Decidability of Reachability for Right-shallow Context-sensitive Term Rewriting Systems

YOSHIHARU KOJIMA,†,‡ MASAHIKO SAKAI,† NAOKI NISHIDA,† KEICHIROU KUSAKARI† and TOSHIKI SAKABE†

The reachability problem for an initial term, a goal term, and a rewrite system is to decide whether the initial term is reachable to goal one by the rewrite system or not. The innermost reachability problem is to decide whether the initial term is reachable to goal one by innermost reductions of the rewrite system or not. A context-sensitive term rewriting system (CS-TRS) is a pair of a term rewriting system and a mapping that specifies arguments of function symbols and determines rewritable positions of terms. In this paper, we show that both reachability for right-linear right-shallow CS-TRSs and innermost reachability for shallow CS-TRSs are decidable. We prove these claims by presenting algorithms to construct a tree automaton accepting the set of terms reachable from a given term by (innermost) reductions of a given CS-TRS.

1. Introduction

The reachability problem for two given terms $s$, $t$, and a reduction of a rewrite system $R$ is to decide whether $s$ is reachable to $t$ by the reduction of $R$ or not. Decision procedures of the problem for ordinary reductions of term rewriting systems (TRSs) are applicable to security protocol verification and to solving other problems of TRSs. Since it is known that this problem is undecidable for general TRSs, efforts have been made to find subclasses of TRSs in which the reachability is decidable or undecidable (1,4,5,9,13-18), as shown in Fig. 1.

A context-sensitive TRS (CS-TRS) is a pair of TRSs and a mapping from a function symbol to a set of natural numbers, where the mapping is used to specify that arguments are rewritable or not. CS-TRS is used in evaluating $\text{if}$

\[
\begin{array}{ll}
\text{\ldots then \ldots else \ldots or case structures.}
\end{array}
\]

We have already shown that reachability is decidable for linear right-shallow CS-TRSs (10). However, linear right-shallow is not a large enough class to express practical programs (e.g., multiplication).

In this paper, we show that reachability is decidable for right-linear right-shallow CS-TRSs. Right-linear right-shallow, however, is still not enough to express practical programs precisely, but we can express programs closer to the precise one.

Innermost reduction is a strategy that rewrites innermost redexes, and is known as good at representing call-by-value computation widely used in most programming languages. Therefore, the languages that adopt call-by-value computation and $\text{if \ldots then \ldots else \ldots}$ structures (e.g., C languages) have computation models defined by the innermost reduction of CS-TRSs. For innermost reduction of TRSs and CS-TRSs, some decidable classes of reachability are known (6,8,10,11).

However, these classes do not have a large enough similarly to the case of the ordinary reduction of CS-TRSs.
In this paper, we also show that reachability for innermost reduction (innermost reachability) is decidable for shallow CS-TRSs.

We show the results of this paper by presenting two algorithms. The first algorithm constructs a tree automaton (TA) recognizing the set of terms reachable from a given term and a given right-linear right-shallow CS-TRS. This algorithm is based on the procedure for linear right-shallow CS-TRSs \(^{10}\), and we introduce the idea in Refs. 14) and 17) to adopt the procedure to left-non-linear CS-TRSs as well. The second algorithm constructs a tree automaton with constraints between brothers (TACBB) that recognize the set of terms innermost reachable from a given term and a shallow CS-TRS. The second algorithm is achieved by introducing the TACBB \(\mathcal{A}_{\text{NF}}\) that accepts the set of normal forms to check whether each subterm is rewritable or unre rewritable

2. Preliminary

We use the usual notations of rewrite systems \(^3\) and tree automata \(^5\). Let \(F\) be a set of function symbols with fixed arity and \(X\) be an enumerable set of variables. The arity of function symbol \(f\) is denoted by \(\text{ar}(f)\). Function symbols with \(\text{ar}(f) = 0\) are constants. The set of terms, defined in the usual way, is denoted by \(T(F, X)\). A term is linear if no variable occurs more than once in the term. The set of variables occurring in \(t\) is denoted by \(\text{Var}(t)\). A term \(t\) is ground if \(\text{Var}(t) = \emptyset\). The set of ground terms is denoted by \(T(F)\).

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 \(\text{Pos}(t)\), where the empty sequence \(\varepsilon\) is used to denote root position. The depth of a position \(p\) is defined as the length of \(p\) and denoted as \(|p|\). The height \(|t|\) of a term \(t\) is defined as \(\max(|p| : p \in \text{Pos}(t))\). A term \(t\) is shallow if the depths of variable occurrences in \(t\) are all 0 or 1. The subterm of \(t\) at position \(p\) is denoted by \(t|_{p}\), and \(t|_{p'}\) represents the term obtained from \(t\) by replacing the subterm \(t|_{p}\) by \(t'\). If a term \(s\) is a subterm of \(t\) and \(s \neq t\), \(s\) is a proper subterm of \(t\). We denote \(s \sqsubseteq t\) (\(s \sqsubset t\)) such that a term \(s\) is a (proper) subterm of \(t\). A context \(C\) is a term that contains the symbol \(\square\), and \(C|_{t|_{p}}\) represents the term obtained by replacing \(\square\) in the position \(p\) of \(C\) by \(t\).

A substitution \(\sigma\) is a mapping from \(X\) to \(T(F, X)\) whose domain \(\text{Dom}(\sigma) = \{x \in X : x \neq \sigma(x)\}\) is finite. The term obtained by applying a substitution \(\sigma\) 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 \(T(F, X)\), written as \(l \rightarrow r\), such that \(l \notin X\) and \(\text{Var}(l) \supseteq \text{Var}(r)\). We say that variables in \(\text{Var}(l) \setminus \text{Var}(r)\) are erasing. A term rewriting system (over \(F\)) (TRS) is a finite set of rewrite rules. Rewrite relation \(\frac{R}{\text{sf}}\) induced by a TRS \(R\) is as follows: \(s \frac{R}{\text{sf}} t\) if and only if \(s = s[l|_{p}σ]_p\) and \(t = s[r|_{p}σ]_p\) for some rule \(l \rightarrow r \in R\), with substitution \(σ\) and position \(p \in \text{Pos}(s)\). We call \(l\sigma\) a redex. We sometimes write \(\frac{R}{\text{sf}}\) by presenting the position \(p\) explicitly.

A rewrite rule \(l \rightarrow r\) is left-linear (resp. right-linear, linear, right-shallow, shallow) if \(l\) is linear (resp. \(r\) is linear, \(l\) and \(r\) are linear, \(r\) is shallow, \(l\) and \(r\) are shallow). A TRS \(R\) is left-linear (resp. right-linear, linear, right-shallow, shallow) if every rule in \(R\) is left-linear (resp. right-linear, linear, right-shallow, shallow).

Let \(\rightarrow\) be a binary relation on a set \(T(F)\). We say \(s \in T(F)\) is a normal form (with respect to \(\rightarrow\)) if there exists no term \(t \in T(F)\) such that \(s \rightarrow t\). If each proper subterm of redex is a normal form, we say the rewriting is innermost. We denote the innermost reduction of the relation \(\rightarrow\) as \(\rightarrow^{\text{in}}\). We use \(\circ\) to denote the composition of two relations. We write \(\to\) for the reflexive and transitive closure of \(\rightarrow\). We also write \(\to^\ast\) for the relation \(\rightarrow \circ \cdots \circ \rightarrow\) composed of \(n\) \(\rightarrow\)'s. The set of reachable terms from a term in \(T\) with respect to the relation \(\rightarrow\) is defined by \(\rightarrow^\ast[T] = \{t \mid s \in T, \ s \to^\ast t\}\). The reachability problem (resp. innermost reachability problem) with respect to \(\rightarrow\) is the problem that decides whether \(s \to^\ast s'\) (resp. \(s \to^\ast_{\text{in}} s'\)) or not, for given terms \(s\) and \(s'\).

A mapping \(\mu : F \rightarrow \mathcal{P}(\mathbb{N})\) is said to be a replacement map (or \(F\)-map) if \(\mu(f) \subseteq \{1, \ldots, \text{ar}(f)\}\) for all \(f \in F\). A context-sensitive term rewriting system (CS-TRS) is the pair \(R = (R, μ)\) of a TRS and a replacement map. The set of \(μ\)-replacing positions \(\text{Pos}^{μ}(t) \subseteq \text{Pos}(t)\) is recursively defined: \(\text{Pos}^{μ}(t) = \{ε\}\) if \(t\) is a constant or a variable, otherwise \(\text{Pos}^{μ}(f(t_1, \ldots, t_n)) = \{ε\} \cup \{ip \mid i \in μ(f), p \in \text{Pos}^{μ}(t_i)\}\). The rewrite relation induced by a CS-TRS \(R\) is defined: \(s \frac{R}{\text{sf}} t\) if and only if \(s \frac{R}{\text{sf}} t\) and \(p \in \text{Pos}^{μ}(t)\). If a term \(t\) has no redex at \(\text{Pos}^{μ}(t)\), we say \(t\) is a context-sensitive normal form. We denote the set of a context-sensitive normal form of \(R\) as \(\text{CS-NF}_R\). If each proper subterm of redex is a context-sensitive
normal form or not, in a μ-replacing position, we say the rewriting with CS-TRS is invariant.

A tree automaton (TA) is a quadruple $A = (F, Q, Q^f, \Delta)$ where $F$ is a finite
set of function symbols, $Q$ is a finite set of states, $Q^f(\subseteq Q)$ is a set of final states,
and $\Delta$ 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 $\text{ar}(f) = n$, and $q_1, \ldots, q_n, q \in Q$. We sometimes omit $F$ if it
is not necessary to specify explicitly. We can regard $\Delta$ as a (ground) TRS over
$\{q_1 \ldots q_n \mid q_i \in Q \}$. We denote $|A|$ as the length of a transition sequence $\alpha$ (if
$\alpha = s \xrightarrow{\Delta} t$, then $|\alpha| = n$). We say that a term $s \in (T(F))$ is accepted
by $q \in Q$ if $s \xrightarrow{\Delta} q$, and if $q \in Q^f$, we also say that $s$ is accepted by $A$. The set of all terms
accepted by $A$ is denoted by $L(A)$. We say $A$ recognizes $L(A)$. A set of terms $T$
is regular if there exists a TA that recognizes $T$. We use a notation $L(A, q)$ or
$L(\Delta, q)$ to represent the set $\{s \mid s \xrightarrow{\Delta} q, s \in T(F)\}$. A TA $A$ is deterministic if
$s \xrightarrow{\Delta} q$ and $s \xrightarrow{\mu} q'$ implies $q = q'$ for any $s \in T(F)$. A TA $A$ is complete if there
exists $q \in Q$ such that $s \xrightarrow{\Delta} q$ for any $s \in T(F)$. A state $q \in Q$ of $A$ is accessible
if $L(A, q) \neq \emptyset$, and if every state in $Q$ is accessible, $A$ is reduced.

A tree automaton with constraints between brothers (TACBB) is an 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 $\bot$ and $\top$ are constraints
between brothers, then a conjunction $c_1 \land c_2$ and a disjunction $c_1 \lor c_2$ are also
constraints between brothers. A term $f(t_1, \ldots, t_n)$ satisfies the constraints between
brothers if $c$ is true by assigning true to $T$, equality $i = j$ if $t_i = t_j$, and disequality $i \neq j$ if $t_i \neq t_j$, and disequality $i \neq j$ if $t_i = t_j$. Each transition rule is of the form $f(q_1, \ldots, q_n) \xrightarrow{\Delta} q$ or $q_1 \xrightarrow{\Delta} q$
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 $f(q_1, \ldots, q_n) \xrightarrow{\Delta} q$ 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 TA and TACBB are known:

**Theorem 1** All of the following holds for TAs and TACBBs:

1. For a given TA (TACBB) $A$, there exists a deterministic complete reduced

   TA (TACBB) $A'$ that recognizes $L(A)$.

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

   and complementation.

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

3. Decidability of Reachability for Right-linear Right-shallow CS-TRSs

In this section, we prove that reachability for right-linear right-shallow
CS-TRSs is decidable. To this end, we show the algorithm $P_c$ that constructs a tree
automaton recognizing the set of terms reachable by a right-linear right-shallow
CS-TRS from an input term.

The algorithm $P_c$ is based on the algorithm in Ref. 9). In Ref. 9), if a term $t$ matches
both a rewrite rule and a transition rule, then we produce transition rules to accept
the term obtained by the rewriting. For example, if we have the rewrite rule $a \to b$
and the transition rule $a \to q$, then we produce the transition rule $b \to q$. However
this algorithm can only deal with linear right-shallow TRSs. Therefore, we introduce the ideas in Refs. 14) and 17) to deal with the left-non-linear system, and the idea in Ref. 10) to deal with context-sensitive TRSs.

In Refs. 14) and 17), to deal with left-non-linear TRSs, we use subsets of the set
of states of input automata as the set of states of output automata. In Ref. 10), to
deal with context-sensitive TRSs, each state $q$ of input automata is divided
to $(q, a)$ and $(q, 1)^{14}$. We merge and modify these ideas to deal with right-linear
right-shallow CS-TRSs. We show an example of automata construction in the
following where it can be seen that the automaton obtained by $P_c$ recognizes
the set of terms reachable from an input term correctly.

**Example 2** Let $R = \{a \to b, b \to d, c \to d, f(x, x) \to g(x, c), g(x, x) \to
h(x)\}$, $\mu(f) = \{1\}$, $\mu(g) = \{1, 2\}$, $\mu(h) = \{1\}$, and
$A = (Q, Q^f, \Delta)$ where $Q = \{q^a, q^b, q^c, q^f(a, b)\}$, $Q^f = \{q^f(a, b)\}$,
$\Delta = \{a \to q^a, b \to q^b, c \to q^c, f(q^a, q^b) \to q^f(a, b)\}$, and hence
$L(A) = \{f(a, b)\}$. $P_c$ output the automaton $A_1 = (Q_1, Q^f_1, \Delta_1)$ that recognizes
$L(A) = \{f(a, b)\}$.

---

1 In Ref. 10), divided states are denoted as $\tilde{q}$ and $q$. 

---
The set of states $Q_*$ is the set $\{(P, a), \{\{p\}, i\}\}$ where $P \subseteq Q$, $P \neq \emptyset$, and $p \in Q$, $Q^f$ is $\{\{P^f, a\} | P^f \subseteq Q, P^f \cap Q^f \neq \emptyset\}$ and the set of transition rules $\Delta_*$ is
\[
\Delta_* = \left\{\begin{aligned}
a &\to \langle \{q^a\}, x \rangle, \\
b &\to \langle \{q^b\}, x \rangle, \\
c &\to \langle \{q^c\}, a \rangle, \\
d &\to \langle \{q^d\}, x \rangle, \\
g &\langle \{q^g\}, \{q^h\}, a \rangle \to \langle \{q^i\}, x \rangle, \\
h &\langle \{q^j\}, \{q^k\}, a \rangle \to \langle \{q^l\}, x \rangle, \\
f &\langle \{q^m\}, \{q^n\}, \{q^o\}, a \rangle \to \langle \{q^p\}, x \rangle
\end{aligned}\right\}
\]
where $P_i \subseteq \{q^a, q^b, q^c\}$ and $x \in \{a, i\}$. We obtain $L(A_*) = \{(f(a), b), (f(b), b), (g(b, c), a), (b, d), g(b, d), d(h(d)) = \overset{R}{\longrightarrow}|L(A)]\}$.

Here we describe Example 2. First, we obtain $Q_*$ by augmenting parameter $a$ or $i$ to each state and taking subset of $Q$ for the first components of the states. From the set of transition rules $\Delta_*$, it can be seen that $\langle \{q^a\}, i \rangle, \langle \{q^b\}, i \rangle, \langle \{q^c\}, i \rangle$, and $\langle \{q^d\}, \{q^e\}, i \rangle$ only accept the terms accepted by $q^a$, $q^b$, $q^c$, and $q^d$, respectively. From $\mu(f) = \{1\}$, the state in the second argument of $f$ in the transition rule must have $i$ as its second component. In this way, $A_*$ does not accept the terms obtained by rewriting the second argument of $f$. Moreover, we have $L(A_*) = \{\langle \{q^a\}, \{q^b\}, i \rangle \in L(A_*) \}$. Indeed, the state $\langle \{q^a\}, i \rangle$ accepts only $d$, and the term reachable from $b$ and $c$ is only $d$, too.

Since the term that is reachable from $f(a, b)$ and matches $f(x, c)$ is only $f(b, d)$, we produce the transition rule $g(\{q^a\}, \{q^b\}, a) \to \langle \{q^c\}, a \rangle$ from the rewrite rule $f(x, c) \to g(x, c)$. Since the term that is reachable from $f(a, b)$ and matches $f(x, c)$ is only $g(d, d)$, we produce the transition rule $h(\{q^a\}, \{q^b\}, a) \to \langle \{q^d\}, a \rangle$ from the rewrite rule $f(x, c) \to h(x)$. Concrete definition of the algorithm $P_{es}$ is the following:

Algorithm $P_{es}$:

Input A term $t$ and a right-shallow CS-TRS $\mathcal{R} = (R, \mu)$.

Output The TA $A_* = \langle Q_*, Q_*^f, \Delta_* \rangle$ such that $L(A_*) = \overset{R}{\longrightarrow}[\{t\}]$, if $R$ is right-linear.

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

(1) If there exists $\sigma : X \to T(F)$ such that $x_i \sigma \overset{\Delta_k}{\longrightarrow} \langle P_i, x_i \rangle$ for all $1 \leq i \leq n$, we apply the following inference rule:

\[
C[x_1, \ldots, x_n] \to g(r_1, \ldots, r_m) \in R, C[\langle P_1, x_1 \rangle, \ldots, \langle P_n, x_n \rangle] \overset{\Delta_k}{\longrightarrow} \langle \{q\}, a \rangle
\]

Let $I_j = \{i \mid x_i = r_j\}$. Each $P_j'$ and $x_j'$ is determined as follows:

- $P_j'$
- $x_j'$

(2) If there exists $\sigma : X \to T(F)$ such that $x_i \sigma \overset{\Delta_k}{\longrightarrow} \langle P_i, x_i \rangle$ for all $1 \leq i \leq n$, we apply the following inference rule:

\[
C[x_1, \ldots, x_n] \to g(r_1, \ldots, r_m) \in R, C[\langle P_1, x_1 \rangle, \ldots, \langle P_n, x_n \rangle] \overset{\Delta_k}{\longrightarrow} \langle \{q\}, a \rangle
\]
Example 3

Let us follow how the algorithm P works. We input the right-linear right-shallow CS-TRS \( R \) of Example 2 and the term \( f(a,b) \).

In the initializing step, at (1) of Step 1, we construct the automaton \( A \) of Example 2, and at (2) of Step 1, we have \( Q_0 = \{ (P,a), \{ (p), i \} \} \) where \( P \subseteq Q \), \( P \neq \emptyset \), and \( p \in Q \). \( Q_f = \{ (P_j, a) \} \) where \( P_j \subseteq Q \) and \( P_j \cap Q_f \neq \emptyset \), and \( \Delta_0 = \{ a \rightarrow \{ (q^a), x \}, b \rightarrow \{ (q^b), x \}, f(\{ (q^a), x \}, \{ (q^b), i \}) \rightarrow \{ (q^{f(ab)}, x) \} \} \) where \( x \in \{ a, i \} \).

In the saturation step at \( k = 0 \), we produce the transition rules \( \{ b \rightarrow \{ (q^a), a \}, d \rightarrow \{ (q^b), a \}, g(\{ (q^a), a \}) \rightarrow \{ (q^{f(ab)}, a) \} \} \) at Step 2.

At \( k = 1 \), we produce the transition rules \( \{ d \rightarrow \{ (q^a), a \}, h(\{ (q^b), a \}) \rightarrow \{ (q^{f(ab)}, a) \} \} \) at Step 2 and \( \{ b \rightarrow \{ (q^a), q^b \}, d \rightarrow \{ (q^b), q^a \} \} \) at Step 3.

At \( k = 2 \), we produce the transition rules \( \{ d \rightarrow \{ (q^a, q^b), a \}, d \rightarrow \{ (q^a, q^b), a \} \} \) at Step 3.

The saturation steps stop at \( k = 3 \), and we have \( \Delta_2 = \Delta_3 \).

The algorithm \( P_{cs} \) eventually terminates at some \( k \), because rewrite rules in \( R \) and states in \( Q_\ast \) are finite and hence possible transitions rules are finite. Apparently \( \Delta_0 \subseteq \cdots \subseteq \Delta_k = \Delta_{k+1} = \cdots \). Here we have two remarks.

Our first remark is that the state in which the second parameter is \( a \) does not always occur at rewritable positions. In Example 3, we have both \( \mu(h) = \emptyset \) and the transition rule \( h(\{ (q^a), q^b \}, a) \rightarrow \{ (q^{f(ab)}, a) \} \). However, this causes no problem. Indeed, the rewriting \( h(\{ (q^a), q^b \} \rightarrow c) \) is forbidden but \( c \) is reachable from \( g(b) \) if \( g(b) \rightarrow R g(c) \rightarrow c \).

Our second remark is that the former part of the input for \( P_{cs} \) is a term while it is an arbitrary tree automaton in Refs. 14, 17, and 10. Otherwise, \( P_{cs} \) may output an incorrect automaton as shown in the following example:

Example 4

Let \( R \) be \( \{ a \rightarrow b, a \rightarrow d, c \rightarrow d, f(x, x) \rightarrow g(x) \} \), \( \mu(f) = \mu(g) = \{ 1 \} \), and \( A = \{ Q_\ast, f, \Delta \} \) where \( Q = \{ q_1, q_2, q^f \} \), \( Q_f = \{ q^f \} \), \( \Delta = \{ a \rightarrow q_1, b \rightarrow q_2, c \rightarrow q_2, d \rightarrow f(q_1, q_2) \rightarrow q^f \} \), and hence \( L(A) = \{ f(a, b), f(a, c) \} \). Thus, \( \rightarrow R L(A) \) is the set \( \{ f(a, b), f(b, b), f(d, b), f(a, c), f(b, c), f(d, c), g(b) \} \).

Then, \( P_{cs} \) outputs the automaton \( \mathcal{A}_\ast \) of which transition rules in \( \Delta \) are \( \{ a \rightarrow \{ (q_1), x \}, b \rightarrow \{ (q_2), i \}, b \rightarrow (P, a), c \rightarrow \{ (q_2), x \}, d \rightarrow (P, a), f(\{ (q_1), x \}, \{ (q_2), i \}) \rightarrow \{ (q^f), x \}, g(\{ (q_2), a \}) \rightarrow \{ (q^f), a \} \} \) where \( P \subseteq \{ q_1, q_2 \} \) and \( x \in \{ a, i \} \). Hence, \( \mathcal{A}_\ast \) accepts the terms \( g(d) \) that is not in \( \rightarrow R L(A) \).

As for Example 4, preparing another state that accepts only \( b \) to construct a correct automaton is enough. However, guaranteeing the termination of a procedure if a new state is added in the procedure is difficult.

In the following, we show the correctness of \( P_{cs} \).

First, we show several propositions that are trivially derived from the definition.
of $P_{cs}$.

**Proposition 5** Let $t \in T(F)$. For any $q' \in Q$, $t \xrightarrow{\Delta}{\alpha}_{\Delta_0} \langle \{q'\}, \xi \rangle$ iff $t \xrightarrow{\Delta} q'$.

**Proof:** Direct consequence of the construction of $\Delta$, and $\Delta_0$. \hfill $\Box$

**Proposition 6** Let $t \in T(F)$. For any $k$, if $t \xrightarrow{\Delta}{\alpha}_{\Delta_k} \langle P, \xi \rangle \in Q_+$, then $t \xrightarrow{\Delta_{k+1}} \langle P, \xi \rangle$. Moreover, $P$ is of the form $\{q\}$.

**Proof:** The first claim follows from the fact that the transition rules in which right-hand-sides is the state having $\xi$ are not added at Step 2 or Step 3. The second claim follows from the construction of states. \hfill $\Box$

**Proposition 7** Let $t \in T(F)$. Then, $t \xrightarrow{\Delta_{k+1}} \langle P, \xi \rangle \in Q_+$ iff $t \xrightarrow{\Delta_{k}} \langle P, a \rangle \in Q_+$.

**Proof:** Direct consequence of the construction of $\Delta_0$. \hfill $\Box$

**Proposition 8** Let $t \in T(F)$. For any $k$, if $t \xrightarrow{\Delta} \langle P, \xi \rangle$, then $t \xrightarrow{\Delta_k} \langle P, \xi \rangle$ from Proposition 6. This proposition follows from Proposition 7 and $\Delta_0 \subseteq \Delta_k$. \hfill $\Box$

Next we show several technical lemmas. Lemmas 9, 10, 12, and 14 below are necessary to prove Lemmas 15 and 19 that are key lemmas to prove completeness and soundness. Lemmas 11 and 13 are auxiliary lemmas for Lemmas 12 and 14, respectively.

**Lemma 9** Let $s, t \in T(F)$, $s \xrightarrow{\Delta} \langle P, x \rangle$, and $t \xrightarrow{\Delta} \langle P', x' \rangle$. Then, $P = P'$ iff $s = t$.

**Proof:** First we have $s \xrightarrow{\Delta} \langle P, \xi \rangle$, $t \xrightarrow{\Delta} \langle P', \xi \rangle$, $P = \{q\}$, and $P' = \{q'\}$ for some $q, q' \in Q$ from Proposition 6 and Proposition 7. Then, we have $s \xrightarrow{\Delta} q = q$ and $t \xrightarrow{\Delta} q' = q'$ from Proposition 5 and the construction of $\Delta$. Thus, we have $P = P'$ iff $s = t$. \hfill $\Box$

**Lemma 10** If $\alpha : t \langle t' \rangle_p \xrightarrow{\Delta} \langle P, a \rangle$ and $p \in Pos^n(t)$, then there exists $\langle P', a \rangle$ such that $t' \xrightarrow{\Delta} \langle P', a \rangle$ and $t \langle t' \rangle_p \xrightarrow{\Delta} \langle P, a \rangle$.

**Proof:** We show this lemma by induction on $|\alpha|$. Let $p \in Pos^n(t)$.

(1) If $p = \varepsilon$, then $t = t'$, and hence $t' \xrightarrow{\Delta} \langle P, a \rangle$ follows.

(2) Consider the case $p = ip'$ for some $i \in N$. Then $\alpha$ can be represented as $t \langle t' \rangle_p \xrightarrow{\Delta} \langle P', a \rangle \xrightarrow{\Delta} \langle P, a \rangle$ or $t \langle t' \rangle_p = f(\ldots, t_i, t_{i+1}, \ldots)$ from $\Delta$. Hence we have $t \langle t' \rangle_p \xrightarrow{\Delta} \langle P, a \rangle$.

In the latter case, this lemma holds from the induction hypothesis.

In the latter case, since $ip' = p \in Pos^n(t)$, we have $i \in \mu(f)$. Hence $t_{i+1} = a$ follows from the construction of $\Delta$.

Moreover, $t \langle t' \rangle_p = f(\ldots, t_i, t_{i+1}, \ldots)$ from the induction hypothesis, there exists $\langle P', a \rangle \in Q_+$ such that $t' \xrightarrow{\Delta} \langle P', a \rangle$ and $t_i \langle \langle t'[P', a] \rangle \rangle \xrightarrow{\Delta} \langle P, a \rangle$. Hence we have $t \langle t' \rangle_p = f(\ldots, t_i, t_{i+1}, \ldots)$ from the induction hypothesis. \hfill $\Box$

**Lemma 11** If $\langle P_0, a \rangle \xrightarrow{\Delta} \langle P_1, a \rangle$ and $\langle P_2, a \rangle \xrightarrow{\Delta} \langle P_3, a \rangle$, then we have $\langle P_0 \cup P_2, a \rangle \xrightarrow{\Delta} \langle P_1 \cup P_3, a \rangle$.

**Proof:** We can assume $\langle P_0, a \rangle \xrightarrow{\Delta} \langle P_1, a \rangle$ and $\langle P_2, a \rangle \xrightarrow{\Delta} \langle P_3, a \rangle$ without loss of generality.

First, we prove the claim that $\langle P_0 \cup P_2, a \rangle \xrightarrow{\Delta} \langle P_1 \cup P_3, a \rangle$. If $n = 0$, the claim trivially holds. If $n = 1$, the claim holds from (2) of Step 3 of $P_{cs}$. If $n > 1$, the claim holds by repeating the process for $n = 1$.

Moreover, we can show the claim that $\langle P_1 \cup P_2, a \rangle \xrightarrow{\Delta} \langle P_1 \cup P_2, a \rangle$ similarly to the previous claim. \hfill $\Box$

**Lemma 12** If $t \xrightarrow{\Delta} \langle P_1, a \rangle$ for $1 \leq j \leq m$, then we have $t \xrightarrow{\Delta} \langle \bigcup_{1 \leq j \leq m} P_j, a \rangle$.

**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 show this lemma.

Let $t = f(t_1, \ldots, t_n)$. Then, each transition sequence is represented as $f(t_1, \ldots, t_n) \xrightarrow{\Delta} f(\langle P_1, x_{i_1} \rangle, \ldots, \langle P_{k_i}, x_{j_i} \rangle) \xrightarrow{\Delta} \langle P_j, a \rangle \xrightarrow{\Delta} \langle P_k, a \rangle$ for $j \in \{1, 2\}$. From Lemma 11, we have $\langle P_1 \cup P_2, a \rangle \xrightarrow{\Delta} \langle P_1 \cup P_2, a \rangle$. Therefore, we show that $f(t_1, \ldots, t_n) \xrightarrow{\Delta} \langle P_1 \cup P_2, a \rangle$. \hfill $\Box$
From (1) of Step 3 of $P_{cs}$, we have the transition rule $f((P_1, x_1), \ldots, (P_n, x_n)) \rightarrow (P'_{1} \cup P'_{2}, a) \in \Delta$, where

- $P_i = \begin{cases} P_i^1 \quad \text{if } x_i' = i \text{ for some } j \in \{1, 2\}, \text{ and} \\ P_i^1 \cup P_i^2 \quad \text{if } x_i' = x_i^2 = a. \end{cases}$
- $x_i = \begin{cases} a \quad \text{if } x_i^1 = x_i^2 = a, \text{ and} \\ i \quad \text{otherwise.} \end{cases}$

Here we show that $t_i \xrightarrow{\Delta} (P, x_i)$ for $1 \leq i \leq n$.

- For $i$ such that $x_i = i$, $P_i$ is $P_i^1$ or $P_i^2$ and hence we have $t_i \xrightarrow{\Delta} (P, x_i)$.
- For $i$ such that $x_i = a$, $P_i$ is $P_i^1 \cup P_i^2$ and hence we have $t_i \xrightarrow{\Delta} (P, x_i)$ from the induction hypothesis.

Thus, we have the transition $f(t_1, \ldots, t_n) \xrightarrow{\Delta} f((P_1, x_1), \ldots, (P_n, x_n)) \xrightarrow{\Delta} (P, a)$.

**Lemma 13** If $(P_1, a) \xrightarrow{\Delta} (P, a)$, then there exists $P'_i \subseteq P_i$ such that $(P'_i, a) \xrightarrow{\Delta} (P', a)$ for all $P' \subseteq P$ where $P' \neq \emptyset$.

**Proof:** By the induction on $|P| + |P_1|$, we show the proof for the case of $(P_1, a) \xrightarrow{\Delta} (P, a)$. If $(P_1, a) = (P, a)$, then this lemma holds trivially. If $|(P_1, a) \rightarrow (P, a)| > 1$, then we can prove this lemma by applying the proof for $(P_1, a) \xrightarrow{\Delta} (P, a)$.

Let $P' = P \setminus P'$. We show that if $P_1$ exists such that $(P'_1, a) \xrightarrow{\Delta} (P', a)$ where $P'_i \subseteq P_i$. If $|P| = 1$, then the claim holds trivially. If $|P| > 1$, we can assume that the transition rule $(P_1, a) \rightarrow (P, a) \in \Delta$ is produced by the rules $((P'_1, a) \rightarrow (P', a)) \in \Delta$, where $j \in \{1, 2\}$, $P_i^1 \cup P_i^2 = P$, and $P_i^1 \cup P_i^2 = P_1$ by (2) of Step 3 of $P_{cs}$. Note that we have $|P| + |P_1| < |P| + |P_1|$ for $j \in \{1, 2\}$ because if $|P| + |P_1| = |P_1| + |P_1|$ then we have $P_i^1 = P_i^2$ and $P_i^1 = P_1$, and hence the rule $(P_1, a) \rightarrow (P, a) \in \Delta$ is the same as $(P'_1, a) \rightarrow (P', a) \in \Delta$, as $j \in \{1, 2\}$.

For each $j$, we also have the transition rule $(P'_j, a) \rightarrow (P', a) \in \Delta$, for some $P'_j \subseteq P_i^j$ from the induction hypothesis.

Thus, we obtain $(P_i^1 \cup P_i^2, a) \xrightarrow{\Delta} ((P_i^1 \cup P_i^2) \setminus P'_{i}, a) = (P', a)$ where $P_i^1 \cup P_i^2 \subseteq P_i^1 \cup P_i^2 = P_1$ by (2) of Step 3 of $P_{cs}$.

**Lemma 14** If $t \xrightarrow{\Delta} (P, a)$, then $t \xrightarrow{\Delta} (P', a)$ for any $P' \subseteq P$.

**Proof:** We can assume that the transition $t \xrightarrow{\Delta} (P, a)$ is represented as $t = f(t_1, \ldots, t_n) \xrightarrow{\Delta} f((P_1, x_1), \ldots, (P_n, x_n)) \xrightarrow{\Delta} (P, a)$. From Lemma 13, there exists $P'_i \subseteq P_i$ such that $(P'_i, a) \xrightarrow{\Delta} (P', a)$ for all $P' \subseteq P$. Therefore, we show that we have $t = f(t_1, \ldots, t_n) \xrightarrow{\Delta} f((P_1, x_1), \ldots, (P_n, x_n)) \xrightarrow{\Delta} (P', a)$. Let $P'_n = P_1 \cup P'_n$. We show the claim by induction on $\Sigma_n = |P_1| + |P'_n|$ and $|t|$. If $|P_1| = 1$, then the claim holds trivially. If $|P_1| > 1$, then the transition rule $f((P_1, x_1), \ldots, (P_n, x_n)) \rightarrow (P', a)$ is produced from the transition rules $f((P'_1, x_1), \ldots, (P'_n, x_n)) \rightarrow (P', a)$ where $j \in \{1, 2\}$ by (1) of Step 3 of $P_{cs}$ and $P_i$’s, and $x_i$’s are represented as follows:

- $P'_i = P_i^1 \cup P_i^2$.
- $P_i = \begin{cases} P_i^1 \quad \text{if } x_i' = i \text{ for some } j \in \{1, 2\}, \text{ and} \\ P_i^1 \cup P_i^2 \quad \text{if } x_i' = x_i^2 = a. \end{cases}$
- $x_i = \begin{cases} a \quad \text{if } x_i^1 = x_i^2 = a, \text{ and} \\ i \quad \text{otherwise.} \end{cases}$

Here, we show that $t \xrightarrow{\Delta} (P'_i, x'_i)$ for $j \in \{1, 2\}$ and $1 \leq i \leq n$.

- For $i$ such that $x_i = a$, we have $x_i^1 = a$ and $P_i^j \subseteq P_i$. Thus, we have $t \xrightarrow{\Delta} (P_i^j, x_i')$ from the induction hypothesis.
- For $i$ such that $x_i = i$, we have $|L(\Delta, (P_i^1, x_i')) \cap L(\Delta, (P_i^1, x_i'))| \neq 0$, and, $x_i^1 \neq x_i^2 = i$. From Lemma 9, $t_i$ is the only term accepted by $(P_i^1, a)$ where $j = 1$ or $j = 2$, and from $L(\Delta, (P_i^1, x_i')) \cap L(\Delta, (P_i^1, x_i')) \neq 0$, we have $t_i \xrightarrow{\Delta} (P_i^1, x_i')$ for both $j = 1$ and $j = 2$.

Thus, we have $f(t_1, \ldots, t_n) \xrightarrow{\Delta} f((P'_1, x'_1), \ldots, (P'_n, x'_n)) \xrightarrow{\Delta} (P', a)$ for both $j = 1$ and $j = 2$.

Moreover, we have $\Sigma_n = |P_1| + |P'_n| < \Sigma_n = |P_1| + |P_1|$ 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 $f((P_1, x_1), \ldots, (P_n, x_n)) \rightarrow (P, a)$. Hence, we have $t \xrightarrow{\Delta} (P_i^1 \cup P_i^2, a)$ for both $j = 1$ and $j = 2$ from the induction hypothesis.

Thus, we have $t \xrightarrow{\Delta} (P'_i \cup P_i^2, a)$ from Lemma 12.

The following lemma is a key lemma for completeness of $P_{cs}$.

**Lemma 15** Let $\mathcal{R}$ be right-shallow CS-TRS. Then $s \xrightarrow{\Delta} (P, a)$ and $s \xrightarrow{\Delta} (P, a)$ implies $t \xrightarrow{\Delta} (P, a)$.\hfill $\Box$
Proof: We present the proof in the case of \( s \stackrel{a}{\rightarrow} t \) because the proof in the case of \( s \stackrel{a}{\rightarrow} t \) is trivial and in the case of \( s \stackrel{a}{\rightarrow} t \) repeatedly. Let \( s \stackrel{a}{\rightarrow} (P, a) \) and \( s = s[l] \) \( \stackrel{a}{\rightarrow} s[l] \) \( \stackrel{a}{\rightarrow} t \) for some rewrite rule \( l \rightarrow r \in R \), where \( p \in \text{Pos}^{d}(s) \). We have a transition sequence \( s[l] \) \( \stackrel{a}{\rightarrow} s[(P', a)] \) \( \stackrel{a}{\rightarrow} (P, a) \) for some \( (P', a) \in Q, \) by Lemma 10.

From Lemma 14, we have \( l \sigma \) \( \rightarrow (\{q\}, a) \) for all \( q \in P \). Therefore, we prove that \( \sigma \) \( \rightarrow (\{q\}, a) \) for all \( q \in P \), because if we can prove this, we have \( s[l] \) \( \stackrel{a}{\rightarrow} s[(P', a)] \) \( \stackrel{a}{\rightarrow} (P, a) \) from Lemma 12.

(1) Consider the case where the rewrite rule is of the form \( C[x_1, \ldots, x_n] \rightarrow g(r_1, \ldots, r_m) \) where \( C \) has no variable. The diagram of this case is shown in Fig. 2. Here, \( C[x_1, \ldots, x_n] \) \( \rightarrow (\{q\}, a) \) is represented in \( C[x_1, \ldots, x_n] \) \( \rightarrow (\{q\}, a) \). For all \( i \), \( \rightarrow (P, x_i) \) \( \rightarrow (P', x_i) \) \( \rightarrow (\{q\}, a) \) for some \( (P, x_i) \) \( \rightarrow (r_1, \ldots, r_m) \) \( \rightarrow (P', x_i) \) \( \rightarrow (\{q\}, a) \), and \( \rightarrow (x, r) \) \( \rightarrow (q, a) \). Therefore, we have \( g(r_1, \ldots, r_m) \sigma \) \( \rightarrow (\{q\}, a) \).

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

(2) In the case where the rewrite rule is of the form \( C[x_1, \ldots, x_n] \rightarrow x_l \), we can prove this lemma similarly to the previous case.

The following lemma shows completeness of \( P_{cs} \).

Lemma 16 If \( R \) is right-shallow CS-TRS, then \( L(A_{\sigma}) \equiv \llbracket R \rrbracket \llbracket L(A) \rrbracket \).

Proof: Let \( s \stackrel{a}{\rightarrow} t \) and \( s \stackrel{a}{\rightarrow} q \in Q' \). Since we have \( s \stackrel{a}{\rightarrow} (\{q\}, a) \) from Proposition 5, we also have \( s \stackrel{a}{\rightarrow} (\{q\}, a) \) from Proposition 7. Hence \( t \stackrel{a}{\rightarrow} (\{q\}, a) \) \( \in Q' \) follows by Lemma 15.

To prove the soundness of \( P_{cs} \), we define the following measures of transition and order. These are necessary to prove soundness of \( P_{cs} \).

Definition 17 Let \( ||t \stackrel{a}{\rightarrow} P|| \) be the sequence of integers defined as follows\(^1\):

\[^1\] Sometimes we have \( k - 1 < 0 \) in this definition. \( \Delta_k \) for \( k < 0 \) is undefined but we assume it as an empty set.
Lemma 19 Let \( \mathcal{R} \) be a right-linear right-shallow CS-TRS. Then, \( \alpha : t \xrightarrow{\Delta} (P,a) \) implies that there exists \( s \) and \( q \in P \) such that \( s \xrightarrow{R} t \) and \( s \xrightarrow{\Delta} (\{q\},a) \).

Proof: We show this lemma by induction on \( \alpha \) with respect to \( \sqsupseteq \). Since we have \( t \xrightarrow{\Delta} (\{q\},a) \) for all \( q \in P \) from Lemma 14, we show the proof in the case where \( P \) is of the form \( \{q\} \).

(1) Consider the case where \( \alpha \) is represented as \( t = g(t_1, \ldots, t_m) \xrightarrow{\Delta} g(P_1, x_1, \ldots, P_m, x_m) \xrightarrow{\Delta_1 \Delta_{m-1}} (\{q\},a) \).

(a) If \( k = 0 \), the transition rule \( g(P_1, x_1, \ldots, P_m, x_m) \xrightarrow{\Delta_1 \Delta_{m-1}} (\{q\},a) \) is produced at Step 1, and hence each \( P_j \) is of the form \( \{q_j\} \) and we have \( j \in \mu(g) \) iff \( x'_j = a \). For \( j \not\in \mu(g) \), we have \( x'_j = a \) and hence there exists \( s_j \) such that \( s_j \xrightarrow{R} t_j \) and \( s_j \xrightarrow{\Delta_0} (P'_j, a) = (\{q_j\},a) \) from the induction hypothesis. For \( j \not\in \mu(g) \), we have \( x'_j = 1 \) and \( t_j \xrightarrow{\Delta} (P'_j,1) \) from Proposition 6. We take \( s_j = t_j \) for \( j \not\in \mu(g) \). Finally, we obtain \( g(s_1, \ldots, s_m) \xrightarrow{\Delta} g(t_1, \ldots, t_m) = t \) and \( g(s_1, \ldots, s_m) \xrightarrow{\Delta} g(P_1', x_1', \ldots, P_m', x_m') \xrightarrow{\Delta} (\{q\},a) \). Thus, this lemma holds in the case \( k = 0 \).

(b) If \( k > 0 \), the transition rule \( g(P_1', x_1', \ldots, P_m', x_m') \xrightarrow{\Delta} (\{q\},a) \) is produced at \( (1) \) of Step 2. The diagram of this case is shown in Fig. 3. From the production of the transition rule, we have \( C[x_1, \ldots, x_n] = g(r_1, \ldots, r_m) \in R \) where \( C \) has no variable and \( C(P_1, x_1), \ldots, (P_n, x_n) \xrightarrow{\Delta_{m-1}} (\{q\},a) \), and \( \sigma' : X \rightarrow T(F) \) such that \( x_0^j \cdot \Delta \sigma' \xrightarrow{\Delta_{m-1}} (P,j) \) for all \( 1 \leq i \leq n \) and each \( (P_j', x'_j) \) is represented as follows:

- \( P_j' = \begin{cases} \{q_j\} & \text{if } r_j \not\in X, \\ \bigcup_{i \in I_j} P_i & \text{if } r_j \in X \land \exists i \in I_j, x_i = 1, \text{ and} \\ a & \text{otherwise.} \end{cases} \)

- \( x'_j = \begin{cases} 1 & \text{if } j \not\in \mu(g) \land (r_j \in X \Rightarrow \exists i \in I_j, x_i = 1), \\ 0 & \text{otherwise.} \end{cases} \)

where \( I_j = \{ i \mid x_i = r_j \} \). In the following, we show that there exists the substitution \( \sigma' \) such that \( g(r_1, \ldots, r_m) \sigma \xrightarrow{\Delta} g(t_1, \ldots, t_m) \) and \( \alpha' : g(r_1, \ldots, r_m) \sigma \xrightarrow{\Delta} g((P_1', x_1'), \ldots, (P_m', x_m')) \xrightarrow{\Delta_{m-1}} (\{q\},a) \) where \( \alpha' \sqsubseteq \alpha \).

(i) For \( j \not\in \mu(g) \) such that \( r_j \not\in X \), we have \( P'_j = \{q_j^*\} \) and \( x'_j = 1 \). Since \( t_j \xrightarrow{\Delta} (\{q_j^*\},1) \) is from Proposition 7, we have \( t_j = r_j \) from Proposition 5 and the construction of \( A \).

(ii) For \( j \in \mu(g) \) such that \( r_j \not\in X \), we have \( P'_j = \{q_j^*\} \) and \( x'_j = a \). Hence, we have \( s_j \xrightarrow{\Delta} t_j \) and \( s_j \xrightarrow{\Delta_0} (P'_j, a) = (\{q_j^*\},a) \) from the induction hypothesis. Since we have \( s_j \xrightarrow{\Delta_0} (\{q_j^*\},a) \) from Proposition 7, we have \( s_j = r_j \) from Proposition 5 and the construction of \( A \).

(iii) For \( j \) such that \( r_j \in X, j \not\in \mu(g) \), and there exists \( i \in I_j \) such that \( x_i = 1 \), we have \( t_j \xrightarrow{\Delta} (P'_j, x'_j) = (P_i,1) \). Hence, we have \( t_j \xrightarrow{\Delta} (P'_j, x'_j) \) from Proposition 7, and let \( r_j \sigma = t_j \).

(iv) For \( j \) such that \( r_j \in X, j \in \mu(g) \), and there exists \( i \in I_j \) such that \( x_i = 1 \), we have \( t_j \xrightarrow{\Delta} (P'_j, x'_j) = (P_i, a) \). Since we have \( \langle P_i, a \rangle \) where \( P_i \) is of the form \( \{q_i\} \) from
Proposition 6, there exists a \( x_j \) such that \( \Delta \overset{\sim}{\rightarrow} P_j, x_j \) and \( \Delta \overset{\sim}{\rightarrow} P_j, x_j' \) from the induction hypothesis. Let \( s_j \) be \( r_j \sigma \).

(Ⅴ) For \( j \) such that \( r_j \in X \) and there exists \( i \in I_j \) such that \( x = i \), we take \( r_j \sigma = t_j \).

Note that \( \sigma \) is well defined from the right-linearity of \( R \) and we have \( \alpha' \subseteq \alpha \) because \( \alpha \) does not occur in \( \alpha' \), and we have \( \alpha' \ni j \overset{\sim}{\rightarrow} \{P_j, x_j\} \subseteq \{t_j \overset{\sim}{\rightarrow} \{P_j, x_j\}\} \) for all \( 1 \leq j \leq m \).

Next, we define a substitution \( \sigma'' : \text{Var}(i_1, \ldots, i_n) \rightarrow T(F) \) as follows:

\[
x \sigma'' = \begin{cases} x \sigma'' \text{ if there exists } r_j \text{ such that } r_j = x \\ x \sigma' \text{ otherwise.} \end{cases}
\]

Here, we show that we have \( x \sigma'' \overset{\sim}{\rightarrow} P_i, x_i \) for all \( 1 \leq i \leq n \).

(i) For \( i \) such that there exists \( j \) such that \( x_i = r_j \) and \( i' \in I_j \) such that \( x_i = i' \), we have \( P_i \subseteq P_j \) and \( x \sigma'' \overset{\sim}{\rightarrow} P_i, x_i \).

(ii) For \( i \) such that there exists \( j \) such that \( x_i = r_j \) and no \( i' \in I_j \) such that \( x_i = i' \), we have \( P_i \subseteq P_j \) and \( x \sigma'' \overset{\sim}{\rightarrow} P_i, x_i \).

(iii) For \( i \) such that there exists no \( j \) such that \( x_i = r_j \), we have \( x \sigma'' \overset{\sim}{\rightarrow} P_i, x_i \) from the construction of the rule.

Thus, we have \( \beta : C[x_1, \ldots, x_n] \sigma'' \overset{\sim}{\rightarrow} C[P_i, x_i] \) from Lemma 9 and \( x \sigma'' \overset{\sim}{\rightarrow} \{P_i, x_i\} \) from Proposition 5. If \( x = i \), we have \( x \sigma'' \overset{\sim}{\rightarrow} \{P_i, x_i\} \) from Proposition 6. If \( x = \alpha \), there exists the term \( s \) such that \( s \overset{\sim}{\rightarrow} t \) and \( s \overset{\sim}{\rightarrow} \{q\}, a \) from Lemma 19 and we have \( s \overset{\sim}{\rightarrow} \{q\}, a \) from Proposition 7.

Thus, we have \( s \overset{\sim}{\rightarrow} \{q\}, a \) from Proposition 5. The following theorem is proved by Lemma 16 and 20.

**Theorem 21** For any right-linear right-shallow CS-TRS \( R \), we can construct a TA recognizing the set of terms that is reachable from a term. Thus, context-sensitive reachability is decidable for right-linear right-shallow TRSs.

4. Decidability of Innermost Reachability for Shallow CS-TRSs

In this section, we show that innermost reachability for shallow CS-TRSs is decidable. Similarly to the previous section, we show the algorithm \( P_{\text{csin}} \) that constructs the tree automaton accepting the set of terms reachable by innermost reduction of a shallow CS-TRS from an input term. The algorithm \( P_{\text{csin}} \) is a modification of the algorithm \( P_{\text{cs}} \) by the idea in Ref. 11). States of output automata obtained by \( P_{\text{csin}} \) have three components, while the one by \( P_{\text{cs}} \) has two components. Since we must check whether each proper subterm of redex is a context-sensitive normal form or not in the innermost case, we augment...
the parameter that shows whether the state accepts the context-sensitive normal form or not. Therefore, first we show the construction of a deterministic complete reduced tree automata accepting the set of context-sensitive normal forms, and then we show \( P_{\text{csm}} \).

### 4.1 Tree Automata Accepting Context-sensitive Normal Forms

In this subsection, we give an algorithm to construct a deterministic complete reduced tree automata recognizing the set of context-sensitive normal forms for shallow CS-TRS \( R \). However, in general, ordinary tree automata cannot recognize the set of context-normal forms for shallow CS-TRS. Therefore we use tree automata with constraints between brothers (TACBB)\(^3\).

The procedure is similar to the ones for TRSs\(^3\). The steps of the algorithm to construct TACBB \( A_{\text{NF}} \) are as follows:

1. Construct the TACBB \( A_i \) that recognizes the set of terms having a redex \( \sigma \) at a \( \mu \)-replacing position for each \( l \rightarrow r \in R \) and determine it.
2. Construct the union of all \( A_i \)'s and convert the TA into complete and reduced TA \( A \).
3. We output a TA recognizing the complement of \( \mathcal{L}(A) \) as \( A_{\text{NF}} \).

The steps (2) and (3) are obviously possible from Theorem 1. Now we show the details of step (1).

Each component of \( A_i \) is as follows.

- \( Q_i = \{ u^i, u_{\bot} \} \cup \{ u_t \mid t < l, t \not\in X \} \).
- \( Q_i^f = \{ u^i \} \)
- \( \Delta_i \) consists of the following transition rules:
  1. \( f(u_1, \ldots, u_n) \xrightarrow{\cdot} u_{\bot} \) for each \( f \in F \).
  2. \( f(u_1, \ldots, u_n) \xrightarrow{\cdot} u_{f(t_1, \ldots, t_n)} \) for each \( f \in F \) and state \( u_{f(t_1, \ldots, t_n)} \).
  3. \( f(u_{s_1}, \ldots, u_{s_n}) \xrightarrow{\cdot} u^i \) where \( f(s_1, \ldots, s_n) \) is the term obtained by replacing all variables in \( f = f(l_1, \ldots, l_n) \) by \( \bot \), and \( c \) is the conjunction of all equalities \( i = j \) where \( l_i = l_j \in X \).

From the induction hypothesis, we have \( t, \sigma \xrightarrow{\Delta_i} u_t \). From the construction of \( \Delta_i \), we have the transition rule \( f(u_1, \ldots, u_n) \xrightarrow{\cdot} u_{f(t_1, \ldots, t_n)} \).

- \( \mathcal{L}(A_i, u_t) \) is equal to the singleton set that consists of \( t \vdash l_i \), (that is, \( \mathcal{L}(A_i, u_t) = \{ t \in T(F) \} \).

#### Lemma 22

\( \mathcal{L}(A_i, u_t) \) is equal to the singleton set that consists of \( t \vdash l_i \), (that is, \( \mathcal{L}(A_i, u_t) = \{ t \in T(F) \} \).

**Proof:**

By induction on the height \( |t| \) of \( t \), we prove the claim that \( t \xrightarrow{\cdot \Delta_i} u_t \) for the proper non-variable subterm \( t \) of \( l \). We can represent \( t \) as \( f(l_1, \ldots, l_n) \) where \( n \geq 0 \). From the construction of (ii) of \( \Delta_i \), we have the transition rule \( f(u_1, \ldots, u_n) \xrightarrow{\cdot \Delta_i} u_{f(t_1, \ldots, t_n)} \).

From the induction hypothesis, we have \( t, \sigma \xrightarrow{\Delta_i} u_t \), for all \( 1 \leq i \leq n \). Thus, we have \( \sigma \xrightarrow{\Delta_i} f(u_1, \ldots, u_n) \xrightarrow{\cdot \Delta_i} u_t \).

We show that if \( \alpha : t \xrightarrow{\cdot \Delta_i} u_{f(t_1, \ldots, t_n)} \) then we have \( t \xrightarrow{\cdot \Delta_i} u_t \) by induction on \( |\alpha| \). From the construction of \( \Delta_i \), the last transition rule applied in \( \alpha \) is represented as \( f(u_1, \ldots, u_n) \xrightarrow{\cdot \Delta_i} u_{f(t_1, \ldots, t_n)} \).

From the induction hypothesis, we have \( t, \sigma \xrightarrow{\Delta_i} u_t \), for all \( 1 \leq i \leq n \).

Thus, we have \( t \xrightarrow{\cdot \Delta_i} f(l_1, \ldots, l_n) \).

#### Lemma 23

\( \mathcal{L}(A_i) = \{ t[s]_p \mid t \in T(F) \}, s \) is a ground instance of \( l, p \in \text{Pos}^0(t) \).

**Proof:**

Let \( l = f(l_1, \ldots, l_n) \). First, we show that \( s = f(l_1, \ldots, l_n) \xrightarrow{\cdot \Delta_i} u^i \). For \( l_i \not\in X, l_i \) is ground from shallowness of \( l \) and \( l_i \xrightarrow{\cdot \Delta_i} u_i \). For \( l_i \in X \), we have \( l_i, \sigma \xrightarrow{\cdot \Delta_i} u_i \). From (iii) of construction of \( \Delta_i \), we have the transition rule \( f(u_{s_1}, \ldots, u_{s_n}) \xrightarrow{\cdot \Delta_i} u^i \) where \( s_i = l_i \) for \( i \not\in X \) and \( s_i = \bot \) for \( i \in X \), and \( c \) is the conjunction of all equalities \( i = j \) where \( l_i = l_j \in X \).

Since we have \( l_i, \sigma \xrightarrow{\cdot \Delta_i} u_i \), we have \( s, \sigma \xrightarrow{\cdot \Delta_i} u^i \) and from (i) and (iv) of construction of \( \Delta_i \), we have \( t[s]_p \xrightarrow{\cdot \Delta_i} u^i \).

Let \( l = f(l_1, \ldots, l_n) \) and \( t \xrightarrow{\cdot \Delta_i} u^i \), then we have \( t \xrightarrow{\cdot \Delta_i} f[l_s]_p \xrightarrow{\cdot \Delta_i} u^i \) where \( s_i = l_i \) for \( l_i \not\in X, s_i = \bot \) for \( l_i \in X \), and \( p \in \text{Pos}^0(t) \) from (iii) and (iv) of the construction of \( \Delta_i \).

From Lemma 22, we have \( t[p] = l_i \not\in X \). Moreover, since the transition \( t[l_s]_p \xrightarrow{\cdot \Delta_i} u^i \) has the constraint \( c \) that is the conjunction of all equalities \( i = j \) where \( l_i = l_j \in X \), we have \( t[p] = t[j]_p \) for \( l_i = l_j \in X \).

Hence \( t[p] \) is a ground substitution of \( l \). Thus we have \( t \xrightarrow{\cdot \Delta_i} t[s]_p \) for some ground instance \( s \) of \( l \).

\[ \square \]
The method of determinization is the so-called “subset construction.” The claim “$t \xrightarrow{\Delta_d} S$ iff $S = \{ q \mid t \xrightarrow{\Delta} q \}$” holds where $\mathcal{A}'$ is determined from $\mathcal{A}$ by subset construction.

Therefore, the following lemma holds.

**Lemma 24** 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 $\mathcal{A}_t$ respectively, and $\mathcal{A}'_t$ be a determined TACBB from $\mathcal{A}_t$ by subset construction. Then, $t \xrightarrow{\mathcal{A}'_t} \{u_s\} \cup S''$ iff $t = s$.

**Proof:** From Lemma 22 and shallowness of $l$. As shown in Lemma 23, the TACBB $\mathcal{A}_t$ recognizes the set of terms having a redex $l \sigma$ at a $\mu$-replacing position. Now we obtain the following lemma.

**Lemma 25** For a CS-TRS $\mathcal{R}$, we can construct a deterministic complete reduced TACBB $\mathcal{A}_{\text{NF}}$ that recognizes CS-NF$_\mathcal{R}$.

**Proof:** By step (1) of the algorithm, we obtain a TACBB $\mathcal{A}_t$ for each $l \rightarrow r \in \mathcal{R}$, and we can determine them. Let the determined TACBB from $\mathcal{A}_t$ be $\mathcal{A}'_t = \langle Q'_t, Q''_t, \Delta'_t \rangle$.

We can obtain a TACBB $\mathcal{A}' = \langle F, Q', Q''_t, \Delta' \rangle$ that recognizes the following set:

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

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

- $Q' = \{\{u_1, \ldots, u_n\} \mid u_i \in Q''_t\}$,
- $Q''_t = \{\{u_1, \ldots, u_n\} \mid \exists i. u_i \in Q''_t\}$,
- $f(\langle u_{11}, \ldots, u_{1m} \rangle, \ldots, \langle u_{n1}, \ldots, u_{nm} \rangle) \xrightarrow{\Delta'} \langle u_{11}, \ldots, u_{nm} \rangle \in \Delta'$ where $f(\langle u_{11}, \ldots, u_{1m} \rangle) \xrightarrow{\Delta} u_i \in \Delta'$ and $c = c_1 \land \cdots \land c_m$.

This is the construction of the union of all $\mathcal{A}'_t$’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 $q_a$ and new transition rules such that $f(q_1, \ldots, q_n) \xrightarrow{\Delta'} q_a$ where $q_1, \ldots, q_n \in Q'$ and $c = T$ if $f(q_1, \ldots, q_n)$, which does not occur in any transition rule of $\Delta'$, otherwise $c$ is equivalent to $\neg (c_1 \lor \cdots \lor c_k)$ where $f(q_1, \ldots, q_n) \xrightarrow{\Delta} q \in \Delta'$ for some $q \in Q'$.

Since the emptiness problem of TACBB is decidable from Theorem 1, we can check whether each state is accessible or not and hence we can construct a reduced TACBB $\mathcal{A}''$ 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.

From Lemma 23, $L(\mathcal{A}'')$ is the set of terms having redex at a $\mu$-replacing position. Thus we can obtain the deterministic, complete, and reduced TACBB $\mathcal{A}_{\text{NF}}$ recognizing CS-NF$_\mathcal{R}$ by the algorithm.

We show an example of $\mathcal{A}_{\text{NF}}$ in Appendix A.1.1.

For the constructed TA $\mathcal{A}_{\text{NF}}$, the following proposition holds from Lemmas 24 and 25.

**Proposition 26** Let $t \in \mathcal{T}(F)$, $u \in Q_{\text{NF}}$, and $t \xrightarrow{\mathcal{A}'_t} u$. If $t$ is a proper subterm of some $l \rightarrow r \in \mathcal{R}$ where $\mathcal{R} = (R, \mu)$ is a shallow CS-TRS, then $u$ accepts no term other than $l$.

**Proof:** Let $\mathcal{A}'_t$ be the deterministic TACBB obtained in step (1) of the algorithm. From Lemma 22, Lemma 24, and shallowness of $R$, we have $t \xrightarrow{\mathcal{A}'_t} S \in Q''_t$ where $S$ contains $u_s \in Q_t$ and there exists no term other than $t$ accepted by $u_s$. Thus, from the construction of $\Delta_{\text{NF}}$, there exists no term accepted by $u$ other than $t$.

**Proposition 27** If $f(u_1, \ldots, u_n) \xrightarrow{\Delta} u \in \Delta_{\text{NF}}$ and $u \in Q_{\text{NF}}$, then $i \in \mu(f)$ implies $u_i \in Q_{\text{NF}}$.

**Proof:** Let $f(u_1, \ldots, u_n) \xrightarrow{\Delta} u \in \Delta_{\text{NF}}$, $u \in Q_{\text{NF}}$, and assume $u_i \notin Q_{\text{NF}}$ for some $i \in \mu(f)$. Since $\mathcal{A}_{\text{NF}}$ is a reduced TACBB from Lemma 25, there exists ground terms $t_1, \ldots, t_n$ such that $t_j \xrightarrow{\Delta} u_j$ for each $j$ $(1 \leq j \leq n)$. Hence we have $f(t_1, \ldots, t_n) \xrightarrow{\Delta_{\text{NF}}} f(u_1, \ldots, u_n) \xrightarrow{\Delta_{\text{NF}}} u$.

Here $f(t_1, \ldots, t_n) \in \text{CS-NF}_\mathcal{R}$ and $t_i \notin \text{CS-NF}_\mathcal{R}$ is from Lemma 25. 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 \( f(t_1, \ldots, t_n) \in \text{CS-NF}_R \). □

### 4.2 An Algorithm to Construct the Set of Reachable Terms by Context-Sensitive Innermost Reduction

In this section, we show the concrete definition of \( P_{\text{csin}} \) to construct a TACBB that recognizes the set of reachable terms by innermost reduction of a shallow CS-TRS. \( P_{\text{csin}} \) is a modification of \( P_{\text{cs}} \). The main difference between \( P_{\text{csin}} \) and \( P_{\text{cs}} \) is the number of components of each state of output automata. States of output TA by \( P_{\text{csin}} \) have an extra component that is a state of \( A_{\text{NF}} \). Since \( A_{\text{NF}} \) is TACBB, output automata by \( P_{\text{csin}} \) are also TACBB.

#### Algorithm \( P_{\text{csin}} \):

**Input** A term \( t \) and a shallow CS-TRS \( R = (R, \mu) \) that has no erasing variable.

**Output** A TA \( A_s = (Q_s, Q'_s, \Delta_s) \) such that \( L(A_s) = \frac{t}{R_{\text{in}}}[L(A)] \).

**Step 1 (initialize)** Prepare a TACBB \( A_{\text{NF}} \) obtained by the algorithm in previous section and a TA \( A = (Q, Q', \Delta) \) where each state \( q^s \) accepts \( s \in \{ t \} \cup RS(R) \), \( RS(R) \) is the set of the proper ground subterm of the right-hand sides of \( R \). Here we assume \( Q = \{ q^s \mid s \leq s', s' \in \{ t \} \cup RS(R) \} \), \( Q' = \{ q' \} \), and \( L(R_{\text{in}}, q^s) = \{ s \} \) for all \( q^s \).

**Step 2** Let \( \Delta_0 \) as follows:

(a) \( f(\{q_1, 1, u_1\}, \ldots, \{q_n, 1, u_n\}) \xrightarrow{\Delta} (\{q, 1, u\}) \in \Delta_0 \) where \( f(q_1, \ldots, q_n) \rightarrow q \in \Delta \) and \( f(u_1, \ldots, u_n) \xrightarrow{\Delta} u \in \Delta_{\text{NF}} \), and \( u \in Q_{\text{NF}} \).

(b) \( f(\{q_1, 1, u_1\}, \ldots, \{q_n, 1, u_n\}) \xrightarrow{\Delta} (\{q, 1, u\}) \in \Delta_0 \) where \( f(q_1, \ldots, q_n) \rightarrow q \in \Delta \), \( f(u_1, \ldots, u_n) \xrightarrow{\Delta} u \in \Delta_{\text{NF}} \), and \( i \in \mu(f) \) then \( x_i = a \), otherwise \( x_i = i \).

Let \( \Delta_{k+1} \) be the set of transition rules produced by augmenting transition rules of \( \Delta_k \) by the following inference rules.

1. If there exists \( \sigma : X \rightarrow T(F) \) such that \( f(l_1, \ldots, l_n) = g(r_1, \ldots, r_m) \in R \), \( f(P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \xrightarrow{\Delta} (\{q, 1, u\}) \) and \( x_i = \mu(f) \) then we apply the following inference rules:

\[
f(l_1, \ldots, l_n) \xrightarrow{\Delta} g(r_1, \ldots, r_m) \in R, \quad f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \xrightarrow{\Delta} (\{q, 1, u\})\]

\[
ge((P, x_n, u'_n), \ldots, (P, x_n, u'_n)) \xrightarrow{\Delta} (\{q, 1, u\})\]

Let \( I_j = \{ i \mid l_i = r_j \} \). Each \( P_j, x_j, u_j, c_j \), and \( u \) is determined as follows:

- \( P_j \) = \( P_j \) if \( r_j \in X \), and \( \exists i \in I_j, x_i = i \) and \( \forall i \in I_j, x_i = a \).

- \( u_j = \{ u \} \) otherwise.

- \( c_j = c_1 \cap c_2 \cap c_3 \) that is a satisfiable constraint, where

\[
-c_1 = \bigwedge_{r_i=r_j \in X} i = j
\]

\[
c_2 = \text{is obtained from } c \text{ by replacing equality and disequality between } i \text{ and } j \text{ as follows. Let } i' \text{ and } j' \text{ be as } l_i = r_{i'} \text{ and } l_j = l_{j'}.

\* If \( u_i = u_j \), we replace \( i = j \) in } c \text{ by } \bot \text{ and } i \not= j \text{ by } \top.

\* If \( u_i = u_j \in Q_{\text{NF}} \setminus L_{\text{NF}} \), we consider the following two cases:

\* In the subcase of \( P_i \neq P_j \), we replace \( i = j \) in } c \text{ by } \bot \text{ and } i \not= j \text{ by } \top.

\* In the subcase of \( P_i = P_j \), we replace \( i = j \) in } c \text{ by } \bot \text{ and } i \not= j \text{ by } \top.

\* If \( u_i = u_j \neq Q_{\text{NF}} \setminus L_{\text{NF}} \), we consider the following two cases:

\* In the subcase of \( l_i \neq l_j \text{ and } l_i, l_j \in X \), we replace \( i = j \) in } c \text{ by } i' = j' \text{ and } i \not= j \text{ by } i' \not= j'.

\* Otherwise, we replace \( i = j \) in } c \text{ by } \top \text{ and } i \not= j \text{ by } \bot.

\* If \( r_i \) is not \( i \) for all \( 1 \leq i \leq n \), then we apply the following inference rules:

Note that \( c' \) is not unique because we may choose more than one constraint for \( c_2 \), and also that the role of \( c_2 \) is to preserve the constraints for variables in the rewrite rule applied produced at the inference rule.
(2) If there exists \( \sigma : X \to T(F) \) such that \( f(l_1, \ldots, l_n)_{\sigma} \overset{\Delta_k}{\longrightarrow} f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \) \( \overset{\Delta_k}{\longrightarrow} \{q\}, a, u \) and \( u_i \in Q_{NF} \) or \( x_i = \varepsilon \) for all \( 1 \leq i \leq n \), we apply the following inference rule:

\[
\frac{f(l_1, \ldots, l_n) \rightarrow x \in R, \ f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \overset{\Delta_k}{\longrightarrow} \{q\}, a, u}{\{q\}, a, u \in \Delta_k}
\]

Let \( I = \{i \mid l_i = x\} \). \( P' \) is determined as the following:

- \( P' = \bigcup_{i \in I} P_i \) if \( \exists i \in I. x_i = \varepsilon \) and \( i \)

Step 3 For all states \( (P^1 \cup P^2, a, u) \in Q_s \) where \( P^1 \neq P^2 \), we add the new transition rules to \( \Delta_{k+1} \) as follows:\footnote{This step is almost the same as the step of \( P_{cs} \) because we do not need to be concerned about third components of states in each transition rule \( f((P_1, x^1, u_1), \ldots, (P_n, x^n, u_n)) \overset{\Delta_k}{\longrightarrow} (P, a, u) \).}

1. \( f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \overset{\Delta_k}{\longrightarrow} (P^1 \cup P^2, a) \in \Delta_{k+1} \) where

   \[
   P_i = \begin{cases} 
   P_i, & \text{if } x_i^j = \varepsilon \text{ for some } j \in \{1, 2\} \text{ and } P_i \neq P^1 \cup P^2; \\
   P_i \cup P^2, & \text{if } x_i^j = x_i^k \neq \varepsilon \text{ for } j \neq k \end{cases}
   \]

   \( x_i = \varepsilon \) if some \( x_i^j = \varepsilon \) for some \( j \in \{1, 2\} \) and \( x_i = \varepsilon \) otherwise.

   \( c' = c_1 \land c_2 \).

2. \( (P^1 \cup P^2, a, u) \overset{\Delta_k}{\longrightarrow} (P_1, a, u) \in \Delta_{k+1} \) if \( (P_1, a, u) \overset{\Delta_k}{\longrightarrow} (P_1, a, u) \in \Delta_k \), and,\( (P^1, a, u) \overset{\Delta_k}{\longrightarrow} (P_2, a, u) \in \Delta_k \) 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.

We show an example that shows how \( P_{cs} \) works in Appendix A.1.2. This procedure \( P_{cs} \) eventually terminates at some \( k \) and apparently \( \Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots \) similarly to \( P_{cs} \).

In the following, we show the correctness of \( P_{cs} \).

First, we show several propositions. Since Propositions 28–31 below are similar to the case of \( P_{cs} \), we abbreviate their proofs.

**Proposition 28** Let \( s \in T(F) \). Then \( s \overset{\Delta_k}{\longrightarrow} q' \in Q \) if \( s \overset{\Delta_k}{\longrightarrow} \{q'\}, a, u \) for some \( u \in Q_{NF} \).

**Proposition 29** Let \( t \in T(F) \). For any \( k \), if \( t \overset{\Delta_k}{\longrightarrow} (P, a, u) \), then \( t \overset{\Delta_k}{\longrightarrow} (P, a, u) \). Moreover, \( P \) is of the form \( \{q\} \).

**Proposition 30** Let \( t \in T(F) \). Then, for any \( k \), \( t \overset{\Delta_k}{\longrightarrow} (P, a, u) \) iff \( t \overset{\Delta_{k+1}}{\longrightarrow} (P, a, u) \).

**Proposition 31** Let \( t \in T(F) \). Then, \( t \overset{\Delta_k}{\longrightarrow} (P, a, u) \) implies \( t \overset{\Delta_{k+1}}{\longrightarrow} (P, a, u) \).

**Proposition 32** If the rule \( f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \overset{\Delta_k}{\longrightarrow} (P, a, u) \in \Delta_{k+1} \) then it is also in \( \Delta_0 \). Moreover, \( x_i = \varepsilon \) for all \( 1 \leq i \leq n \).

**Proposition 33** If the rule \( f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \overset{\Delta_k}{\longrightarrow} (P, a, u) \) in \( \Delta_{k+1} \), then \( i \in \mu(f) \) implies \( x_i = \varepsilon \).

**Proposition 34** For any \( k \), if \( a : t \overset{\Delta_k}{\longrightarrow} (P, x, u) \), then \( t \overset{\Delta_{k+1}}{\longrightarrow} u \).

**Proposition 35** For any \( k \), if \( a : t \overset{\Delta_k}{\longrightarrow} (P, x, u) \), then \( t \overset{\Delta_{k+1}}{\longrightarrow} u \).

**Proposition 36** We show the proof by induction on \( |\alpha| \). If the last transition rule applied in \( \alpha \) is of the form \( (P', x, u) \overset{\Delta_k}{\longrightarrow} (P, x, u) \), then we have \( t \overset{\Delta_{k+1}}{\longrightarrow} u \) from the induction hypothesis. Otherwise, let the last transition rule applied in \( |\alpha| \) is \( f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \overset{\Delta_k}{\longrightarrow} (P, x, u) \).

If \( f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \overset{\Delta_k}{\longrightarrow} (P, x, u) \in \Delta_k \), then we have...
In the case where the last transition rule applied in Lemma 35

We show this lemma by induction on $u$.

Proof: Similar to the proof of Lemma 9.

Lemma 36 Let $s, t \in T(F)$, $s \xrightarrow{a} \langle Pxu \rangle$, and $t \xrightarrow{a} \langle Px'u' \rangle$. Then, $P = P'$ iff $s = t$.

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

(1) Consider the case where the last transition rule applied in $\alpha$ is (of the form)

$$f(|P_1, x_1, u_1|, \ldots, |P_n, x_n, u_n|) \xrightarrow{\alpha} \langle P, a, u' \rangle$$

Then $\alpha$ can be represented as $t \xrightarrow{a} f(|P_1, x_1, u_1|, \ldots, |P_n, x_n, u_n|) \xrightarrow{\alpha} \langle P, a, u' \rangle$.

In this case, the position $p$ can be represented as $ip^l$ for $1 \leq i \leq m$.

From the construction of the transition rule

$$f(|P_1, x_1, u_1|, \ldots, |P_n, x_n, u_n|) \xrightarrow{\alpha} \langle P, a, u' \rangle$$

we have the transition rule $f(|u_1, \ldots, u_n|) \xrightarrow{\alpha} \langle P, a, u' \rangle$.

First, from $i \in \mu(g)$ and Proposition 27, we have $u_i \in Q_{\alpha}^F$, and hence we also have $u \in Q_{\alpha}^F$ from the induction hypothesis.

(2) In the case where the last transition rule applied in $\alpha$ is (in the form of)

$$\langle Px_0, u_0 \rangle \xrightarrow{\alpha} \langle P, a, u \rangle \in \Delta_k$$

we have $u' = u$ from the construction of $\Delta_0$ or the second inference rule of Step 2. Hence this lemma holds by the induction hypothesis.

Lemma 37 Let $j \notin \mu(g)$ and $g(\ldots, \langle P_j', x_j', u_j' \rangle, \ldots) \xrightarrow{\alpha} \langle P, x', u' \rangle \in \Delta_k$, then $u_j' \notin Q_{\alpha}^F$ or $x_j' = 1$.

Proof:

(1) If $k = 0$, then $x_j' = 1$ from the construction of $\Delta_0$.

(2) Consider the case of $k > 0$. We can assume $g(\ldots, \langle P_j', x_j', u_j' \rangle, \ldots) \xrightarrow{\alpha} \langle P, x', u' \rangle \in \Delta_k \setminus \Delta_{k-1}$ without loss of generality. This rule is introduced by (1) of Step 2 or (1) of Step 3. In the latter case, if $x_j' = 1$, then we have $g(\ldots, \langle P_j', a, u_j' \rangle, \ldots) \xrightarrow{\alpha} \langle q_j, x_j, u_j \rangle \in \Delta_k$ for any $q_j \in P$ where this rule is in $\Delta_0$ 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, $x_j' = a$ implies $x_j' = a$ from Proposition 32, and there exist $f(|l_1, \ldots, l_n|) \rightarrow g(|r_1, \ldots, r_m|) \in R$ and $f(|P_1, x_1, u_1|, \ldots, |P_n, x_n, u_n|) \xrightarrow{\alpha} \langle P, a, u \rangle \in \Delta_{k-1}$ where $u_i \in Q_{\alpha}^F$ and $x_i = 1$ for all $1 \leq i \leq n$. If $j \notin \mu(g)$ and $x_j' = a$, then there exists some $i$ such that $u_i = u_j'$ and $x_i = a$ for all $i'$ such that $l_i' = r_j$. Hence we have $u_i = u_j' \notin Q_{\alpha}^F$.

Lemma 38 Let $\alpha : t[\nu] p \xrightarrow{\Delta} t[\langle P, a, u \rangle] p \xrightarrow{\Delta} \langle P', a, u \rangle$. If $u \in Q_{NF} \setminus Q_{\alpha}^F$ and $p \in Pos^u(t)$, then there exists $v' \in Q_{NF}$ such that $t[v'] p \xrightarrow{\Delta} t[\langle P, a, v \rangle] p \xrightarrow{\Delta} \langle P', a, v' \rangle$ 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 $\alpha$ is (of the form)

$$g(|P_1, x_1, u_1|, \ldots, |P_m, x_m, u_m|) \xrightarrow{\alpha} \langle P', x', u' \rangle \in \Delta_k$$

Then $\alpha$ can be represented as $t[\nu] p \xrightarrow{\Delta} t[\langle P, x, u \rangle] p \xrightarrow{\Delta} g(|P_1, x_1, u_1|, \ldots, |P_m, x_m, u_m|) \xrightarrow{\Delta} \langle P', x', u' \rangle$.

Let $p = jp^l$ where $1 \leq j \leq m$.

If the rule $g(|P_1, x_1, u_1|, \ldots, |P_m, x_m, u_m|) \xrightarrow{\alpha} \langle P, x, u \rangle$ is in $\Delta_0$, the rule is produced at (2) of Step 1 of Pcons. Therefore, for any $u'' \notin Q_{NF}$, there exists the constraints $c''$ and $u'' \in Q_{NF}$ such that $g(|P_1, x_1, u_1|, \ldots, |P_m, x_m, u_m|) \xrightarrow{\alpha} \langle P, x, u'' \rangle \in \Delta_0$ where $t$ satisfies $c''$ from the completeness of $A_{NF}$. Hence this lemma holds from the induction hypothesis.

Consider the case where the rule $g(|P_1, x_1, u_1|, \ldots, |P_m, x_m, u_m|) \xrightarrow{\alpha} \langle P, x, u \rangle$ is in $\Delta_k \setminus \Delta_{k-1}$ for $k > 0$. In this case, $j \notin \mu(g)$ from $p \in Pos^u(t)$, and we have $x_j' = a$ from $x = a$ and Proposition 33. For $\alpha_{ij} : (t[i])t[i'] p \xrightarrow{\Delta} (t[i])(|P, a, u|) p \xrightarrow{\Delta} (P_j', a, u_j')$, we have $(t[i])t[i'] p \xrightarrow{\Delta} (t[i])(|P, a, v|) p \xrightarrow{\Delta} \langle P_j', a, v_j' \rangle$ for some $v_j' \in Q_{NF}$ from the induction hypothesis. Note that we have $u_j' \notin Q_{NF}$ from $u \notin Q_{NF}$ and Lemma 36. Thus, we prove there exists the transition rule $g(|P_1, x_1, u_1|, \ldots, |P_{m'}, x_{m'}, u_{m'}|) \xrightarrow{\alpha} \langle P, x, v' \rangle \in \Delta_\alpha$.

Here, there are two cases in which the rule $g(|P_1, x_1, u_1|, \ldots, |P_{m'}, x_{m'}, u_{m'}|) \xrightarrow{\alpha} \langle P, x, v' \rangle \in \Delta_\alpha$. 

IPSJ Transactions on Programming Vol. 4 No. 4 12–35 (Sep. 2011) © 2011 Information Processing Society of Japan
Lemma 39 If \( (P_1', a, u) \xrightarrow{\Delta^*} (P, a, u) \) and \( p \in \text{Pos}^p(t) \), then there exists \( (P', a, u') \) such that \( t' \xrightarrow{\Delta^*} (P', a, u') \) and \( t' \downarrow (P', a, u') \) implies \( t' \xrightarrow{\Delta^*} (P, a, u) \).

Proof: Similar to the proof of Lemma 10.

Proofs of the following Lemmas 40–43 are similar to the ones of Lemma 11–14, because we do not need to consider the third components of the states.

Lemma 40 If \( (P'_1, a, u) \xrightarrow{\Delta^*} (P', a, u) \) and \( (P'_2, a, u) \xrightarrow{\Delta^*} (P_, a, u) \), then we have \( (P'_1 \cup P'_2, a, u) \xrightarrow{\Delta^*} (P_1 \cup P_2, a, u) \).

Proof: Similar to Lemma 11.
we show that if each component of states is determined as the definition of $P_{\text{sin}}$ have

However, we must show that the term $g(r_1, \ldots, r_m)\sigma$ satisfies the constraint $c'$, which is the point that the proof of Lemma 15 does not have. Here, we show that if $f(l_1, \ldots, l_n)\sigma$ satisfies the constraint $c$ of the transition rule $f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \xrightarrow{\sigma}\{q, a, u\}$, 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 $P_{\text{csin}}$.

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

(2) Here, we show 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 $\bot$.

- For $i$ and $j$ such that $u_i \neq u_j$, we have $l_i\sigma \neq l_j\sigma$ from Lemma 34 and the determinacy of $\mathcal{A}_{\text{NF}}$. 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 $\bot$ in $c_2$.

- For $i$ and $j$ such that $u_i = u_j \in Q_{\text{NF}} \backslash Q_{\text{NF}}$, we have $x_i = x_j = \bot$ and hence $l_i\sigma \xrightarrow{\Delta_0} (P_i, x_i, u)\sigma$ and $l_j\sigma \xrightarrow{\Delta_0} (P_j, x_j, u)\sigma$ from Proposition 29. Therefore, we have $P_j = P_j$ if $l_i\sigma = l_j\sigma$ from Lemma 35. 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 $i = j$ in $c$ by $\bot$ in $c_2$ if $P_i = P_j$ and $i = j$ in $c$ by $\bot$ in $c_2$ if $P_i \neq P_j$.

- For $i$ and $j$ such that $u_i = u_j \in Q_{\text{NF}}$, 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 $l_i\sigma = l_j\sigma$ if $r_i\sigma = r_j\sigma$. Thus, we have $f(l_1, \ldots, l_n)\sigma$ satisfies $i = j$ in $c$ if $g(r_1, \ldots, r_m)\sigma$ satisfies $i' = j'$ in $c_2$, and $f(l_1, \ldots, l_n)\sigma$ satisfies $i \neq j$ in $c$ if $g(r_1, \ldots, r_m)\sigma$ 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 $\bot$ in $c_2$.

- If $l_i \notin X$, then we have $l_i = l_j\sigma$ from Lemma 34 and Proposition 26.

  Thus, there is no problem replacing $i \neq j$ in $c$ by $\bot$ 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 $s(l_1, \ldots, l_n)\sigma$ and therefore, there is no problem replacing these constraints by $\bot$.

Moreover, we have a constraint $c_3$ that is satisfied by $g(r_1, \ldots, r_m)\sigma$ from the completeness of $\mathcal{A}_{\text{NF}}$. Thus, we have a constraint $c'$ that is satisfied by $g(r_1, \ldots, r_m)\sigma$ and hence we have the transition $g(r_1, \ldots, r_m)\sigma \xrightarrow{\Delta_0} g((P_1', x_1', u_1'), \ldots, (P_m', x_m', u_m')) \xrightarrow{\sigma}\{q, a, u\}$.

The following lemma shows the completeness of $P_{\text{csin}}$.

**Lemma 45** Let $\mathcal{R}$ be shallow. Then $\mathcal{L}(A_s) \supseteq \xrightarrow{\mathcal{R}}_{\mathcal{L}}\{\mathcal{L}(A)\}$.

**Proof:** Let $s \xrightarrow{\mathcal{R}}_{\mathcal{L}} t$ and $s \xrightarrow{\Delta_0} q \in Q'$. Since $s \xrightarrow{\Delta_0} \{\{q, i, u\} \in Q_{\mathcal{L}}^t$ from Proposition 28, we have $s \xrightarrow{\Delta_0} \{\{q, a, u\} \in Q_{\mathcal{L}}^t$ by Proposition 30. Hence $t \xrightarrow{\Delta_0} \{\{q, a, u\} \in Q_{\mathcal{NF}}$ by Lemma 44.

Next we define the measure and order of transition in order to prove the soundness. These definitions are similar to the case of $P_{\text{cs}}$.

**Definition 46** Let $\|t \xrightarrow{\Delta_0} (P, x, u)\|$ be the sequence of integer defined as follows:

$\|t \xrightarrow{\Delta_0} (P, x, u)\| =$

$\|t \xrightarrow{\Delta_0} (P', x', u')\| \cdots$ if $t \xrightarrow{\Delta_0} (P', x', u')$ by $\Delta_0 \xrightarrow{\Delta_{\mathcal{L}}} (P, x, u)$

$\|t = f(l_1, \ldots, l_n) \xrightarrow{\Delta_0} (P_1, x_1, u_1, \ldots, P_n, x_n)\|$

$\|t \xrightarrow{\Delta_0} (P_1, x_1, u_1)\| \cdots$

$\forall i \neq j.\|t \xrightarrow{\Delta_0} (P_i, x_i, u_i)\|$

$\|t \xrightarrow{\Delta_0} (P_j, x_j, u_j)\|$

$\|t \xrightarrow{\Delta_0} (P_1, x_1, u_1)\|

$\|t \xrightarrow{\Delta_0} (P_j, x_j, u_j)\|$

The order $\sqsubseteq$ and $\sqsubseteq$ for transition sequences is defined similarly to Definition 18.

The following lemma is the key lemma to prove soundness of $P_{\text{csin}}$.

**Lemma 47** Let $\Delta_0$ be generated from a shallow CS-TRS $\mathcal{R}$. Then $\alpha : t \xrightarrow{\Delta_0} (P, x, u')$ implies that both $s \xrightarrow{\mathcal{R}}_{\mathcal{L}} t$ and $\beta : s \xrightarrow{\Delta_0} \{\{q, i, u\} \in Q_{\mathcal{L}}^t$ for some term $s$, $q \in P$, and $u \in Q_{\mathcal{NF}}$. 

IPSJ Transactions on Programming  Vol. 4  No. 4  12–35 (Sep. 2011) © 2011 Information Processing Society of Japan
Proof: Similarly to Lemma 44, we describe some points of the proof in Appendix A.3.2. Here we show the proof in the case of the last transition rule in $\sigma$ is in $\Delta_k \setminus \Delta_{k-1}$ for $k > 0$ and $|P| = 1$. We abbreviate the proof in the case for $k = 0$ or $|P| > 0$ because it is similar to the proof of Lemma 19.

Assume that $t = (t_1, \ldots, t_m)$ and the last transition rule in $\alpha$ is $g((P_1', x_1', u_1'), \ldots, (P_m', x_m', u_m')) \xrightarrow{\Delta} \langle q, x, u' \rangle \in \Delta_k \setminus \Delta_{k+1}$. Since this rule is introduced at (1) of Step 2, there exist $f(l_1, \ldots, l_n) \rightarrow (r_1, \ldots, r'_m) \in R$, $f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \xrightarrow{\Delta} \langle q, a, u \rangle \in \Delta_k - 1$ such that $u_i \in Q_{\text{NF}}$ or $x_i = 1$ for all $1 \leq i \leq n$. Let $\sigma' : F \rightarrow \mathcal{T}(F)$ such that $\alpha' \xrightarrow{\Delta_{k-1}} \langle P_1, x_1, u_1 \rangle$, and $(P_j', x_j', u_j') \xrightarrow{\Delta} \langle q, a, v \rangle$ are given as the definition of $\text{P}_{\text{cin}}$. Then, we have the substitution $\sigma$ such that $g(r_1, \ldots, r_m) \xrightarrow{\sigma} g(t_1, \ldots, t_m)$ and $\alpha' : g(r_1, \ldots, r_m) \xrightarrow{\Delta} g((P_1', x_1', v_1'), \ldots, (P_m', x_m', v_m')) \xrightarrow{\Delta} \langle q, a, v' \rangle$ for some $v'$, where $\sigma' \subseteq \alpha$ similarly to the proof of Lemma 19 (see Appendix A.3.2).

Note that the substitution $\sigma$ is well-defined because for all $r_j \in X$ such that there exists $i$ such that $l_i = r_j$ and $x_i = 1$, we have the term $s_j \xrightarrow{\Delta} t_j$ for $j \notin \mu(g)$ such that $s_j \xrightarrow{\Delta} t_j$ and $s_j \xrightarrow{\Delta} \langle P_j', x_j', u_j' \rangle$. Since there is no term other than $s_j$ that transits to $(P_j', x_j', u_j')$, 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 $x_i = 1$, the constraint $\sigma$ of (1) in the procedure) has the equality that implies all $t_j$'s are the same for such $j$.

Next we show that $g(r_1, \ldots, r_m) \xrightarrow{\sigma} \langle q, \alpha, v \rangle$ satisfies $c_1$ and $c_2$ of $\sigma$ defined as the definition of $\text{P}_{\text{cin}}$.

Obviously, $g(r_1, \ldots, r_m) \xrightarrow{\sigma} \langle q, \alpha, v \rangle$ satisfies $c_1$ because $\alpha = \sigma = \sigma$ for all $r_j \in X$. Moreover, it is not so difficult to show that $g(r_1, \ldots, r_m) \xrightarrow{\sigma} \langle q, \alpha, v \rangle$ satisfies $c_2$. This is because for all $r_j \neq t_j$, there exists $i$ such that $l_i = r_j$ and $x_i = 1$ and we have $u_i' = u_i$ from Lemma 34 and the determinacy of $\text{D}_{\text{NF}}$. In this case, we have $u_i = u_i' \notin Q_{\text{NF}}$ and hence there is no equality or disequation that constrains such $j$. From the completeness of $\text{D}_{\text{NF}}$, we have $c_j$ that is satisfied by $g(r_1, \ldots, r_m) \xrightarrow{\sigma} \langle q, \alpha, v \rangle$. Thus we have the transition rule $g((P_1', x_1', v_1'), \ldots, (P_m', x_m', v_m')) \xrightarrow{\sigma} \langle q, a, v' \rangle$ where $g(r_1, \ldots, r_m) \xrightarrow{\sigma} \langle q, a, v' \rangle$.

On the other hand, we have $f(l_1, \ldots, l_n) \xrightarrow{\sigma} g(r_1, \ldots, r_m) \sigma$. Here, we show that we can construct $\beta : f(l_1, \ldots, l_n) \xrightarrow{\Delta} f((P_1, x_1, u_1), \ldots, (P_n, x_n, u_n)) \xrightarrow{\Delta_{k-1}} \langle q, a, v \rangle$ and hence $\beta \subseteq \alpha$. For $l_i \notin X$, $l_i \sigma = l_i \xrightarrow{\Delta_{k-1}} (P_i, x_i, u_i)$ from Step 2 of $\text{P}_{\text{cin}}$. For $l_i \in X$ and there exists $h$ such that $l_h = l_i$ and $x_h = 1$, since there is no term other than $l_h \sigma$ that transits to $\langle P_h, x_h, u_h \rangle$ from Lemma 35, we have $l_i \sigma = l_i \sigma'$ and hence $l_i \sigma \xrightarrow{\Delta_{k-1}} (P_i, x_i, u_i)$. For $l_i \in X$ such that there is no $h$ such that $l_h = l_i$ and $x_h = 1$, we have $l_i \sigma \xrightarrow{\Delta_{k-1}} (P_i, x_i, u_i)$ from Lemma 42. In the following, we show that if $(r_1, \ldots, r_m) \sigma$ satisfies $c'$ then $f(l_1, \ldots, l_n) \sigma$ satisfies $c$.

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

- Consider the case where $\tau$ in $c_2$ is obtained by replacing $i = j$ in $c$ where $u_i \neq u_j$. In this case, we have $f(l_1, \ldots, l_n) \sigma$ satisfies $c'$.\n
- Consider the case where $\tau$ in $c_2$ is obtained by replacing $i = j$ or $i \neq j$ in $c$ where $u_i = u_j \in Q_{\text{NF}} \setminus Q_{\text{NF}}$. Then, we have $x_i = x_j = 1$. 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 29 and Lemma 35, and if $P_i \neq P_j$ then we have $i = j$ but $f(l_1, \ldots, l_n) \sigma$ satisfies it.

- Consider the case where $\tau$ in $c_2$ is obtained by replacing $i = j$ in $c$ where $u_i = u_j \in Q_{\text{NF}}$.\n
\[\begin{align*}
\text{If } l_i &= l_j \in X, \text{ we have } i = j \text{ in } c \text{ but } f(l_1, \ldots, l_n) \sigma \text{ satisfies it trivially.} \\
\text{If } l_i &\neq l_j \text{ and } l_i, l_j \in X, \text{ then } c \text{ does not have equality or disequation replaced by } \tau \text{ in } c_2. \\
\text{If } l_i &\notin X, \text{ we have } i = j \text{ in } c \text{ but we have } l_i = l_j \sigma \text{ from Lemma 34 and Proposition 26.}
\end{align*}\]

Moreover, we have $i = j$ or $i \neq j$ in $c$ for $i' = j' = i'$ or $i' \neq j'$ in $c_2$. These kinds of constraints are satisfied by $f(l_1, \ldots, l_n) \sigma$ similarly to the statement in Lemma 44.

Since $u_i \in Q_{\text{NF}}$ or $x_i = 1$ for all $i$, $l_i \sigma$ is a normal form or $i \notin \mu(f)$ for each $i$ from Lemma 34 and the procedure. Hence we have $f(l_1, \ldots, l_n) \sigma \xrightarrow{\text{cin}} g(r_1, \ldots, r_m) \sigma$. Here $\alpha' \subseteq \alpha \subseteq \beta$ follows. Thus, we have $s \xrightarrow{\text{cin}} f(l_1, \ldots, l_n) \sigma \xrightarrow{\text{cin}} g(r_1, \ldots, r_m) \sigma \xrightarrow{\text{cin}} g(t_1, \ldots, t_m) = t$ and $s \xrightarrow{\Delta} \langle q, a, u \rangle$ for some $u$ by the induction hypothesis.

If a CS-TRS has an erasing variable, we cannot prove Lemma 47 as the above proof. Assume that the transition rule $g((P_1', x_1', u_1'), (P_m', x_m', u_m')) \xrightarrow{\text{cin}} \langle q, a, u' \rangle \in \Delta_{k+1}$ is produced from $f(l_1, \ldots, l_n) \rightarrow (r_1, \ldots, r_m) \in R$ and $f((P_1', x_1', u_1'), (P_m', x_m', u_m')) \xrightarrow{\text{cin}} \langle q, a, u \rangle \in \Delta_{k-1}$. If we have the equality $\langle q, a, u \rangle$
For some $q$ 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 soundness of $P_{\text{csin}}$.

**Lemma 48** If $R$ be shallow, then $L(A_i) \subseteq \triangleleft \rightarrow = \rightarrow_{\text{in}} \rightarrow L(A)$.

**Proof:** Let $t \xrightarrow{\Delta} \langle P, x, u' \rangle \in Q_f$ then we have $s \xrightarrow{\rightarrow_{\text{in}}} \rightarrow t$ and $s \xrightarrow{\Delta} \rightarrow \langle \{q\}, x, u \rangle \in Q_f$ for some $q \in P$ from Lemma 47. Since $s \xrightarrow{\rightarrow_{\text{in}}} \rightarrow \langle \{q\}, i, u \rangle$ from Proposition 30, we have $s \xrightarrow{\rightarrow} \rightarrow q \in Q_f$ from Proposition 28. Finally we obtain the following theorems from Lemma 45 and 48.

**Theorem 49** For any shallow CS-TRS $R$, we can construct a TACBB recognizing the set of terms that is innermost reachable from a term. Thus, innermost reachability is decidable for shallow CS-TRSs.

However, in general, we cannot always construct a TACBB recognizing the innermost reachable set from a regular set of terms for a CS-TRS, while we can construct a TACCB in the case of the ordinary TRS. Another future work is to find other subclasses that reachability, innermost reachability, or reachability of other strategies is decidable for TRSs or CS-TRSs. One of the candidates is reachability for right-linear finite pass overlapping CS-TRSs where it is known that reachability is decidable for ordinary TRSs in Ref. 17. However, the class is complex and hence we think this is not easy. Innermost reachability for right-linear right-shallow TRSs is also a candidate. In the case of this class, to recognize the set of normal forms, we need TA with equality or disequality constraints. This automata has more complex constraints than that of TACBB and sometimes more complex constraints than constraints nests. Therefore, we think that this problem is much more complex than the result of this paper. Moreover, outermost reachability is a candidate. Outermost reduction is a strategy that rewrites outermost redexes. Today, no class is known such that outermost reachability is decidable.

**Acknowledgments** We would like to thank the anonymous referees for their helpful comments and remarks. This work is partly supported by MEXT. KAKENHI #10J08166, #20300010, #20500008, and #21700011.

**References**

5. Durand, I. and Sénizergues, G.: Bottom-Up Rewriting Is Inverse Recognizability Preserving, Proc. 18th International Conference on Term Rewriting and Applica-

Appendix

A.1 Examples for Section 4

A.1.1 An Example of TACBB Accepting the Set of Context-sensitive Normal Forms

Let CS-TRS \( R = (R, \mu) \) be as \( R = \{ a \rightarrow b, a \rightarrow c, f(x, b) \rightarrow g(x, a), g(x, x) \rightarrow h(x, x) \} \) and \( \mu(f) = \emptyset, \mu(g) = \{1, 2\}, \mu(h) = \{1, 2\} \). We construct the TACBB \( A_{\text{NF}} \) such that \( L(A_{\text{NF}}) = \text{CS-NF}_R \) by the algorithm shown in Section 4.1.

First, we construct a deterministic TACBB \( A_a, A_f(x, a) \), and \( A_g(x, x) \) at the first step of the algorithm.

The set of final states of \( A_a \) is \( Q_a^f = \{ U^o \} \) and the set of transition rules is \( \Delta_a = \{ a \rightarrow U^0, b \rightarrow U_1, c \rightarrow U_1, f(U, U) \rightarrow U_1, g(U_1, U_1) \rightarrow U_1, g(U_1, U_2) \rightarrow U_1, h(U_1, U_1) \rightarrow U_1, h(U_1, U_2) \rightarrow U_1 \} \) where \( U_1, U_2 \in \{ U_1, U_2 \} \) and one of \( U_1 \) and \( U_2 \) is \( U^0 \).

The set of final states of \( A_f(x, a) \) is \( Q_f(x, a) = \{ U^0 \} \) and the set of transition rules is \( \Delta_f(x, b) = \{ a \rightarrow U_1, b \rightarrow U_b, c \rightarrow U_1, f(U, U_1) \rightarrow U_1, f(U, U^o) \rightarrow U_1, g(U_1', U_2') \rightarrow U_1, g(U_1', U_2) \rightarrow U_1, h(U_1', U_2') \rightarrow U_1, h(U_1', U_2) \rightarrow U_1 \} \) where \( U_1, U_2 \in \{ U_1, U_2 \} \), \( U', U'_1, U'_2 \in \{ U_1', U_2' \} \), and one \( U_1 \) or \( U_2 \) is \( U^0 \).

The set of final states of \( A_g(x, x) \) is \( Q_g(x, x) = \{ U^0 \} \) and the set of transition rules is \( \Delta_g(x, x) = \{ a \rightarrow U_1, b \rightarrow U_b, c \rightarrow U_1, f(U, U) \rightarrow U_1, g(U_1, U_1) \rightarrow U_1, g(U_1, U_2) \rightarrow U_1, h(U_1, U_1) \rightarrow U_1, h(U_1, U_2) \rightarrow U_1 \} \) where \( U_1, U_2 \in \{ U_1, U_2 \} \) and one \( U_1 \) or \( U_2 \) is \( U^0 \).

At the second step, we construct the TACBB \( A' \) accepting all unions of \( A_a, A_f(x, a) \), and \( A_g(x, x) \).

The set of final states of \( A' \) is \( Q' = \{ U_1, U_2, U_3 \} \) where \( U_1, U_2, U_3 \in \{ U_1', U_2' \} \) and one \( U_1, U_2, \) or \( U_3 \) is \( U^0 \).
The set of transition rules of $A'$ is $\Delta' = \{ a \xrightarrow{u} \langle U^0, U_\bot, U_\bot \rangle, b \xrightarrow{U_\bot, U_\bot} \langle U_\bot, U_\bot, U_\bot \rangle, f(U, U_\bot) \xrightarrow{U_\bot, U_\bot, U_\bot} \langle U_\bot, U_\bot, U_\bot \rangle, f(U, U_\bot) \xrightarrow{U_\bot, U_\bot, U_\bot} \langle U_\bot, U_\bot, U_\bot \rangle, g(U, U_\bot, U_\bot) \xrightarrow{U_\bot, U_\bot, U_\bot} \langle U_\bot, U_\bot, U_\bot \rangle, g(U, U_\bot, U_\bot) \xrightarrow{U_\bot, U_\bot, U_\bot} \langle U_\bot, U_\bot, U_\bot \rangle, h(U, U_\bot, U_\bot) \xrightarrow{U_\bot, U_\bot, U_\bot} \langle U_\bot, U_\bot, U_\bot \rangle \}$. We abbreviate the conversion to complete and reduced TACBB because the number of transition rules becomes huge. Let $A''$ be the TACBB obtained by converting $A'$ to a complete and reduced TACBB.

Finally, at the third step of the algorithm, we obtain $A_{NF}$ from $A''$ by replacing the final state. We show the set of final states and the set of transition rules of $A_{NF}$ in the following. However, since $A_{NF}$ originally obtained from the algorithm is huge, we show a minified one. If we minify TACBB obtained by the algorithm, Proposition 26 may not hold. Therefore, we should not minify the TACBB obtained by the algorithm. In the case of following TACBB, Proposition 26 holds.

The set of states is $Q_{NF} = \{ u_b, u_\bot, u_c \}$, the set of final states is $Q_{NF}^F = \{ u_b, u_\bot \}$, and the set of transition rules is $\Delta_{NF} = \{ a \xrightarrow{u} b, b \xrightarrow{u} a, c \xrightarrow{u} \bot, g(u, u) \xrightarrow{u} u, g(u, u) \xrightarrow{u} u, g(u, u) \xrightarrow{u} u, h(u, u) \xrightarrow{u} u, h(u, u) \xrightarrow{u} u, h(u, u) \xrightarrow{u} u \}$ where $u_1, u_4 \in Q_{NF}$, $u_2 \in \{ u^c, u_\bot \}$, $u_3 \in Q_{NF}^F$, and $u_1 \neq u_4$.

A.1.2 An Example of TACBB Obtained by $P_{csin}$

Let CS-TRS $R$ be the CS-TRS of A.1.1. We input the term $f(a, b)$, and the shallow CS-TRS $R$ to $P_{csin}$. Here, we have $\overline{\mathcal{R}}[[f(a, b)]] = \{ f(a, b), g(a, a), g(b, a), g(a, b), g(b, b), g(c, a), g(a, c), g(b, c), g(c, b), h(b, b), h(c, c) \}$.

In the initializing step, at (1) of Step 1 of $P_{csin}$, we have the TA $A = (Q, Q^I', \Delta')$ where $Q = \{ q^a, q^b, q^{f(a,b)} \}$, $Q^I = \{ q^{f(a,b)} \}$, and $\Delta = \{ a \rightarrow q^a, b \rightarrow q^b, f(q^a, q^b) \rightarrow q^{f(a,b)} \}$, and TACBB $A_{NF}$ as a previous subsection. At (2) of Step 1, we have $Q_{NF} = \{ (P, a, u), (\{ P \}, b, u) \}$ for $P \subseteq Q$, $P \neq ?$, $p \in Q$, and $u \in Q_{NF}$, $Q_{NF}^F = \{ (P^f, a, u) \}$ where $P^f \cap Q^I' \neq ?$ and $u \in Q_{NF}$, and $\Delta_0$ is as follows:

$\Delta_0 = \{ a \rightarrow (\{ q^a \}, x, u^c), b \rightarrow (\{ q^b \}, x, u^c), f(\{ q^a \}, x, u^c), f(\{ q^b \}, x, u^c) \}$

where $x \in \{ a, i \}$, $u_1 \in Q_{NF}$, and $u_2 \in \{ u^c, u_\bot \}$.

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

$\{ b \xrightarrow{u} (\{ q^a \}, x, u^c), c \xrightarrow{u} (\{ q^b \}, x, u^c), g(\{ q^a \}, a, u_3), g(\{ q^a \}, a, u_3) \} \xrightarrow{\Delta_{NF}} (\{ q^{f(a,b)} \}, a, u^c), g(\{ q^b \}, a, u_3), g(\{ q^b \}, a, u_3) \xrightarrow{\Delta_{NF}} (\{ q^{f(a,b)} \}, a, u^c) \}$

where $x \in \{ a, i \}$, and $u_3 \in Q_{NF}$, and $\{ b \xrightarrow{u} (\{ q^a, q^b \}, x, u_3) \}$ at Step 2.

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

$\{ h(\{ q^a \}, a, u_3), \} \xrightarrow{\Delta_{NF}} (\{ q^{f(a,b)} \}, a, u^c) \}$

where $x \in \{ a, i \}$ and $u_3 \in Q_{NF}$, and $\{ b \xrightarrow{u} (\{ q^a, q^b \}, x, u_3) \}$ at Step 3 where $x \in \{ a, i \}$.

The saturation step at $k = 2$, and we have $\Delta_2 = \Delta_1$. TA $A_2 = (Q_2, Q^I_2, \Delta_2)$ holds that $L(A_2) = \overline{\mathcal{R}}[\{ f(a, b) \}]$.

A.2 Concrete Proofs of Lemma 42 and 43

A.2.1 Lemma 42

Proof: Similarly to the proof of Lemma 12, we give the proof for $m = 2$ by induction on $|t|$

Let $t = t(t_1, \ldots, t_n)$. Then, each transition sequence is represented as $f(t_1, \ldots, t_n) \xrightarrow{\Delta} \langle P^f_1, x_{1,1}, \ldots, P^f_1, x_{1,n}, u_1 \rangle \xrightarrow{\Delta} \langle P^f_1, a, u \rangle \xrightarrow{\Delta} \langle P^f_1 \cup P^f_2, a, u \rangle.$

Thus, we have $f(t_1, \ldots, t_n) \xrightarrow{\Delta} \langle P^f_1 \cup P^f_2, a, u \rangle$.

From (1) of Step 3 of $P_{csin}$, we have the transition rule $f(\{ P_1, x_{1,1}, \ldots, P_n, x_{n,n}, u_n \}) \xrightarrow{\Delta} \{ P, a, u \} \in \Delta_2$ where

- $P_i = \{ P^f_1 \}$ if $x_{1,j} = i$ for some $j \in \{ 1, 2 \}$ and
- $P^f_i \cup P^f_j \cdots$ if $x_{1,j} = a$
Let $i = 1$ and $j = 1$. Hence, we have $\sigma_i \rightarrow \Delta_i$ for all $i \in \{1, 2\}$ from (1) of Step 3 of $P_{in}$ and $P_{in}$, $\sigma$'s, and $\sigma$'s are represented as the following:

**A.2.2 Lemma 43**

**Proof:** We show this lemma by induction on $|t|$. We can assume that the transition $t \rightarrow (P, a, u)$ is represented as $t = f(t_1, \ldots, t_n) \rightarrow (P, a, u)$.

If $|P| = 1$, then the claim holds trivially. If $|P| > 1$, the transition rule $f((P_x, x, u)) \rightarrow (P_y, x, u)$ is produced from the transition rules $f((P_x, x, u)) \rightarrow (P_y, x, u)$ where $j \in \{1, 2\}$ from (1) of Step 3 of $P_{in}$ and $P_{in}$, $\sigma$'s, and $\sigma$'s are represented as the following:

- $P_1 = \bigcup_{i=1}^{|P|} P_i$
- $P_i = \bigcup_{j=1}^{|P|} P_{ij}$
- $x_i = \bigcup_{j=1}^{|P|} x_i^j = a$
- $\sigma_i = \bigcup_{j=1}^{|P|} \sigma_i^j \cup \mathcal{C}$
- $c = c_1 \land c_2$

Here, we have $t \rightarrow (P_i, x_i, u)$ for $j \in \{1, 2\}$ and $1 \leq i \leq n$ similarly to the proof of Lemma 14. Since $t$ satisfies $c = c_1 \land c_2$, $t$ also satisfies both $c_1$ and $c_2$. Thus, we have $f(P_1, x_1, u) \rightarrow (P_2, x_1, u)$ for both $j=1$ and $j=2$.

Thus, similarly to the proof of Lemma 14, we have $t \rightarrow (P_1 \cup P_2, a, u)$ from Lemma 42.

**A.3 Supplement of Proofs of Lemma 44 and 47**

**A.3.1 Lemma 44**

(1) Here we show that if we have the transition $f(l_1, \ldots, l_n) \rightarrow \Delta_i$ and the rewrite rule $f(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$, we have $f(l_1, \ldots, l_n) \rightarrow \Delta_i$ and the rewrite rule $g(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$.

Since $f(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$, where $l_i \in Q_{nf}$ for all $i = 1, \ldots, n$. Therefore, we have $f(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$, where $l_i \in Q_{nf}$ for all $i = 1, \ldots, n$.

Thus, we have $f(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$, where $l_i \in Q_{nf}$ for all $i = 1, \ldots, n$.

Since $f(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$, where $l_i \in Q_{nf}$ for all $i = 1, \ldots, n$. Therefore, we have $f(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$, where $l_i \in Q_{nf}$ for all $i = 1, \ldots, n$.

Thus, we have $f(l_1, \ldots, l_n) \rightarrow r(l_1, \ldots, l_n)$, where $l_i \in Q_{nf}$ for all $i = 1, \ldots, n$.

(2) Here, we show that $r_j \sigma \rightarrow (P_j', x_j', u_j')$.

(a) For $j$ such that $r_j \in X$ and there exists $i$ such that $l_i = r_j$ and $x_i = a$, we have $r_j \sigma \rightarrow (P_j, i, u_i)$ from Proposition 31.

(b) For $j$ such that $r_j \in X$ and $x_i = 1$ for all such that $l_i = r_j$, then let $i_1, \ldots, i_k$ be all the numbers such that $l_{i_k} = r_j$ for $1 \leq h \leq k$. In this case, we have $l_i \sigma \rightarrow (P_1, x_i, u_i)$ for all $i$. Note that all $u_i$'s are equal from the derivability of $\Delta_{nf}$. Hence, we have $r_j \sigma \rightarrow (P_1 \cup \cdots \cup P, a, u_j') \rightarrow (P_j', x_j', u_j')$ where $u_j'$ is equal to all $u_i$'s from Lemma 43.

(c) For $j$ such that $r_j \not\in X$, we have $P_j = \{q_j^r\}$ and $r_j \sigma = r_j$ since $R$ is right-shallow so we can take the arbitrary state in $Q_{nf}$ as $u_j'$. Since $r_j \sigma \rightarrow (q_j^r, a, v'''),$ we have $r_j \sigma \rightarrow (q_j^r, i, v''')$ for some $v''' \in Q_{nf}$ from Proposition 28. Moreover, since we also have $r_j \sigma \rightarrow (q_j^r, i, v''')$ by Proposition 30, we obtain $r_j \sigma = r_j \sigma \rightarrow (q_j^r, i, v''')$ where $u_j'' = v'''$.

Thus, we have $g(r_1, \ldots, r_m) \sigma \rightarrow (g(r_1, \ldots, r_m) \sigma \rightarrow (P_r, x_r, u_r))$. More-
over, since \( g(r_1, \ldots, r_m) \sigma \) satisfies \( e' \), we have \( g(r_1, \ldots, r_m) \sigma \xrightarrow{\Delta_1} g((P'_1, x'_1, u'_1), \ldots, (P'_m, x'_m, u'_m)) \xrightarrow{\Delta_{k+1}} \langle \{ q \}, a, u' \rangle \)

**A.3.2 Lemma 47**

In the following, we show that we have substitution \( \sigma \) such that \( g(r_1, \ldots, r_m) \sigma \xrightarrow{\Delta_1} g(t_1, \ldots, t_m) \) and \( a' : g(r_1, \ldots, r_m) \sigma \xrightarrow{\Delta} g((P'_1, x'_1, v'_1), \ldots, (P'_m, x'_m, v'_m)) \xrightarrow{\Delta_1} \langle \{ q \}, a, v' \rangle \), where \( a' \subseteq a \).

1. For \( j \) such that \( r_j \in X \), \( j \not\subseteq \mu(g) \), and there exists \( i \) such that \( l_i = r_j \) and \( x_i = i \), we have \( t_j \xrightarrow{\Delta} \langle P'_1, x'_1, u'_1 \rangle = \langle P_1, i, u_1 \rangle \). Hence, we have \( t_j \xrightarrow{\Delta_0} \langle P'_1, x'_1, v'_1 \rangle \) from Proposition 29, and let \( r_j \sigma = t_j \).

2. For \( j \) such that \( r_j \in X \), \( j \in \mu(g) \), and there exists \( i \) such that \( l_i = r_j \) and \( x_i = i \), we have \( t_j \xrightarrow{\Delta_0} \langle P'_1, x'_1, u'_1 \rangle = \langle P_1, a, u \rangle \) where \( u \) is an arbitrary state in \( Q_{\text{NF}} \). Since \( P_1 \) is of the form \( \{ q_1 \} \) from Proposition 29, there exists some \( s_j \) such that \( s_j \xrightarrow{\Delta_0} t_j \) and \( s_j \xrightarrow{\Delta_{R_S}} \langle P'_1, x'_1, v'_1 \rangle \) for some \( v'_1 \) from the induction hypothesis. Let \( s_j \) be \( r_j \sigma \).

3. For \( j \) such that \( r_j \in X \) and there exists no \( i \) such that \( l_i = r_j \) and \( x_i = i \), we take \( r_j \sigma = t_j \).

4. For \( j \not\subseteq \mu(g) \) such that \( r_j \not\in X \), we have \( P'_j = \{ q' \} \) and \( x'_j = i \), and \( u'_j \) is an arbitrary state in \( Q_{\text{NF}} \). Since \( t_j \xrightarrow{\Delta} \langle \{ q' \}, i, u'_j \rangle \) by Proposition 29, we have \( t_j = r_j \) from Proposition 28 and the construction of \( \Delta_{R_S} \).

5. For \( j \in \mu(g) \) such that \( r_j \not\in X \), we have \( P'_j = \{ q' \} \) and \( x'_j = i \), and \( u'_j \) is an arbitrary state in \( Q_{\text{NF}} \). Here, we have \( s_j \xrightarrow{\Delta_0} t_j \) and \( s_j \xrightarrow{\Delta} \langle P'_j, x'_j, v'_j \rangle = \langle \{ q' \}, a, v'_j \rangle \) for some \( v'_j \in Q_{\text{NF}} \) from the induction hypothesis. Since \( s_j \xrightarrow{\Delta_0} \langle \{ q' \}, i, v'_j \rangle \) by Proposition 30, we have \( s_j = r_j \) from Proposition 28 and the construction of \( \Delta \).

\( \Box \)

(Received February 14, 2011)

(Accepted May 20, 2011)
Keiichirou Kusakari received his B.E. from Tokyo Institute of Technology in 1994, received M.E. and the Ph.D. degrees from Japan Advanced Institute of Science and Technology in 1996 and 2000. From 2000, he was a research associate at Tohoku University. He transferred to Nagoya University’s Graduate School of Information Science in 2003 as an assistant professor and became an associate professor in 2006. His research interests include term rewriting systems, program theory, and automated theorem proving. He is a member of IEICE, IPSJ and JSSST.

Toshiki Sakabe was born in 1949. He received his B.E., M.E. and D.E. degrees from Nagoya University in 1972, 1974 and 1978, respectively. He was a research associate at Nagoya University during 1977–1985, and an associate professor at Mie University and Nagoya University during 1985–1987 and 1987–1993, respectively. He has been a professor of Nagoya University since 1993. His research interests are in the field of theoretical foundations of software including algebraic specifications, rewriting computation, program verification, model checking and so on. He is a member of IPSJ, IEICE, JSAI and JSSST.