Context-sensitive Innermost Reachability is Decidable for Linear Right-shallow Term Rewriting Systems

Yoshiharu Kojima,† Masahiko Sakai,† Naoki Nishida,† Keiichirou Kusakari† and Toshiki Sakabe†

The reachability problem for given an initial term, a goal term, and a term rewriting system (TRS) is to decide whether the initial one is reachable to the goal one by the TRS or not. A term is shallow if each variable in the term occurs at depth 0 or 1. Innermost reduction is a strategy that rewrites innermost redexes, and context-sensitive reduction is a strategy in which rewritable positions are indicated by specifying arguments of function symbols. In this paper, we show that the reachability problem under context-sensitive innermost reduction is decidable for linear right-shallow TRSs. Our approach is based on the tree automata technique that is commonly used for analysis of reachability and its related properties. We show a procedure to construct tree automata accepting the sets of terms reachable from a given term by context-sensitive innermost reduction of a given linear right-shallow TRS.

1. Introduction

The reachability problem for given two terms s, t, and a term rewriting system (TRS) R is to decide whether s is reachable to t by R or not. Decision procedures of the problem are applicable to security protocol verification and 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. Reachability properties for several subclasses of TRSs have been proved to be decidable\(^5,4,6,11\)–\(^16\) as shown in Fig. 1.

Innermost reduction, a strategy that rewrites innermost redexes, is known as call-by-value computation widely used in most programming languages. Context-sensitive reduction\(^9\) is a strategy in which rewritable positions are indicated by specifying arguments of function symbols, and is used in evaluating if \(\cdots\) then \(\cdots\) else \(\cdots\) or case structures. Therefore, the languages that adopt call-by-value computation and if \(\cdots\) then \(\cdots\) else \(\cdots\) structure (e.g., C language) have computation models defined by context-sensitive innermost reduction. For innermost reduction and context-sensitive reduction, some decidable classes of reachability are known\(^5,8\), but it is not known for context-sensitive innermost reduction.

The reachability relation by context-sensitive innermost reduction is not equal to the intersection of reachability relations for innermost reductions and context-sensitive reduction, as shown in the following Example 1. Thus the decidability of reachability by context-sensitive innermost reductions cannot be obtained by the direct combination of the two existing results.
Example 1 Consider the following context-sensitive TRS:
\[
\begin{aligned}
\text{if}(\text{true}, x, y) & \rightarrow x \\
\text{if}(\text{false}, x, y) & \rightarrow y \\
a & \rightarrow b \\
f(x) & \rightarrow g(x)
\end{aligned}
\]

where reductions below \( f \) are inhibited. Then \( g(a) \) is reachable from \( \text{if}(\text{true}, f(a), a) \) by context-sensitive innermost reduction, but not reachable by innermost reduction.

This paper shows a procedure to construct a tree automaton (TA) accepting the set of terms reachable from a given term if a given TRS is linear and right-shallow, which means that the reachability and the joinability are both decidable. The procedure basically consists of both constructions\(^\text{(a)}\) for innermost reduction and context-sensitive reduction. However, the natural merger of the two procedures is not enough as shown later (Example 10).

2. Preliminary

We use usual notations of term rewriting system\(^1\) and tree automata\(^2\). 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 \( |\varepsilon\rangle \) of a position \( p \) is defined as \(|p|\). The height \( |t| \) of a term \( t \) is defined as \( \max\{|p| \mid p \in \text{Pos}(t)\} \). A term \( t \) is shallow if depths of variable occurrences in \( t \) are all 0 or 1. The subterm of \( t \) at position \( p \) is denoted by \( t[p] \), and \( t[t[p]'p] \) 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 \( s \preceq t \) that a term \( s \) is a subterm of \( t \).

A substitution \( \sigma \) is a mapping from \( X \) to \( T(F, X) \) whose domain \( \text{Dom}(\sigma) = \{x \in X \mid \sigma(x) \neq x\} \) 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 \( \xrightarrow{R} \) induced by a TRS \( R \) is as follows: \( s \overset{R}{\rightarrow} t \) if and only if \( s = s[l\sigma]_p \) and \( t = s[r\sigma]_p \) for some rule \( l \rightarrow r \in R \), with substitution \( \sigma \) and position \( p \in \text{Pos}(s) \). We call \( l\sigma \) a redex. We sometimes write \( \xrightarrow{R} \) by presenting the position \( p \) explicitly.

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

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 \( s \rightarrow t \). We use \( \circ \) to denote the composition of two relations. We write \( \rightarrow^* \) for the reflexive and transitive closure of \( \rightarrow \). We also write \( \rightarrow^\circ \) for the relation \( \rightarrow \circ \rightarrow \circ \cdots \rightarrow \circ \) composed of \( n \rightarrow \)'s. The set of reachable terms from a term in \( T \) is defined by \( \rightarrow[T] = \{t \mid s \rightarrow^* t\} \). The reachability problem (resp. joinability problem) with respect to \( \rightarrow \) is a problem that decides whether \( s \rightarrow^* s' \) (resp. \( s \rightarrow^\circ s' \)) or not, for given terms \( s \) and \( s' \).

A tree automaton (TA) is a quadruple \( A = (F, Q, Q^f, \Delta) \) where \( Q \) 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 \( f(q_1, \ldots, q_n) \rightarrow q \) or \( q_1 \rightarrow q \) where \( f \in F \) with \( \text{ar}(f) = n \), and \( q_1, \ldots, q_n, q \in Q \). We can regard \( \Delta \) as a (ground) TRS over \( F \cup Q \). The rewrite relation induced by \( \Delta \) is called a transition relation denoted by \( \rightarrow^\Delta \). We denote \( |\alpha| \) as the length of a transition sequence \( \alpha \) (if \( \alpha = s \rightarrow^\Delta t \), then \( |\alpha| = n \)). We say that a term \( s \in T(F) \) is accepted by \( A \) if \( s \rightarrow^\Delta q \in Q^f \). The set of all terms accepted by \( A \) is denoted by \( \mathcal{L}(A) \). We say \( A \) recognizes \( \mathcal{L}(A) \). A set of terms \( T \) is regular if there exists a TA that recognizes \( T \). We use a notation \( \mathcal{L}(A, q) \) or \( \mathcal{L}(\Delta, q) \) to represent the set \( \{s \mid s \rightarrow^\Delta q\} \). A TA \( A \) is deterministic if \( s \rightarrow^\Delta q \)
and \( s \xrightarrow{R} q' \) implies \( q = q' \) for any \( s \in T(F) \). A TA \( \mathcal{A} \) is complete if there exists \( q \in Q \) such that \( s \xrightarrow{R} q \) for any \( s \in T(F) \).

The following properties on TA are known\(^2\).

**Theorem 2**  
(1) For a given regular set \( T \in T(F) \), there exists a deterministic complete TA \( \mathcal{A} = (F, Q, Q', \Delta) \) that recognizes \( T \) and has no needless states:

\[
\forall q \in Q, L(\Delta, q) \neq \emptyset.
\]

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

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

### 3. Context-sensitive Innermost Reduction

We follow usual definitions of context-sensitivity\(^9\). A context-sensitive rewrite relation is a subrelation of the ordinary rewrite relation in which rewritable positions are indicated by specifying arguments of function symbols. A mapping \( \mu : F \rightarrow \mathcal{P}(\mathbb{N}) \) is said to be a replacement map (or \( F \)-map) if \( \mu(f) \subseteq \{1, \ldots, ar(f)\} \) for all \( f \in F \). A context-sensitive term rewriting system (CS-TRS) is a pair \( \mathcal{R} = (\mathcal{R}, \mu) \) of a TRS and a replacement map. The set of \( \mu \)-replacing positions \( \text{Pos}^\mu(t) \subseteq \text{Pos}(t) \) is recursively defined: \( \text{Pos}^\mu(t) = \{\varepsilon\} \) if \( t \) is a constant or a variable, otherwise \( \text{Pos}^\mu(f(t_1, \ldots, t_n)) = \{\varepsilon\} \cup \{ip \mid i \in \mu(f), p \in \text{Pos}^\mu(t_i)\} \). The rewrite relation induced by a CS-TRS \( \mathcal{R} \) is defined: \( s \xrightarrow{\mathcal{R}} t \) if and only if \( s \xrightarrow{\mathcal{R}} t \) for some \( p \in \text{Pos}^\mu(t) \). We call a normal form for CS-TRS \( \mathcal{R} \) context-sensitive normal form and we denote the set of context-sensitive normal forms by \( \text{CS-NF}_R \).

A rewrite step \( s \xrightarrow{\mathcal{R}} t \) is innermost if all proper subterms of \( s \) are normal forms. If every proper subterm is context-sensitive normal forms or not in \( \text{pos}^\mu(s) \), we say that the step is context-sensitive innermost, and we denote context-sensitive innermost reduction by \( \xrightarrow{\text{R}}_\text{in} \). A context-sensitive innermost rewrite relation is a subrelation of a context-sensitive rewrite relation, but it is not a subrelation of an innermost rewrite relation as shown in Example 1.

### 4. Tree Automaton Accepting Context-sensitive Normal Forms

In this section, we give a procedure to construct a deterministic complete TA \( \mathcal{A}_{\text{NF}} \) that recognizes the set of context-sensitive normal forms over \( F \) for left-linear CS-TRS \( \mathcal{R} \). This procedure is similar to ones for TRSs\(^3\).

The steps of the procedure to construct \( \mathcal{A}_{\text{NF}} \) is as follows:

1. Construct TA \( \mathcal{A}_l \) that recognizes the set of terms having a redex \( l\sigma \) at \( \mu \)-replacing position for each \( l \rightarrow r \in \mathcal{R} \).
2. Construct the union of all \( \mathcal{A}_l \)'s and take the complementation.
3. Convert the TA constructed in the step (2) into a deterministic and complete TA having no needless states. We output the TA as \( \mathcal{A}_{\text{NF}} \).

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

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

- \( Q_l = \{u^\circ \} \cup \{u^\downarrow \mid t \leq l\} \)
- \( Q_l' = \{u^\circ \} \)
- \( \Delta_l \) consists of following transition rules:
  - (i) \( f(u_1, \ldots, u_n) \rightarrow u \) for each \( f \in F \),
  - (ii) \( f(u_1^\downarrow, \ldots, u_n^\downarrow) \rightarrow u_f(t_1, \ldots, t_n)^\downarrow \) for each \( f \in F \) and state \( u_f(t_1, \ldots, t_n)^\downarrow \),
  - (iii) \( u \rightarrow u^\circ \),
  - (iv) \( f(u_1, \ldots, u_n) \rightarrow u^\circ \) for each \( f \in F \) if exactly one \( u_j \) such that \( j \in \mu(f) \) is \( u^\circ \) and the other \( u_j \)'s are \( u \).

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

We obtain the following lemmas for \( \mathcal{A}_l \).

**Lemma 3** If \( l \) is linear then \( L(\mathcal{A}_l, u^\downarrow) \) is equal to the set of all ground instances of \( l \), (that is, \( L(\mathcal{A}_l, u^\downarrow) = \{l\sigma \mid \sigma \in T(F)\} \))

**Proof:**

1. By induction on height \( |t| \) of \( t \), we show the claim that \( t\sigma \xrightarrow{\Delta_l} u^\downarrow \) for every substitution \( \sigma \) and subterm \( t \) of \( l \) such that \( t\sigma \) is ground.

2. In the case that \( t \) is a variable, we have \( t^\downarrow = \downarrow \). Since transition rules...
(i) guarantee that $s \xrightarrow{\Delta_i} u_\perp$ for any ground term $s$, the claim follows.

(2) Otherwise, we can write $t = f(t_1, \ldots, t_n)$ for $n = ar(f) \geq 0$. Since $t, \sigma \xrightarrow{\Delta_i} u_\perp$ by the induction hypothesis, we have $t \sigma = f(t_1, \ldots, t_n) \sigma \xrightarrow{\Delta_i} f(u_{t_1}, \ldots, u_{t_n}) \xrightarrow{\Delta_i} u_\perp$ from (ii) of the construction of $\Delta_i$.

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

**Lemma 4** If $l$ is linear then $\mathcal{L}(A_l) = \{t[s] \mid t \in T(F), s$ is a ground instance of $l, p \in \text{Pos}^b(t)\}$.

**Proof:**

\[\bigcirc\] From Lemma 3 and (iii) of the construction of $\Delta_i$, we have $t[s] \xrightarrow{\Delta_i} t[u_{s}] \xrightarrow{\Delta_i} t[u_{s}]$. Since $p \in \text{Pos}^b(t)$, we have $t[u_{s}] \xrightarrow{\Delta_i} u^\circ$ from the definition of $\text{Pos}^b$ and (i) and (iv) of the construction of $\Delta_i$.

\[\bigcirc\] Let $t \xrightarrow{\Delta_i} u^\circ$, then we have $t \xrightarrow{\Delta_i} t[u_{s}] \xrightarrow{\Delta_i} t[u_{s}] \xrightarrow{\Delta_i} u^\circ$ for some $p \in \text{Pos}^b(t)$ from (iii) and (iv) of the construction of $\Delta_i$. Hence we have $t = t[s] \xrightarrow{\Delta_i} t[s]$ for some ground instance $s$ of $l$ from Lemma 3.

As shown by the Lemma 4, the TA $A_l$ recognizes the set of terms having a redex $l \sigma$ at a $\mu$-replacing position. Now we obtain the following lemma.

**Lemma 5** For a left-linear CS-TRS $R$, we can construct a deterministic complete TA $A_{\text{NF}}$ that recognizes CS-NF and has no needless states.

**Proof:** By the step (1) and (2) of the proceeding procedure, we obtain a TA $A'$ that recognizes the complementation of the following set:

$$\bigcup_{l \in R} \mathcal{L}(A_l).$$

From Lemma 4 and (2) of Theorem 2, $\mathcal{L}(A')$ is the set of context-sensitive normal forms. The TA $A_{\text{NF}}$ obtained by the step (3) of the procedure is a deterministic and complete TA that recognizes CS-NF and having no needless states from (1) of Theorem 2.

We show an example of $A_{\text{NF}}$.

**Example 6** Consider the following $R = (R, \mu)$:

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

Final states and transition rules of a deterministic complete TA $A_{\text{NF}}$ that is minimized are as follows:

$$Q_{\text{NF}} = \{u_+, u_g, u_T\}$$

$$\Delta_{\text{NF}} = \left\{ 
  a \rightarrow u_+, 
  b \rightarrow u_g, 
  c \rightarrow u_e, 
  e \rightarrow u_+,
  g(u_+) \rightarrow u_g, 
  g(u_T) \rightarrow u_g, 
  g(u_g) \rightarrow u_g, 
  g(u_e) \rightarrow u_e, 
  g(u_c) \rightarrow u_c, 
  b(u_g) \rightarrow u_g, 
  f(u_+) \rightarrow u_+, 
  f(u_T) \rightarrow u_T, 
  f(u_g) \rightarrow u_g, 
  f(u_e) \rightarrow u_e, 
  f(u_c) \rightarrow u_c, 
  f(u_+) \rightarrow u_+, 
  f(u_T) \rightarrow u_T.
\right\}$$

For the constructed TA $A_{\text{NF}}$, the following proposition holds from Lemma 5.

**Proposition 7** If $f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{\text{NF}}$ and $u \in Q_{\text{NF}}^f$, then $i \in \mu(f)$ implies $u_i \in Q_{\text{NF}}^f$. 

**Proof:** Let $f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{\text{NF}}$, $u \in Q_{\text{NF}}^f$, and assume $u_i \notin Q_{\text{NF}}^f$ for some $i \in \mu(f)$. Since $A_{\text{NF}}$ has no needless states by Lemma 5, there exists a terms $t_1, \ldots, t_n$ such that $t_j \xrightarrow{\Delta_{\text{NF}}} u_j$ for each $j (1 \leq j \leq n)$. Hence $f(t_1, \ldots, t_n) \xrightarrow{\Delta_{\text{NF}}} f(u_1, \ldots, u_n) \xrightarrow{\Delta_{\text{NF}}} u$. Here that $f(t_1, \ldots, t_n) \in \text{CS-NF}_R$ and $t_i \notin \text{CS-NF}_R$ from Lemma 5. 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...
5. Procedure to Construct the Set of Reachable Terms by Context-sensitive Innermost Reduction

In this section, we give a procedure to construct a tree automaton that recognizes the set of reachable terms by context-sensitive innermost reduction of a linear right-shallow TRS. Procedures to construct a tree automaton that recognizes the set of reachable terms from a given set of terms by innermost (resp. context-sensitive) reduction have already shown for linear right-shallow TRSs\(^8\). One may think that we can construct a tree automaton that recognizes the set of reachable terms by context-sensitive innermost reduction, by natural merger of these two procedures and by introducing context-sensitive normal form. However, it appears that this is not true and we need some more modification.

Before giving the procedure, we define \(\text{RS}(R)\) and \(\mathcal{A}_{\text{RS}}\), which are necessary to describe the procedure.

**Definition 8** \(\text{RS}(R)\) is the set of all non-variable direct subterms of the right-hand sides of rules in TRS \(R\): \(\text{RS}(R) = \{ t \in \text{RS}(R) | l \rightarrow f(r_1, \ldots, r_m) \in R \} \).

Note that \(\text{RS}(R)\) is a set of ground terms if \(R\) is right-shallow.

**Definition 9** \(\mathcal{A}_{\text{RS}} = (F, Q_{\text{RS}}, Q_{\text{RS}}^f, \Delta_{\text{RS}})\) is a TA that accepts \(\text{RS}(R)\) and satisfies with the following conditions:

- \(Q_{\text{RS}}^f = \{ q^f | t \in \text{RS}(R) \}\) and
- \(\mathcal{L}(\mathcal{A}_{\text{RS}}, q^f) = \{ t \}\).

Now we give the procedure \(\text{P}_\text{csin}\) that constructs a TA recognizing a set of reachable terms by context-sensitive innermost reduction. The procedure design is based on merging the procedures for innermost case and context-sensitive case. More precisely, each state of resulting TA consists of three components; the first component originates in the input automata, the second one indicates whether we can add the transition rules in order to transit to this state or not, The third one remembers whether the corresponding terms are context-sensitive normal forms or not.

**Procedure \(\text{P}_\text{csin}\):**

**Input** A TA \(\mathcal{A} = (F, Q, Q_f, \Delta)\) such that \(w_1 \rightarrow q \in \Delta\) and \(w_2 \rightarrow q \in \Delta\) imply \(w_1 = w_2\), and a left-linear right-shallow CS-TRS \(R\) over \(F\) of \(\mathcal{A}\).

**Output** A TA \(\mathcal{A}_* = (F, Q, Q_f^*, \Delta_*)\) such that \(\mathcal{L}(\mathcal{A}_*) = \mathcal{L}(\mathcal{A}) \cap [\mathcal{L}(R)]\) if \(R\) is right-linear.

**Step 1 (initialize)** (1) Prepare \(\mathcal{A}_{\text{RS}}\) and \(\mathcal{A}_{\text{NF}}\).

(2) Let

- \(k := 0\),
- \(Q_i = (Q \cup Q_{\text{RS}}) \times \{ a, i \} \times Q_{\text{NF}}\),
- \(Q_f^i = Q_f \times \{ a \} \times Q_{\text{NF}}\), and
- \(\Delta_0\) as follows:
  - \((a)\) \(\langle q', x, u \rangle \rightarrow \langle q, x, u \rangle \in \Delta_0\) for \(x \in \{ a, i \}\) where \(q' \rightarrow q \in \Delta\) and \(u \in Q_{\text{NF}}\),
  - \((b)\) \(f(\langle q_1, u_1 \rangle, \ldots, \langle q_n, u_n \rangle) \rightarrow \langle q, u \rangle \in \Delta_0\) where \(f(q_1, \ldots, q_n) \rightarrow q \in \Delta \cup \Delta_{\text{RS}}\) and \(f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{\text{NF}}\), and
  - \((c)\) \(f(\langle q_1, x_1, u_1 \rangle, \ldots, \langle q_n, x_n, u_n \rangle) \rightarrow \langle q, a, u \rangle \in \Delta_0\) where \(f(q_1, \ldots, q_n) \rightarrow q \in \Delta \cup \Delta_{\text{RS}}\), \(f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{\text{NF}}\), and if \(i \in \mu(f)\) then \(x_i = a\), otherwise \(x_i = i\).

**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:

(1) Produce the transition rule \(g(\langle q_i', x_i', u_i' \rangle, \ldots, \langle q_m', x_m', u_m' \rangle) \rightarrow \langle q, a, u' \rangle \in \Delta_{k+1}\) from \(f(l_{i1}, \ldots, l_{in}) \rightarrow g(r_{j1}, \ldots, r_{jm}) \in R\) and \(f(q_1, x_1, u_1), \ldots, (q_n, x_n, u_n) \rightarrow (q, a, u) \in \Delta_k\) where \(u_i \in Q_{\text{NF}}\) or \(x_i = i\) for all \(1 \leq i \leq n\) if there exists \(\theta : X \rightarrow (Q \cup Q_{\text{RS}}) \times \{ a, i \} \times Q_{\text{NF}}\) such that

- \(\mathcal{L}(\Delta_k, x, \theta) \neq \emptyset\) for all \(x \in \text{Var}(f(l_{11}, \ldots, l_{in}))\),
- \(l_{i1} \theta \xrightarrow{\Delta_k} \langle q, x_i, u_i \rangle\),

where

- let \(r_j \theta = \langle q_j'', x_j'', u_j'' \rangle\) for each \(r_j \in X\) then
  \(q_j'' = \begin{cases} q_j' & \text{if } r_j \in X \\ q_j & \text{if } r_j \notin X \end{cases}\)
  \(x_j'' = \begin{cases} x_j' & \text{if } j \notin \mu(g), r_j \in X \\ i & \text{if } j \notin \mu(g), r_j \notin X \\ a & \text{otherwise} \end{cases}\)
— $u'_j = \begin{cases} u'' \quad \text{if } r_j \in X, j \not\in \mu(g) \lor (j \in \mu(g) \land x''_j = a) \\ v \in Q_{NF} \quad \text{otherwise} \end{cases}$

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

• $g(u'_1, \ldots, u'_m) \xrightarrow{\Delta_{k_0}} u'$.

(2) Produce $\langle q', a, u' \rangle \rightarrow \langle q, a, u'' \rangle \in \Delta_{k+1}$ from $f(l_1, \ldots, l_n) \rightarrow x \in R$ and $f(q_0, a, c) \rightarrow q_a, g(q_a) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q_f) \rightarrow q_a, f(q_g) \rightarrow q_f, f(q_f) \rightarrow q_g$ of transition rules. At the initializing, we have $A_{RS}$ and $A_{NF}$ at (1). $A_{RS}$ is defined by a set of $q^*$ of final state and a set of $\{e \rightarrow q^*\}$ transition rules, and $A_{NF}$ is defined as Example 6. At (2) of initializing step, we have the following:

• $Q_* = \{q_0, q, x, u\}, \{q_g, x, u\}, \{q_g, x, u\}, \{q_f, x, u\}, \{q_f, x, u\}, \{q_f, x, u\}$, where $u \in Q_{NF}$ and $x \in \{a, i\}$.

• $Q^I = \{q_{fa}, a, u\}, \{q_{fc}, a, u\}$ where $u \in Q_{NF}$.

• $\Delta_0 = \begin{cases} a \rightarrow \langle q_a, x, u \rangle, c \rightarrow \langle q_c, x, u \rangle, e \rightarrow \langle q_e, x, u \rangle, \\ g(q_0, a, c) \rightarrow q_a, g(q_a) \rightarrow q_g, g(q_g) \rightarrow q_f, f(q_f) \rightarrow q_a, f(q_g) \rightarrow q_f, f(q_f) \rightarrow q_g \end{cases}$

where $x \in \{a, i\}$, $u_1 \in \{\overline{x}, u_c\}$, $u_2 \in \{u_\perp, u_g, \overline{u}_g\}$, $u_3 \in \{u_\perp, u_c, u_g, \overline{u}_g\}$, and $u \in \{\overline{x}, \overline{u}_g\}$.

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

• $\Delta_1 = \Delta_0 \cup \{b \rightarrow \langle q_a, \overline{x}, u \rangle, e \rightarrow \langle q_c, a, \overline{u}_c \rangle, e \rightarrow \langle q_g, a, \overline{u}_c \rangle, \langle q_f, a, u \rangle \rightarrow \langle q_f, a, u \rangle \}$ where $u \in Q_{NF}$.

The output TA $A_*$ recognizes terms in $\{f(g(a)), f(g(c)), a, b, f(e)\} = \overline{R_{im}}[f(g(a)), f(g(c))]$.

This procedure $P_{csin}$ eventually terminates at some $k$, because the procedure add transition rules in the repetition steps but it does not change states $Q_*$, which guarantees that possible transitions rules are finite. Apparently $\Delta_0 \subset \cdots \subset \Delta_k = \Delta_{k+1} = \cdots$

Differences from natural merger of existing procedures are the following:

(i) a restriction that the input TA has no rules such that $s \rightarrow q$ and $t \rightarrow q$ where $s \neq t$.

(ii) the condition $L(\Delta_k, x\theta) \neq \emptyset$ at Step 2 is added, and

(iii) in the third item of Step 2-(1), $u'_j$ is an arbitrary state $v \in Q_{NF}$ even if $j \in \mu(g) \land x''_j = i$, while $u''_j = u''_j$ in the naturally merged procedure.

Unless (ii), we produce a transition rule $\langle q_a, a, \overline{u}_c \rangle \rightarrow \langle q_{fc}, a, \overline{u}_c \rangle$ from $f(g(x)) \xrightarrow{\Delta_{k_0}} x \in R$, $f(q_g, a, \overline{u}_c) \rightarrow \langle q_{fc}, a, \overline{u}_c \rangle \in \Delta_0$, and $x \theta = \langle q_a, a, \overline{u}_c \rangle$. In this case, the output TA accepts $e$, which is unreachable from $f(g(a))$ nor $f(g(c))$.

If we do not have the modification (iii), then $u''_j$ is $u''_j$ of $r_j \theta$. Then a transition rule $b \rightarrow \langle q_a, a, \overline{u}_c \rangle$ is not produced and hence the output TA does not accept $b$ which is reachable from $f(g(a))$. However, the extension (iii) causes a problem that if the input TA $A$ in Example 10 has set $\{a \rightarrow q_ac, c \rightarrow q_ac, g(q_ac) \rightarrow q_g, f(q_a) \rightarrow q_g\}$ of transition rules, and the output TA accepts $c$ and $e$. To avoid this problem, we introduce the restriction (i). Even if we have the restriction (i), we can construct a TA accepting only one term and hence it suffices for the reachability problem and joinability problem.

We show several propositions and lemmas to show $L(A_*) = \overline{R_{im}}[L(A)]$ for the input TA $A$ and the output TA $A_*$ of the procedure $P_{csin}$. 
Proposition 11 (a) Let \( s \in \mathcal{T}(F) \), then \( s \xrightarrow{\Delta} q \in Q \) or \( s \xrightarrow{\Delta_{RS}} q \in Q_{RS} \) if and only if \( s \xrightarrow{\Delta_0} (q, i, u) \) for some \( u \in Q_{NF} \).

(b) \( f(q_1, x_1, u_1), \ldots, (q_n, x_n, u_n)) \rightarrow (q, x, u) \in \Delta_* \) implies \( f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{NF} \).

Proof: The claim is a direct consequence of the construction of \( \Delta_0 \) and the completeness of \( \Delta_{NF} \).

Proposition 12 If a rule \( f \) is introduced at (a) of Step 1 (2), and \( \Delta \) is (in the form of) \( q \rightarrow^* q' \), then \( \Delta_0 \) is also in \( \Delta_0 \). Moreover, \( x_i = i \) for all \( 1 \leq i \leq n \).

Proof: Such rules are introduced at Step 1 and hence the claim follows from construction of \( \Delta_0 \).

Proposition 13 Let \( t \in \mathcal{T}(F) \), \( q \in Q \cup Q_{RS} \), and \( u \in Q_{NF} \), then \( t \xrightarrow{\Delta_0} (q, i, u) \) implies \( t \xrightarrow{\Delta_0} (q, i, u) \).

Proof: The proposition follows from Proposition 12.

Proposition 14 Let \( t \in \mathcal{T}(F) \), \( q \in Q \cup Q_{RS} \), and \( u \in Q_{NF} \). Then, \( t \xrightarrow{\Delta_0} (q, a, u) \) if and only if \( t \xrightarrow{\Delta_0} (q, i, u) \).

Proof: We present a proof for “only if" part, since the proof for the other part is not only similar to this part but also simpler. We use induction on the length of the former sequence. In the case of \( t \xrightarrow{\Delta_0} (q', x, u') \xrightarrow{\Delta_0} (q, a, u) \), the rule \( (q', x, u') \xrightarrow{\Delta_0} (q, a, u) \) is introduced at (a) of Step 1 (2), and \( x = a \).

Thus \( (q', i, u') \rightarrow (q, i, u) \) is also in \( \Delta_0 \). Since \( t \xrightarrow{\Delta_0} (q', i, u') \) by the induction hypothesis, the claim follows.

Otherwise, \( t \xrightarrow{\Delta_0} (q, a, u) \) can be represented as \( t = f(t_1, \ldots, t_n) \xrightarrow{\Delta_0} f(q_1, x_1, u_1), \ldots, (q_n, x_n, u_n)) \xrightarrow{\Delta_0} (q, a, u) \). The rule \( f(q_1, x_1, u_1), \ldots, (q_n, x_n, u_n) \) is introduced at (c) of Step 1 (2), where \( x_i \in \{i, a\} \), \( f(q_1, \ldots, q_n) \rightarrow q \in \Delta \cup \Delta_{RS} \), and \( f(u_1, \ldots, u_n) \rightarrow u \in \Delta_{NF} \). Thus we also have \( f(q_1, x_1, u_1), \ldots, (q_n, x_n, u_n)) \rightarrow (q, i, u) \in \Delta_0 \) (by (b) of Step 1 (2)). Since we have \( t_i \xrightarrow{\Delta_0} (q, i, u_i) \) for each \( t_i \) by the induction hypothesis, the claim follows.

Proposition 15 Let \( t \in \mathcal{T}(F) \), \( q \in Q \cup Q_{RS} \), and \( u \in Q_{NF} \). Then, \( t \xrightarrow{\Delta_0} (q, i, u) \) implies \( t \xrightarrow{\Delta_0} (q, a, u) \).

Proof: Let \( t \xrightarrow{\Delta_0} (q, i, u) \), then \( t \xrightarrow{\Delta_0} (q, i, u) \) by Proposition 13. The proposition follows from Proposition 14 and \( \Delta_0 \subseteq \Delta_k \).

Lemma 16 Let \( s, t \in \mathcal{T}(F) \), \( s \xrightarrow{\Delta_0} (q, x, u) \), and \( t \xrightarrow{\Delta_0} (q', x', u') \). Then, \( q = q' \) implies \( s = t \).

Proof: First we have \( s \xrightarrow{\Delta_0} (q, x, u) \) and \( t \xrightarrow{\Delta_0} (q', x', u') \) from Proposition 14. If \( q = q' \), then we have either \( s \xrightarrow{\Delta_0} q \wedge t \xrightarrow{\Delta_0} q \) or \( s \xrightarrow{\Delta_{RS}} q' = q \wedge t \xrightarrow{\Delta_{RS}} q' = q' \). From Proposition 11 (a) and the construction of \( \Delta_{RS} \). In the case of \( s \xrightarrow{\Delta_{RS}} q' \) and \( t \xrightarrow{\Delta_{RS}} q', q' = q \) implies \( s = t \) from the construction of \( \Delta_{RS} \). We show the case of \( q \geq 0 \).

(1) In the case of \( |\alpha| = 1 \), we have \( s \xrightarrow{\alpha} q \). In this case, \( s \) is the only term to transit to \( q \) from the restriction for \( \alpha \). Hence \( t \xrightarrow{\alpha} q \) implies \( s = t \).

(2) (a) In the case of \( |\alpha| > 1 \), we consider that the last transition rule applied in \( \alpha \) is (in the form of) \( f(q_1, \ldots, q_n) \xrightarrow{\alpha} q \). Let \( s = f(s_1, \ldots, s_n) \), then we have the transition sequence \( s = f(s_1, \ldots, s_n) \xrightarrow{\alpha} f(q_1, \ldots, q_n) \xrightarrow{\alpha} q \). From the restriction for \( \alpha \), \( f(q_1, \ldots, q_n) \) is the only term to transit to \( q \), and hence we have \( t = f(t_1, \ldots, t_n) \xrightarrow{\alpha} f(q_1, \ldots, q_n) \xrightarrow{\alpha} q \) by letting \( t = f(t_1, \ldots, t_n) \).

Since we have \( s_i = t_i \) for each \( i \) from the induction hypothesis, we also have \( s = t \).

(b) In the case where \( |\alpha| > 0 \) and the last transition rule applied in \( \alpha \) is (in the form of) \( q' \xrightarrow{\alpha} q \), we can show the lemma similarly to the previous case (a).
Lemma 19

Proof: If \( u \in \text{IPSJ Transactions on Programming Vol. 2 No. 3 20–32 (July 2009)} \) c, follows from Lemma 17. Hence this lemma holds by the induction hypothesis.

(2) In the case where the last transition rule applied in \( x \) is (in the form of) \( \langle q''', a, u'' \rangle \rightarrow \langle q', a, u' \rangle \in \Delta_k \), 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.

\( \square \)

Lemma 18 If \( j \not\in \mu(\alpha) \) and \( g(\ldots, \langle q_j', x_j', u_j' \rangle, \ldots) \rightarrow \langle q, x', u'' \rangle \in \Delta_k \), then \( u_j' \in Q_{\text{NF}} \) if \( x_j' = 1 \).

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 \( g(\ldots, \langle q_j', x_j', u_j' \rangle, \ldots) \rightarrow \langle q, x', u'' \rangle \in \Delta_k \), without loss of generality. Since this rule is introduced by the first inference rule of Step 2, we have \( x' = a \), and there exist \( f(l_1, \ldots, l_n) \rightarrow g(r_1, \ldots, r_m) \in R \) and \( f((q_1, x_1, u_1), \ldots, (q_n, x_n, u_n)) \rightarrow \langle q, a, u \rangle \in \Delta_k \), where \( u \in Q_{\text{NF}} \) or \( x_i = 1 \) for all \( 1 \leq i \leq n \). Then \( \theta : X \rightarrow Q_\ast \) such that \( \Delta_1^{\sim} \rightarrow \langle q, x, u \rangle \). From \( j \not\in \mu(\alpha) \), we have either \( \langle q_j', x_j', u_j' \rangle = r_j \theta \) if \( r_j \in X \), or \( x_j' = 1 \) if \( r_j \not\in X \). In the case \( r_j \in X \), we have \( l_i \rightarrow r_j \) for some \( i \) and \( p \), we have \( l_i = l_i[[\langle q', x', u' \rangle]_p] \rightarrow \langle q, x, u \rangle \) if \( x_j' = a \), then we have \( x_i = a \) by Proposition 12 and hence \( u \in Q_{\text{NF}} \). Therefore, \( u_j' \in Q_{\text{NF}} \) follows from Lemma 17.

\( \square \)

Lemma 19 Let \( \alpha : s[(q, x, u)]_p \rightarrow \langle q', x', u' \rangle \). If \( k = 0 \) or \( u \in Q_{\text{NF}} \) \( \land x = x' = a \land p \in \text{Pos}^\mu(s) \), then there exists \( v' \in Q_{\text{NF}} \) such that \( s[(q, x, v)]_p \rightarrow \langle q', x', v' \rangle \) for \( v \in Q_{\text{NF}} \).

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

Following we suppose \( |\alpha| > 0 \).

(1) Consider the case where the last transition rule applied in \( x \) is (in the form of) \( \langle q_j', x_j', u_j' \rangle, \ldots, \langle q_m', x_m', u_m' \rangle \rightarrow \langle q', x', u' \rangle \in \Delta_k \). Then \( \alpha \) can be represented as \( s[(q, x, u)]_p \rightarrow \langle q_j', x_j', u_j' \rangle, \ldots, \langle q_m', x_m', u_m' \rangle \rightarrow \langle q', x, u' \rangle \). In this case, the position \( p \) can be represented as \( j' \) for \( 1 \leq j \leq m \). From \( j \not\in \mu(g) \), Proposition 11 (b), and Proposition 7, we have \( u_j' \in Q_{\text{NF}} \) and hence we also have \( u \in Q_{\text{NF}} \) by the induction hypothesis.

(2) In the case where the last transition rule applied in \( x \) is (in the form of) \( \langle q', a, u' \rangle \rightarrow \langle q', a, u' \rangle \in \Delta_k \), 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.

\( \square \)
We show the lemma by induction on $\langle q', a, u' \rangle \in \Delta_k \setminus \Delta_{k-1}$ for some $u' \in Q_{\text{NF}}$ from the construction.

(2) In the case where the last transition rule applied in $\alpha$ is (in the form of)
$\langle q', x', u' \rangle \rightarrow (q, x, u) \in \Delta_k$, we have $u' = u$ from the construction of $\Delta_0$ or the second inference rule of Step 2. Hence this lemma holds from the induction hypothesis.

\[ \square \]

**Lemma 20**

If $\alpha : l[t'_p]_p \xrightarrow{\Delta_k} (q, a, u)$ and $p \in \text{Pos}^c(t)$, then there exists $\langle q', a, u' \rangle$ such that $\alpha = \langle q', a, u' \rangle$ and $l[t'(q', a, u')]_p \xrightarrow{\Delta_k} (q, a, u)$. 

**Proof:** We show the lemma by induction on $|\alpha|$.

(1) If $|\alpha| = 0$, then we have $p = \epsilon$ and $t = t'$. Hence $\alpha \xrightarrow{\Delta_k} (q, a, u)$.

(2) In the case of $|\alpha| > 0$, we have $p = ip'$ for some $i \in \mathbb{N}$. Then $\alpha$ can be represented as $l[t'_p] = f(\ldots, ti(t'_p'), \ldots) \xrightarrow{|\alpha|-1} f(\ldots, (q_i, x_i, u_i), \ldots) \xrightarrow{\Delta_k} (q, a, u)$. Since $ip \neq p \in \text{Pos}^c(t)$, we have $i \in \mu(f)$. Hence $x_i = a$ is follows from the construction of $\Delta_k$.

By the induction hypothesis, there exists $\langle q', a, u' \rangle$ such that $\alpha = \langle q', a, u' \rangle$ and $l[t'(q', a, u')]_p \xrightarrow{\Delta_k} (q, a, u)$. Thus we have $\alpha = l[t'(q', a, u')]_p = f(\ldots, ti(q', a, u')]_p, \ldots) \xrightarrow{\Delta_k} (\ldots, (q_i, a, u_i), \ldots) \xrightarrow{\Delta_k} (q, a, u)$.

\[ \square \]

**Lemma 21**

Let $R$ be left-linear and right-shallow. Then $s \xrightarrow{\Delta_k} \langle q, a, u \rangle$ and $s \xrightarrow{\Delta_k} t$ imply $t \xrightarrow{\Delta_k} \langle q, a, u' \rangle$ for some $u' \in Q_{\text{NF}}$.

**Proof:** We present the proof in the case where $s \xrightarrow{\Delta_k} \langle q, a, u \rangle$ and $s \xrightarrow{\Delta_k} t$ imply $t \xrightarrow{\Delta_k} \langle q, a, u' \rangle$ for some $u' \in Q_{\text{NF}}$.

Let $\sigma = s[R]_p \in \text{Pos}^c(s)$. Then we have the rewrite rule $\beta \rightarrow \gamma$. Let $n = 1$ for all rewrite rules and $m = (a \rightarrow a).$ Then for the case where rewrite rule $\beta \rightarrow \gamma$ in $R$. We have the transition sequence $s \xrightarrow{\Delta_k} \langle q'', a, u'' \rangle \xrightarrow{\Delta_k} \langle q, a, u \rangle$ by Lemma 20.

(1) Consider the case where the rewrite rule in the form $f(l_1, \ldots, l_n) \rightarrow g(l_1, \ldots, l_n)$. The diagram of this case is shown in Fig. 2. Then we have $s = s[f(l_1, \ldots, l_n)]_p \xrightarrow{\Delta_k} s[g(l_1, \ldots, l_n)]_p = t$, and thus $l_1 \sigma$ is a context-sensitive normal form for each $i \in \mu(f)$.

**Fig. 2** The diagram of the proof of Lemma 21.

$\langle q, a, u \rangle$ is represented as $s = s[f(l_1, \ldots, l_n)]_p \xrightarrow{\Delta_k} s[f(l_1, \ldots, l_n)]_p \xrightarrow{\Delta_k} s[f(l_1, \ldots, l_n)]_p \xrightarrow{\Delta_k} \langle q', a, u' \rangle$. Since $u'' \notin Q_{NF}$ since $s[R]_p$ is not a context-sensitive normal form.

For $i \in \mu(f)$, $l_i \sigma$ is a context-sensitive normal form and hence we have $u_i \in Q_{\text{NF}}$. For $i \not\in \mu(f)$, we have $x_i = a$ or $u_i \in Q_{\text{NF}}$ from Lemma 18. Let $r_j \theta = (q''_j, x''_j, u''_j)$ for each $r_j \in X$. Then, we have $r_j \sigma \xrightarrow{\Delta_k} \theta \sigma \tau$ if $r_j \in X$.

Since $f(l_1, \ldots, l_n) \rightarrow g(l_1, \ldots, l_n)$, we have $r_j \sigma \xrightarrow{\Delta_k} \theta \sigma \tau \xrightarrow{\Delta_k} \langle q', a, u' \rangle$. Since $\Delta_k \ni \delta = (q''_j, x''_j, u''_j)$, there exist transition rules $g(q''_j, x''_j) \rightarrow (q'', a, u'')$.

Now we show that $r_j \sigma \xrightarrow{\Delta_k} \langle q''_j, x''_j, u''_j \rangle$.

(a) For $j$ such that $r_j \in X$, since we have $q''_j = q''_j$ and since we can take $u''_j = u''_j$, we have $r_j \sigma \xrightarrow{\Delta_k} \theta \sigma \tau = (q''_j, x''_j, u''_j)$. If $x''_j \neq x''_j$ then $x''_j = a$ and $x''_j = a$. Thus we obtain $r_j \sigma \xrightarrow{\Delta_k} \theta \sigma \tau$ by Proposition 15.

(b) For $j$ such that $r_j \notin X$, we have $q''_j = q''_j$ and $r_j \sigma \xrightarrow{\Delta_k} r_j \sigma = (q''_j, x''_j, u''_j)$ by the construction of $\Delta_k$. Since $Q_{\text{NF}}$ is right-linear,
shallow. And then we can take arbitrary state in $Q_{NF}$ as $u'$. Since $r_j \xrightarrow{\Delta_{RS}} q^o, a$ we have $r_j \xrightarrow{\Delta_{RS}} (q^o, i, i')$ for some $i' \in Q_{NF}$ from Proposition 11 (a). Since we also have $r_j \xrightarrow{\Delta_{RS}} (q^o, i, i')$ by Proposition 15, we obtain $r_j \sigma = r_j \xrightarrow{\Delta_{RS}} (q^o, i, i') = (q_j, x_j', u_j')$ where $u_j' = i'$. It follows from Lemma 19 and $s[(q'', a, u'')] \xrightarrow{\Delta_{RS}} (q, a, u)$ with $u'' \notin Q_{NF}$ that $s[(q'', a, u'')] \xrightarrow{\Delta_{RS}} (q, a, u)$ for some $u'' \in Q_{NF}$. Therefore we have $t \xrightarrow{\Delta_{RS}} s[(q_1, x_1, u_1), \ldots, (q_n, x_n, u_n)] \xrightarrow{\Delta_{RS}} s[(q', a, u')]$ by Proposition 15. Hence the lemma holds by letting $s = t$. □

Lemma 22 Let $R$ be left-linear and right-shallow. Then $\mathcal{L}(A_*) \supseteq \xrightarrow{\text{R}_{in}}[\mathcal{L}(A)]$. Proof: Let $s \xrightarrow{\text{R}_{in}} t$ and $s \xrightarrow{\Delta_{RS}} q \in Q_f$. Since $s \xrightarrow{\Delta_{RS}} (q, i, u) \in Q_f$ from Proposition 11 (a), we have $s \xrightarrow{\Delta_{RS}} (q, a, u) \in Q_f$ by Proposition 15. Therefore $t \xrightarrow{\Delta_{RS}} (q', a, u')$ for some $u' \in Q_{NF}$ by Lemma 21. □

Before proving soundness of the procedure $P_{\text{sim}}$, we define a measurement of transitions.

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

$$||s_0 \xrightarrow{\Delta_{RS}} s_1 \xrightarrow{\Delta_{RS}} \cdots \xrightarrow{\Delta_{RS}} s_n \xrightarrow{\Delta_{RS}} t|| = \left( ||s_0 \xrightarrow{\Delta_{RS}} \cdots \xrightarrow{\Delta_{RS}} s_n || \right)$$

where $||s \xrightarrow{\Delta_{RS}} t|| = 0$ for $i < n$.

Now we can define the following order $\supseteq$ on transition sequences by $\Delta_{RS}$, which is necessary in proofs:

$$\alpha \supseteq \beta \iff ||\alpha|| >_{\text{mul}} ||\beta||$$

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

Lemma 23 Let $\Delta_{RS}$ be generated from a linear right-shallow CS-TRS $R$. Then $\alpha : t \xrightarrow{\Delta_{RS}} (q, x, u')$ implies $s \xrightarrow{\text{R}_{in}} t$ and $\beta : \xrightarrow{\Delta_{RS}} (q, x, u)$ for some term $s$ and $u \in Q_{NF}$.

Proof: Let $t = g(t_1, \ldots, t_m)$. We show this lemma by induction on $||\alpha||$ with respect to $\supseteq$.

(1) Consider the case where the last transition rule applied in $\alpha$ is in the form of $g((q'_1, x'_1, u'_1), \ldots, (q'_n, x'_n, u'_n)) \xrightarrow{\Delta_{RS}} (q, x, u') \in \Delta_k$. Let $\alpha_j : t_j \xrightarrow{\Delta_{RS}} (q'_j, x'_j, u'_j)$.
\[ q_j' = \begin{cases} 
q_j'' \ldots & \text{if } r_j \in X 
q_j'' \ldots & \text{if } r_j \not\in X 
x_j'' \ldots & \text{if } j \not\in \mu(g), r_j \in X 
1 \ldots & \text{if } j \not\in \mu(g), r_j \not\in X 
a \ldots & \text{otherwise} 
\end{cases} \]

\[ x_j' = \begin{cases} 
x_j'' \ldots & \text{if } j \not\in \mu(g), r_j \in X 
x_j'' \ldots & \text{if } j \not\in \mu(g), r_j \not\in X 
i \ldots & \text{if } j \not\in \mu(g), r_j \not\in X 
a \ldots & \text{otherwise} 
\end{cases} \]

\[ u_j' = \begin{cases} 
u_j'' \ldots & \text{if } r_j \in X, j \not\in \mu(g) \lor (j \in \mu(g) \land x_j'' = a) 
v \in Q_{\text{NF}} \ldots & \text{otherwise} 
\end{cases} \]

where \( r_j \theta = \langle q_j'', x_j', u_j'' \rangle \) for \( r_j \in X \).

The following \((i)-(iv)\) show that for each \( j \), there exists \( s_j \) such that \( s_j \xrightarrow{R^*_k} t_j \), \( s_j \xrightarrow{R^*_k} r_j \theta \) and \( \alpha'': s_j \xrightarrow{R^*_k} \langle q_j', x_j', v_j' \rangle \) with \( \alpha \supseteq \alpha'' \) for some \( v_j' \in Q_{\text{NF}} \).

\((i)\) For each \( j \) such that \( r_j \in X \) and \( j \not\in \mu(g) \lor (j \in \mu(g) \land x_j'' = a) \), we have \( t_j \xrightarrow{R^*_k} \langle q_j', x_j', v_j' \rangle = \langle q_j'', x_j', u_j'' \rangle = r_j \theta \). We take \( t_j \) as \( s_j \) and \( v_j' \) as \( u_j'' \).

\((ii)\) For each \( j \) such that \( r_j \in X \) and \( x_j'' = 1 \), we have \( s_j \xrightarrow{R^*_k} \langle q_j', x_j', v_j' \rangle = \langle q_j'', x_j', u_j'' \rangle \). We can construct \( s_j \xrightarrow{R^*_k} r_j \theta = \langle q_j', x_j', v_j' \rangle \) from \( s_j \in Q_{\text{NF}} \) from the induction hypothesis.

\((iii)\) For each \( j \) such that \( r_j \not\in X \), we have \( q_j'' = q_j' \) and \( x_j'' = 1 \), and \( u_j'' \) is an arbitrary state in \( Q_{\text{NF}} \). Since \( t_j \xrightarrow{R^*_k} \langle q_j', 1, u_j'' \rangle \) by Proposition 13, we have \( t_j = r_j \) from Proposition 11 (a) and the construction of \( \Delta_{RS} \). Therefore \( t_j = r_j \theta \) follows from right-shallowness. We take \( s_j \) as \( t_j \) and \( v_j' \) as \( u_j'' \).

\((iv)\) For each \( j \) such that \( r_j \not\in X \), we have \( s_j \xrightarrow{R^*_k} \langle q_j', x_j', v_j' \rangle \) for some \( v_j' \in Q_{\text{NF}} \) from the induction hypothesis. Since \( s_j \xrightarrow{R^*_k} \langle q_j', 1, v_j' \rangle \) by Proposition 14, we have \( s_j = r_j \) from Proposition 11 (a) and the construction of \( \Delta_{RS} \). Therefore \( s_j = r_j \theta \) follows from right-shallowness.

Thus we have \( g(s_1, \ldots, s_m) \xrightarrow{R^*_k} g(t_1, \ldots, t_m) \), \( \alpha': g(s_1, \ldots, s_m) \xrightarrow{R^*_k} g(r_1 \theta, \ldots, r_m \theta) \) such that \( \alpha \supseteq \alpha' \). From the construction of \( \Delta_{RS} \), we also have \( g(\langle q_1', x_1', v_1' \rangle, \ldots, \langle q_m', x_m', v_m' \rangle) \xrightarrow{R^*_k} \langle q, a, v' \rangle \).

We define a substitution \( \sigma : \text{Var}(f(l_1, \ldots, l_n)) \rightarrow T(F) \) as follows:

\[ x \sigma = \begin{cases} 
s_j \ldots & \text{if there exists } j \text{ such that } r_j = x 
s' \ldots & \text{otherwise, choose an arbitrary } t' \text{ such that } t' \xrightarrow{\Delta} x \theta 
\end{cases} \]

where \( x \) is well-defined from the right-linearity of rewrite rules. We can construct \( \beta : f(l_1, \ldots, l_n) \sigma \xrightarrow{R^*_k} f(\langle q_1, x_1, u_1 \rangle, \ldots, \langle q_m, x_m, u_m \rangle) \xrightarrow{R^*_k} \langle q, a, v \rangle \).

Since \( u_i \in Q_{\text{NF}} \) or \( x_i = 1 \), \( i \sigma \) is a normal form or \( i \not\in \mu(f) \) for each \( i \). Hence we have \( f(l_1, \ldots, l_n) \sigma \xrightarrow{R^*_k} g(r_1, \ldots, r_m) \sigma = g(s_1, \ldots, s_m) \).

Lemma 24 If \( R \) be linear and right-shallow, then \( L(A_k) \subseteq R^*_k \subseteq L(A_k) \).

Lemma 25 If \( R \) be linear and right-shallow, then \( L(A_k) = R^*_k \subseteq [A_k] \).

Lemma 26 Context-sensitive innermost reachability is decidable for linear right-shallow TRSs.
a TA $A_*$ such that $\mathcal{L}(A_*) = \overset{*}{\overset{\rightarrow}{R}}_{\text{in}}[\{t_1\}]$ from Lemma 25. Since the problem whether $t_2 \in \mathcal{L}(A_*)$ or not is decidable from Theorem 2, whether $t_1$ is reachable to $t_2$ by $R$ or not is also decidable. □

Theorem 27 Context-sensitive innermost joinability is decidable for linear right-shallow TRSs.

Proof: Let ground terms $t_1$, $t_2$ and CS-TRS $R$ be an instance of joinability problem. We can construct TAs $A^1$ and $A^2$ such that $\mathcal{L}(A^1) = \{t_1\}$ and $\mathcal{L}(A^2) = \{t_2\}$, and output $A^1$ and $A^2$ by $P_{\text{cisin}}$, such that $\mathcal{L}(A^1) = \overset{*}{\overset{\rightarrow}{R}}_{\text{in}}[\{t_1\}]$, $\mathcal{L}(A^2) = \overset{*}{\overset{\rightarrow}{R}}_{\text{in}}[\{t_2\}]$ from Lemma 25. We can construct the TA $A$ such that $\mathcal{L}(A) = \mathcal{L}(A^1) \cap \mathcal{L}(A^2)$, and it is decidable whether $\mathcal{L}(A)$ is empty or not from Theorem 2 (3). □

6. Conclusion

In this paper, we proved that context-sensitive innermost reachability and context-sensitive innermost joinability are decidable for linear right-shallow TRSs.

One of the future works is to show regularity preservation that is more powerful property than reachability and joinability. We say the relation $\rightarrow$ preserves regularity if we can construct a TA $A_*$ such that $\mathcal{L}(A_*) = \neg [\mathcal{L}(A)]$ for any TA $A$. To adjust $P_{\text{cisin}}$ for regularity preservation, one of possible way is modifying the third component of the states that occur in $r_j \in X$ such that $r_j \in \text{Var}(l)$, $r_j \notin \text{Pos}^a(l)$, and $r_j \in \text{Pos}^a(g(r_1, \ldots, r_m))$ for $l \rightarrow g(r_1, \ldots, r_m) \in R$. In this paper, the third component of such states is arbitrary $u \in Q_{\text{NF}}$, but we must have some restriction to $u$. We think it may be complex.

Another future work is to find other subclasses that context-sensitive innermost reachability is decidable. The class of non-linear shallow TRSs is one of candidates, because reachability and joinability of that class are undecidable for normal TRSs\(^5\),\(^10\), while they are decidable in innermost case\(^5\).

Acknowledgments We would like to thank the anonymous referees for their helpful comments and remarks. This work is partly supported by MEXT. KAKENHI #18500011, #20300010 and #20500008.

References

Context-sensitive Innermost Reachability is Decidable for Linear Right-shallow Term Rewriting Systems


(Received December 19, 2008)
(Received March 18, 2009)

Yoshiharu Kojima received the B.E. degree from Nagoya University. Now he is a student of graduate school of Information Science, Nagoya University. His research interest is term rewriting system.

Naoki Nishida graduated with the D.E. degree from the Graduate School of Engineering at Nagoya University in 2004. He became a Research Associate in the Graduate School of Information Science at Nagoya University in 2004. From April 2007, he has been an Assistant Professor. He is interested in term rewriting, program transformation, and program verification. He is a member of IEICE and JSSST.

Masahiko Sakai completed graduate course of Nagoya University in 1989 and became Assistant Professor, where he obtained the D.E. degree in 1992. From April 1993 to March 1997, he was Associate Professor in JAIST, Hokuriku. In 1996 he stayed at SUNY at Stony Brook for six months as Visiting Research Professor. From April 1997, he was an Associate Professor in Nagoya University. Since December 2002, he has been a Professor. He is interested in term rewriting system, verification of specification and software generation. He received the Best Paper Award from IEICE in 1992. He is a member of IEICE and JSSST.

Keiichirou Kusakari received the B.E. from Tokyo Institute of Technology in 1994, received the M.E. and Ph.D. degree 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 and JSSST.

Toshiaki Nishida graduated with the D.E. degree from the Graduate School of Engineering at Nagoya University in 2004. He became a Research Associate in the Graduate School of Information Science at Nagoya University in 2004. From April 2007, he has been an Assistant Professor. He is interested in term rewriting, program transformation, and program verification. He is a member of IEICE and JSSST.

Toshiki Sakabe received the 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, object-oriented computation, program verification, model checking and so on. He is a member of IEICE, JSAI and JSSST.