewrites to t in one step by a recursive pCntTRS $\mathcal{R}$, denoted by $s \xrightarrow[\mathcal{R}]{} t$, if there exists a controlled rewrite rule $\mathcal{A} : \ell \to r \in \mathcal{R}$, where $\mathcal{A}$ is a prefix SA, a substitution σ, a position $p \in \mathrm{Pos}(s)$, and a context $C[x_1]$ such that $C[x_1] \xrightarrow[\mathcal{R}]{*} s[x_1]_p$ and the position of x_1 is selected in $C[x_1]$ by $\mathcal{A}$, such that $s|_p = \ell\sigma$ and $t = s[r\sigma]_p$.

Example 168 Let $\mathcal{F} = \{a, b, c, d, \perp\}$ be a unary signature, and let $\mathcal{R}$ be the flat recursive pCntTRS containing the rules $\mathcal{C}_1 : a(a(x)) \to b(x)$, and $\mathcal{C}_2 : c(x) \to d(x)$, where the SA $\mathcal{C}_1$ selects the position after a prefix aa, and $\mathcal{C}_2$ selects the position after a prefix $aaaa$. Then we have with $\mathcal{R}$ (we omit the parentheses and the tail $\perp$, and underline the part of the term which is rewritten)

$$aa\underline{aa}c \xrightarrow[\mathcal{R}]{} aab\underline{c} \xrightarrow[\mathcal{R}]{} aabd$$

Note that for the last step, we have use the fact that $aaaa \xrightarrow[\mathcal{R}]{} aab$, *i.e.* there exists $C[x_1] = aaaa(x_1)$ with $C[x_1] \xrightarrow[\mathcal{R}]{} aab(x_1)$ and the position of x_1 in $C[x_1]$ is selected by $\mathcal{C}_2$. The last rewrite step would not be possible if $\mathcal{R}$ would not be recursive, because aab is not a prefix admitted by $\mathcal{C}_2$.

Theorem 169 *Regular model-checking is decidable in EXPTIME for linear and right-shallow recursive pCntTRS.*

Proof: Given a right-shallow and linear recursive pCntTRS $\mathcal{R}$ and the language $L \subseteq \mathcal{T}(\mathcal{F})$ of a TA $\mathcal{A}_L = \langle Q_L, F_L, \Delta_L \rangle$, the proof is the construction of an alternating tree automata with ε-transitions (ε-ATA) $\mathcal{A}'$ recognizing the rewrite closure $\xrightarrow[\mathcal{R}]{}[L]$.

Intuitively, an alternating tree automata $\mathcal{A}$ is a top-down tree automaton that can spawn in several copies during computation on a term t. At every computation step, there are several copies of $\mathcal{A}$ in different positions of t, each copy in its own state. Initially, there is one copy of $\mathcal{A}$ in its initial state at the root of t. Then, the copies can be propagated down to the leaves step by step.

Formally, an *alternating TA* with ε-transitions (ε-ATA) over a signature $\mathcal{F}$ is a tuple $\mathcal{A} = \langle Q, q_0, \delta \rangle$ where Q is a finite set of states, $q_0 \in Q$ is the initial state and δ is a function which associates to every state $q \in Q$ a disjunction of conjunctions of propositional predicates of the following form

- $a \in \mathcal{F}$,
- $\langle q', \varepsilon \rangle$, for $q' \in Q \setminus \{q\}$,
- $\langle q', i \rangle$, for $q' \in Q$ and $1 \leq i \leq m$ where m is the maximal arity of a symbol in $\mathcal{F}$.

We say that a set P of predicates as above satisfies a disjunction of conjunctions D, denoted by $P \models D$, if either D is the empty disjunction or $D = C_1 \vee \ldots \vee C_k$, for $k \geq 1$, and there exists one j, $1 \leq j \leq k$, such that every propositional predicate in the conjunction C_j belongs to P.

A *run* of $\mathcal{A}$ on $t \in \mathcal{T}(\mathcal{F})$ is a function ρ from $\mathrm{Pos}(t)$ into 2^Q such that for all position $p \in \mathrm{Pos}(t)$, with $t(p) = a \in \mathcal{F}_n$, $n \geq 0$, and for all state $q \in \rho(p)$, it holds that

$$a, \langle \rho(p.1), 1\rangle, \ldots, \langle \rho(p.n), n\rangle, \langle \rho(p), \varepsilon\rangle \models \delta(q)$$

where $\langle S, p\rangle$ is a notation for all the predicates $\langle q, p\rangle$ with $q \in S$.

The language $\mathcal{L}(\mathcal{A})$ of $\mathcal{A}$ is the set of terms $t \in \mathcal{T}(\mathcal{F})$ on which there exists a run ρ of $\mathcal{A}$ such that $q_0 \in \rho(\varepsilon)$ (terms *recognized* by $\mathcal{A}$).

Given an ε-ATA $\mathcal{A} = \langle Q, q_0, \delta\rangle$, one can build a TA $\mathcal{A}' = \langle 2^Q, \{S \subseteq Q \mid q_0 \in S\}, \Delta\rangle$ where Δ contains all the TA transitions of the form $a(S_1, \ldots, S_n) \to S$, for $a \in \mathcal{F}_n$, $S_1, \ldots, S_n, S \subseteq Q$ such that for all $q \in S$, $a, \langle S_1, 1\rangle, \ldots, \langle S_n, n\rangle, \langle S, \varepsilon\rangle \models \delta(q)$. With these definitions, it is easy to see that every run of $\mathcal{A}$ on t is also a run of $\mathcal{A}'$ on t, and reciprocally, hence $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.

On the other hand, given a TA $\mathcal{A} = \langle Q, F, \Delta\rangle$, one can construct an ε-ATA $\mathcal{A}' = \langle Q \cup \{q_0\}, q_0, \delta\rangle$, with $q_0 \notin Q$, $\delta(q_0) = \bigvee_{q \in F} \langle q, \varepsilon\rangle$, and for all $q \in Q$,

$$\delta(q) = \bigvee_{a(q_1,\ldots,q_n) \to q \in \Delta} (a \wedge \langle q_1, 1\rangle \wedge \ldots \wedge \langle q_n, n\rangle)$$

We will now construct an ε-ATA $\mathcal{A}'$ recognizing the rewrite closure $\xrightarrow[\mathcal{R}]{}[L]$. Let $sr(\mathcal{R})$ be the set of ground direct subterms of the *rhs* of rules of $\mathcal{R}$. For each $r \in sr(\mathcal{R})$, we construct a TA $\mathcal{A}_r = \langle Q_r, F_r, \Delta_r\rangle$ such that $\mathcal{L}(\mathcal{A}_r) = \{r\}$. It is defined by $Q_r = \mathrm{Pos}(r)$, $F_r = \{\varepsilon\}$ and $\Delta = \{a(p.1, \ldots, p.n) \to p \mid p \in \mathrm{Pos}(r), r(p) = a \in \mathcal{F}_n\}$.

Let $\mathcal{A}'_r$ be the ε-ATA associated to $\mathcal{A}_r$ as above. Let $\mathcal{A}'_L = \langle Q_L, q_0, \delta_L\rangle$ be the ε-ATA associated to $\mathcal{A}_L$ and for each controlled rule $\mathcal{C} : \ell \to r$ of $\mathcal{R}$, let $\mathcal{C}'$ be the ε-ATA associated to $\mathcal{C}$. We assume *wlog* that all the automata defined above have disjoint state sets. We will construct incrementally a finite sequence of ε-ATA $\mathcal{A}'_0, \ldots, \mathcal{A}'_k$ whose last element $\mathcal{A}'_k$ recognizes $\xrightarrow[\mathcal{R}]{}[L]$.

The first ε-ATA of the sequence, $\mathcal{A}'_0 = \langle Q_0, q_0, \delta_0\rangle$ is defined as the disjoint union of the ε-ATAs $\mathcal{A}'_L$, all the $\mathcal{A}'_r$ for $r \in sr(\mathcal{R})$, and all $\mathcal{C}'$, for $\mathcal{C}$ prefix SA controlling a rule of $\mathcal{R}$. Every other $\mathcal{A}'_i$ in this sequence will be $\langle Q_0, q_0, \delta_i\rangle$, where the transition function δ_i is defined as follows.

We introduce a notation that will be useful for the construction of the δ_i. Let $t = a(t_1, \ldots, t_n)$ be a ground term of $\mathcal{T}(\mathcal{F} \cup Q)$, where $n \geq 0$ and the symbols of Q are assumed of arity 0. We write $t \models_i q$ if we are in one of the following cases

$\delta_i(q) = \phi_1 \vee \phi_2$ and $t \models_i \phi_1$ or $t \models_i \phi_2$

$\delta_i(q) = \phi_1 \wedge \phi_2$ and $t \models_i \phi_1$ and $t \models_i \phi_2$

$\delta_i(q) = \langle q', \varepsilon\rangle$ and $t \models_i q'$

$\delta_i(q) = \langle q', j\rangle$, with $1 \leq j \leq n$ and $t_j \models_i q'$

$\delta_i(q) = a$.

Assume that we have constructed all the functions up to δ_i, and let us define δ_{i+1}. If there is a linear controlled rule $\mathcal{C} : \ell \to b(r_1, \ldots, r_m)$ in $\mathcal{R}$, where every r_i is either a variable or a ground term of $\mathcal{T}(\mathcal{F})$, and a substitution σ from $\mathcal{X}$ into Q grounding for ℓ, for all state $q \in Q$ such that $\ell\sigma \models_i q$, let

$$\delta_{i+1}(q) = \delta_i(q) \vee \big(b \wedge \bigwedge_{i \in N} \langle q_{r_i}, i\rangle \wedge \bigwedge_{j \in V} \langle r_j\sigma, j\rangle \wedge \langle q_s, \varepsilon\rangle\big),\ \delta_{i+1}(q') = \delta_i(q') \text{ for all } q' \neq q$$

where $N = \{i \leq m \mid r_i \in \mathcal{T}(\mathcal{F})\}$, $V = \{j \leq m \mid r_j \in \mathcal{X}\}$, and q_s is the unique selection state of the ε-ATA $\mathcal{C}'$ associated to $\mathcal{C}$.

If there is a linear controlled rule $\mathcal{C} : \ell \to x$ in $\mathcal{R}$, where $x \in X$, and a substitution σ from $\mathcal{X}$ into Q grounding for ℓ and such that $\ell\sigma \models_i q$ then let

$$\delta_{i+1}(q) = \delta_i(q) \vee \langle q_s, \varepsilon\rangle, \quad \delta_{i+1}(q') = \delta_i(q') \text{ for all } q' \neq q$$

where q_s is the unique selection state of the ε-ATA $\mathcal{C}'$ associated to $\mathcal{C}$.

The number of conjunctions that can be added to a $\delta_i(q)$ in the above construction is bounded. Assuming that we do not add twice the same conjunction, the process will terminate with a fixpoint $\mathcal{A}'_k = \mathcal{A}'$. The size of $\mathcal{A}'$ is polynomial in the sizes of $\mathcal{A}_L$ and $\mathcal{R}$. It can be shown that $\mathcal{L}(\mathcal{A}') \subseteq \xrightarrow[\mathcal{R}]{}[L]$ by induction on the multiset of indexes i of the transition functions δ_i used in a run of $\mathcal{A}'$ on a term, and that $\mathcal{L}(\mathcal{A}') \supseteq \xrightarrow[\mathcal{R}]{}[L]$ by induction on the length of a rewrite sequence.

It follows that there exists a TA $\mathcal{A}$ whose size is exponential in the size of $\mathcal{A}'$ and such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}') = \xrightarrow[\mathcal{R}]{}[L]$, and hence we can decide the problem of regular model checking for $\mathcal{R}$ in exponential time. $\square$

Example 170 Let us come back the recursive pCntTRS $\mathcal{R}$ of Example 168. The transitions of the prefix SA $\mathcal{C}_1$ and $\mathcal{C}_2$ for control in R are explicitly the following (their final states are respectively q_2^1 and q_4^2)

$$\begin{aligned}
\mathcal{C}_1 :\ & \perp \to q_s^1 \mid q_0^1, f(q_0^1) \to q_s^1 \mid q_0^1, a(q_s^1) \to q_1^1, a(q_1^1) \to q_2^1 \quad (f \in \{a,b,c,d\}) \\
\mathcal{C}_2 :\ & \perp \to q_s^2 \mid q_0^1, f(q_0^2) \to q_s^2 \mid q_0^2, a(q_s^2) \to q_1^2, a(q_1^1) \to q_2^2, a(q_2^2) \to q_3^2, a(q_3^2) \to q_4^2
\end{aligned}$$

The associated ε-ATA $\mathcal{C}_1'$ and $\mathcal{C}_2'$ have the following transition functions

$$\begin{aligned}
\delta^1 :\ & q_2^1 \mapsto a \wedge \langle q_1^1, 1\rangle, q_1^1 \mapsto a \wedge \langle q_s^1, 1\rangle, q_s^1 \mapsto \perp \vee \bigvee_{f=a,b,c,d} (f \wedge \langle q_0^1, 1\rangle), \\
& q_0^1 \mapsto \perp \vee \bigvee_{f=a,b,c,d} (f \wedge \langle q_0^1, 1\rangle) \\
\delta^2 :\ & q_4^2 \mapsto a \wedge \langle q_3^2, 1\rangle, q_3^2 \mapsto a \wedge \langle q_2^2, 1\rangle, q_2^2 \mapsto a \wedge \langle q_1^2, 1\rangle, q_1^2 \mapsto a \wedge \langle q_s^2, 1\rangle, \\
& q_s^2 \mapsto \perp \vee \bigvee_{f=a,b,c,d} (f \wedge \langle q_0^2, 1\rangle),\ q_0^2 \mapsto \perp \vee \bigvee_{f=a,b,c,d} (f \wedge \langle q_0^2, 1\rangle)
\end{aligned}$$

Let us consider the initial language $L = \{aaaac\}$, recognized by the ε-ATA $\mathcal{A}_L'$ with the following transition function

$$\begin{aligned}
\delta_L :\ & q_5 \mapsto a \wedge \langle q_4, 1\rangle, q_4 \mapsto a \wedge \langle q_3, 1\rangle, q_3 \mapsto a \wedge \langle q_2, 1\rangle, q_2 \mapsto a \wedge \langle q_1, 1\rangle, q_1 \mapsto \\
& c \wedge \langle q_0, 1\rangle, q_0 \mapsto \perp.
\end{aligned}$$

The transition function δ_0 in the above construction is the union of δ^1, δ^2 and δ_L. It holds that $a(a(q_1)) \models_0 q_3$. Hence, with the first rule of $\mathcal{R}$, we can let

$$\delta_1(q_3) = \delta_0(q_3) \vee \big(b \wedge \langle q_1, 1\rangle \wedge \langle q_s^1, \varepsilon\rangle\big)$$

Similarly, $a(a(q_s^2)) \models_1 q_2^2$, and we can let $\delta_2(q_2^2) = \delta_0(q_2^2) \vee \big(b \wedge \langle q_s^2, 1\rangle \wedge \langle q_s^1, \varepsilon\rangle\big)$. Moreover, $c(q_0) \models_2 q_1$, and with the second rule of $\mathcal{R}$, we can let

$$\delta_3(q_1) = \delta_0(q_1) \vee \big(d \wedge \langle q_0, 1\rangle \wedge \langle q_s^2, \varepsilon\rangle\big)$$

With these transitions functions, we have the following runs for 2 descendants of L

$$
\begin{array}{ccccc}
a & a & b & c & \perp \\
\left\{\begin{array}{c} q_5 \\ q_2^1 \end{array}\right\} &
\left\{\begin{array}{c} q_4 \\ q_1^1 \end{array}\right\} &
\left\{\begin{array}{c} q_3 \\ q_s^1 \end{array}\right\} &
\left\{\begin{array}{c} q_1 \\ q_0^1 \end{array}\right\} &
\left\{\begin{array}{c} q_0 \\ q_0^1 \end{array}\right\}
\end{array}
$$

$$
\begin{array}{ccccc}
a & a & b & d & \perp \\
\left\{\begin{array}{c} q_5 \\ q_4^2 \\ q_2^1 \end{array}\right\} &
\left\{\begin{array}{c} q_4 \\ q_3^2 \\ q_1^1 \end{array}\right\} &
\left\{\begin{array}{c} q_3 \\ q_2^2 \\ q_s^1 \end{array}\right\} &
\left\{\begin{array}{c} q_1 \\ q_s^2 \\ q_0^1 \end{array}\right\} &
\left\{\begin{array}{c} q_0 \\ q_0^2 \\ q_0^1 \end{array}\right\}
\end{array}
$$

Chapter 8

Conclusion

From Chapter 3 to Chapter 6, we have shown decidability and undecidability of reachability for some classes with respect to the reduction strategies as shown in Figure 1.1 and Table 1. Moreover, for linear right-shallow TRSs, we showed effective preservation of regularity with respect to the context-sensitive reduction and the context-sensitive innermost reduction.

In Chapter 7, we formalized CntTRSs and showed some results on complexity.

A future work is to find other subclasses such that reachability is decidable or undecidable. One of the candidates is right-linear right-shallow TRSs with respect to the innermost reduction. In the class, reachability is decidable with respect to the ordinary reduction and the context-sensitive reduction, but it is open with respect to the innermost reduction. Besides the classes whose reachability is decidable with respect to the outermost reduction, are little known. Therefore, there are many open problems for the outermost reduction.

Improvement of complexity of reachability checking is another future work. Output automata of the algorithms in this paper except for $\mathrm{P}_{\mathrm{cs}}^{\mathrm{lrs}}$ have no less than exponential number of states. Therefore, these algorithms take exponential time. We think that it may be possible to construct a non-deterministic TA $\mathcal{A}_{\mathrm{NF}}$ that accepts normal forms, and the complexity of algorithms for the context-sensitive innermost reduction may be improved by using it instead of deterministic one. However, determinacy of $\mathcal{A}_{\mathrm{NF}}$ is necessary to proofs in this paper. Hence it does not seem to be easy to improve complexity.

For CntTRSs, there are many future works. The proof of undecidability for ground CntTRS (Proposition 167) does not work when restricted to prefix control. It could be interesting to know whether reachability is decidable for ground pCntTRS. We are also interested in knowing how the decidability result of Theorem 166 can be generalized to non-unary signatures.

In [6], the conditional grammars (*i.e.* controlled (in the above sense) context-free word grammars) are related to grammars with a restriction on the possible production sequences (the list of names of production rules used must belong to a regular language). It could be interesting to establish a similar comparison for term rewriting. In particular, results on the restriction defined by authorized sequences of rewrite rules could be useful for the analysis of languages for the extensional specification of the set of possible rewrite derivations like in [2].

Rewriting of unranked ordered labeled tree has been much less studied than its counterpart for ranked terms. We would like to study controlled rewriting in this case, in particular in the context of update rules for XML [4, 20].

Bibliography

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

[2] P. Borovanský, C. Kirchner, H. Kirchner, and C. Ringeissen. *Rewriting with strategies in* ELAN*: a functional semantics.* International Journal of Foundations of Computer Science, vol. 12, pp. 69–98, 2001.

[3] J. Chabin and P. Réty. *Visibly pushdown languages and term rewriting.* In B. Konev and F. Wolter, editors, 6th International Symposium on Frontiers of Combining Systems, (FroCoS 2007), vol. 4720 of Lecture Notes in Computer Science, pp 252–266, 2007.

[4] D. Chamberlin and J. Robie. *Xquery update facility 1.0.* Available on: http://www.w3.org/TR/xquery-update-10/, 2009.

[5] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. *Tree automata techniques and applications.* Available on: http://tata.gforge.inria.fr/, 2008.

[6] J. Dassow, G. Păun, and A. Salomaa. *Grammars with controlled derivations*, pages 101–154. Springer-Verlag New York, Inc., 1997.

[7] M. Dauchet, T. Heuillard, P. Lescanne, and S. Tison. *Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems.* Information and Computation, vol. 88, pp. 187–201, 1990.

[8] I. Durand and G. Sénizergues. *Bottom-up rewriting is inverse recognizability preserving.* In F. Baader, editor, 18th International Conference on

Term Rewriting and Applications, (RTA 2007), vol. 4533 of Lecture Notes in Computer Science, pp. 107–121, 2007.

[9] M. Frick, M. Grohe, and C. Koch. *Query evaluation on compressed trees.* 18th IEEE Symposium on Logic in Computer Science, (LICS 2003), 2003.

[10] A. Gascón, G. Godoy, and F. Jacquemard. *Closure of tree automata languages under innermost rewriting.* Electronic Notes in Theoretical Computer Science, vol. 237, pp. 23–38, 2009.

[11] T. Genet and F. Klay. *Rewriting for cryptographic protocol verification.* In D. A. McAllester, editor, 17th International Conference on Automated Deduction, (CADE-17), vol. 1831 of Lecture Notes in Computer Science, pp. 271–290, 2000.

[12] R. Gilleron and S. Tison. *Regular tree languages and rewrite systems.* Fundamenta Informcae, vol. 24, pp. 157–174, 1995.

[13] G. Godoy and E. Huntingford. *Innermost-reachability and innermost-joinability are decidable for shallow term rewrite systems.* In F. Baader, editor, 18th International Conference on Term Rewriting and Applications, (RTA 2007), vol. 4533 of Lecture Notes in Computer Science, pp. 184–199, 2007.

[14] G. Gottlob and C. Koch. *Monadic datalog and the expressive power of languages for web information extraction.* Journal of the ACM, vol. 51, pp. 74–113, 2004.

[15] I. Guessarian. *Pushdown tree automata.* Mathematical Systems Theory, vol. 16, pp. 237–263, 1983.

[16] D. Hofbauer and J. Waldmann. *Deleting string rewriting systems preserve regularity.* Theoretical Computer Science, vol. 327, pp. 301–317, 2004.

[17] Mitsuhashi I., M. Oyamaguchi, and F. Jacquemard. *The confluence problem for flat TRSs.* In J. Calmet, T. Ida, and D. Wang, editors, 8th International Conference on Artificial Intelligence and Symbolic Computation,

(AISC 2006), vol. 4120 of Lecture Notes in Artificial Intelligence, pp. 68–81, 2006.

[18] F. Jacquemard. *Decidable approximations of term rewriting systems.* In H. Ganzinger, editor, 7th International Conference on Rewriting Techniques and Applications, (RTA 1996), vol. 1103 of Lecture Notes in Computer Science, pp. 362–376, 1996.

[19] F. Jacquemard. *Reachability and confluence are undecidable for flat term rewriting systems.* Information Processing Letters, vol. 87, pp.265–270, 2003.

[20] F. Jacquemard and M. Rusinowitch. *Rewrite-based verification of xml updates.* In T. Kutsia, W. Schreiner, and M. Fernández, editors, 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming, (PPDP 2010), pp. 119–130. 2010.

[21] Y. Kojima and M. Sakai. *Innermost reachability and context sensitive reachability properties are decidable for linear right-shallow term rewriting systems.* In A. Voronkov, editor, 19th International Conference on Rewriting Techniques and Applications, (RTA 2008), vol. 5117 of Lecture Notes in Computer Science, pp. 187–201, 2008.

[22] S.-Y. Kuroda. *Classes of languages and linear-bounded automata.* Information and Control, vol. 7, pp. 207–223, 1964.

[23] S. Lucas. *Context-sensitive computations in functional and functional logic programs.* Journal of Functional and Logic Programming, 1998.

[24] T. Nagaya and Y. Toyama. *Decidability for left-linaer growing term rewriting systems.* In P. Narendran and M. Rusinowitch, editors, 10th International Conference on Rewriting Techniques and Applications, (RTA 1999), vol. 1631 of Lecture Notes in Computer Science, pp. 256–270, 1999.

[25] F. Neven and T. Schwentick. *Query automata over finite trees.* Theoretical Computer Science, vol. 275, pp. 633–674, 2002.

[26] M. Oyamaguchi. *The reachability problem for quasi-ground term rewriting systems.* Journal of information processing, vol. 9, pp. 232–236, 1986.

[27] M. Oyamaguchi. *The reachability and joinability problems for right-ground term-rewriting systems.* Journal of information processing, vol. 13, pp. 347–354, 1990.

[28] M. Penttonen. *One-sided and two-sided context in formal grammars.* Information and Control, vol. 25, pp. 371–392, 1974.

[29] E. L. Post. *A variant of a recursively unsolvable problem.* Bulletin of The American Mathematical Society, vol. 52, pp. 264–269, 1946.

[30] P. Réty and J. Vuotto. *Tree automata for rewrite strategies.* Journal of Symbolic Computation, vol. 40, pp. 749–794, 2005.

[31] K. Salomaa. *Deterministic tree pushdown automata and monadic tree rewriting systems.* Journal of Computer and System Sciences, vol. 37, pp. 367–394, 1988.

[32] T. Takai, Y. Kaji, and H. Seki. *Right-linear finite path overlapping term rewriting systems effectively preserve recognizability.* In L. Bachmair, editor, 11th International Conference on Rewriting Techniques and Applications, (RTA 2000), vol. 1833 of Lecture Notes in Computer Science, pp. 246–260, 2000.

[33] T. Takai, H. Seki, Y. Fujinaka, and Y. Kaji. *Layered transducing term rewriting system and its recognizability preserving property.* IEICE transactions on information and systems, vol. 86, pp. 285–295, 2003.

[34] Togashi,A. and Noguchi,S. *Some decision problems and their time complexity for term rewriting systems.* IEICE transactions on information and systems, vol. 66, pp. 1177–1184, 1983.

[35] R. Verma. *New undecidability results for properies of term rewrite systems.* 9th International workshop on Rule-Based Programming, (RULE 2008), pp. 1–16, 2008.

Publications

Journal papers

[1] Yoshiharu Kojima, Masahiko Sakai, Naoki Nishida, Keiichirou Kusakari, and Toshiki Sakabe. *Context-Sensitive Innermost Reachability is Decidable for Linear Right-Shallow Term Rewriting Systems.* IPSJ Transactions on Programming, vol. 2, pp. 20–32, 2009.

[2] Yoshiharu Kojima, Masahiko Sakai, Naoki Nishida, Keiichirou Kusakari, and Toshiki Sakabe. *Decidability of Reachability for Right-Shallow Context-Sensitive Term Rewriting Systems.* IPSJ Transactions on Programming, vol. 4, pp. 12–35, 2011.

Conferences (refereed)

[1] Yoshiharu Kojima, Masahiko Sakai. *Innermost Reachability and Context-Sensitive Reachability Properties are Decidable for Linear Right-Shallow Term Rewriting Systems.* 19th International Conference on Rewriting Techniques and Applications, (RTA 2008), vol. 5117 of Lecture Notes in Computer Sciences, pp. 187–201, 2008.

[2] Frolent Jacquemard, Yoshiharu Kojima, and Masahiko Sakai. *Controlled Term Rewriting.* Proceedings of 8th International Symposium Frontiers of Combining Systems, (FroCoS 2011), vol. 6989 of Lecture Notes in Artificial Intelligience, pp.179–194, 2011.

Others

[1] Yoshiharu Kojima, Masahiko Sakai, Naoki Nishida, Keiichirou Kusakari, and Toshiki Sakabe. *Context-Sensitive Innermost Reduction for Linear Right-Shallow Term Rewriting Systems Effectively Preserves Regularity.* LA Symposium 2009 Summer, pp.14-1–14-8, Japan, 2009.

[2] Yoshiharu Kojima, Masahiko Sakai, Naoki Nishida, Keiichirou Kusakari, and Toshiki Sakabe. *Decidability and Undecidability of Reachability for Right-Linear Left-Shallow Non-Erasing Term Rewriting Systems.* LA Symposium 2011 Summer, pp. 15-1–15-10, Japan, 2011.