数理解析研究所講究録 1375 巻 2004 年 233-239

The Joinability and Unification Problems for Confluent Semi-Constructor TRSs

Ichiro Mitsuhashi, Michio Oyamaguchi, Yoshikatsu Ohta, Toshiyuki Yamada
(Faculty of Engineering, Mie University)
三橋一郎, 大山口通夫, 太田義勝, 山田俊行 (三重大学工学部)

Abstract

The unification problem for term rewriting systems (TRSs) is the problem of deciding, for a TRS R and two terms s and t, whether s and t are unifiable modulo R. Mitsuhashi et al. have shown that the problem is decidable for confluent simple TRSs. Here, a TRS is simple if the right-hand side of every rewrite rule is a ground term or a variable. In this paper, we extend this result and show that the unification problem for confluent semi-constructor TRSs is decidable. Here, a semi-constructor TRS is such a TRS that every subterm of the right-hand side of each rewrite rule is ground if its root is a defined symbol. We first show the decidability of joinability for confluent semi-constructor TRSs. Then, using the decision algorithm for joinability, we obtain a unification algorithm for confluent semi-constructor TRSs.

1 Introduction

The unification problem for term rewriting systems (TRSs) is the problem of deciding, for a TRS R and two terms s and t, whether s and t are unifiable modulo R. This problem is undecidable in general and even if we restrict to either right-ground TRSs [9] or terminating, confluent, monadic, and linear TRSs [7]. Here, a TRS is monadic if the height of the right-hand side of every rewrite rule is at most one [12]. On the other hand, it is known that unification is decidable for some subclasses of TRSs [2, 4, 5, 8, 11]. Recently, Mitsuhashi et al. have shown that the unification problem is decidable for confluent simple TRSs [7]. Here, a TRS is simple if the right-hand side of every rewrite rule is a ground term or a variable. In this paper, we extend the result of [7] and show that unification for confluent semi-constructor TRSs is decidable. Here, a semi-constructor TRS is such a TRS that every subterm of the right-hand side of each rewrite rule is ground if its root is a defined symbol.

In order to obtain this result, we first show the decidability of joinability for confluent semi-constructor TRSs. Joinability of several subclasses of TRSs has been shown to be decidable so far [13]. Many of these decidability results have been proved by reducing these problems to decidable ones for tree automata, so that these decidable subclasses are restricted to those of right-linear TRSs. In this paper, we provide a decidability result of joinability for possibly non-right-linear TRSs. To our knowledge, such attempts were very few so far.

Moreover, in this paper we show that confluence is necessary to show the decidability of joinability for semi-constructor TRSs, that is, joinability for (non-confluent) linear semi-constructor TRSs is undecidable.

2 Preliminaries

We assume that the reader is familiar with standard definitions of rewrite systems (see [1, 14]) and we just recall here the main notations used in this paper.

Let X be a set of variables, F a finite set of operation symbols graded by an arity function $\mathrm{ar}\colon F \to \mathbb{N}$, $F_n = \{f \in F \mid \mathrm{ar}(f) = n\}$, $Leaf = X \cup F_0$ the set of *leaf symbols*, and T the set of terms constructed from X and F. We use x, y, z as variables, b, c, d as constants, and r, s, t as terms. A term is *ground* if it has no variable. Let G be the set of ground terms and let $S = T \setminus (G \cup X)$. Let $\mathrm{V}(s)$ be the set of variables occurring in s. The *height* of s is defined as follows: $\mathrm{h}(a) = 0$ if a is a leaf symbol and

$h(f(t_1,\ldots,t_n)) = 1 + \max\{h(t_1),\ldots,h(t_n)\}$. The *root symbol* is defined as $\mathrm{root}(a) = a$ if a is a leaf symbol and $\mathrm{root}(f(t_1,\ldots,t_n)) = f$.

A position in a term is expressed by a sequence of positive integers, which are partially ordered by the prefix ordering $\leq$. To denote that positions u and v are disjoint, we use $u|v$. The subset of all minimal positions (w.r.t. $\leq$) of W is denoted by $\mathrm{Min}(W)$. Let $\mathcal{O}(s)$ be the set of positions of s.

Let $s_{|u}$ be the subterm of s at position u. We use $s[t]_u$ to denote the term obtained from s by replacing the subterm $s_{|u}$ by t. For a sequence $(u_1,\cdots,u_n)$ of pairwise disjoint positions and terms $r_1,\cdots,r_n$, we use $s[r_1,\cdots,r_n]_{(u_1,\ldots,u_n)}$ to denote the term obtained from s by replacing each subterm $s_{|u_i}$ by $r_i (1 \leq i \leq n)$.

A *rewrite rule* is defined as a directed equation $\alpha \to \beta$ that satisfies $\alpha \notin X$ and $V(\alpha) \supseteq V(\beta)$. Let $\leftarrow$ is the inverse of $\to$, $\leftrightarrow = \to \cup \leftarrow$ and $\downarrow = \to^* \cdot \leftarrow^*$. Let $\gamma\colon s_1 \overset{u_1}{\leftrightarrow} s_2 \cdots \overset{u_{n-1}}{\leftrightarrow} s_n$ be a *rewrite sequence.* This sequence is abbreviated to $\gamma\colon s_1 \leftrightarrow^* s_n$ and $\mathcal{R}(\gamma) = \{u_1,\cdots,u_{n-1}\}$ is the set of the redex positions of γ. If the root position ε is not a redex position of γ, then γ is called ε-invariant. For any sequence γ and position set W, $\mathcal{R}(\gamma) \geq W$ if for any $v \in \mathcal{R}(\gamma)$ there exists a $u \in W$ such that $v \geq u$. If $\mathcal{R}(\gamma) \geq W$, we write $\gamma\colon s_1 \overset{\geq W}{\leftrightarrow^*} s_n$.

Let $\mathcal{O}_G(s) = \{u \in \mathcal{O}(s) \mid s_{|u} \in G\}$. For any set $\Delta \subseteq X \cup F$, let $\mathcal{O}_\Delta(s) = \{u \in \mathcal{O}(s) \mid \mathrm{root}(s_{|u}) \in \Delta\}$. Let $\mathcal{O}_x(s) = \mathcal{O}_{\{x\}}(s)$. The set D of *defined symbols* for a TRS R is defined as $D = \{\mathrm{root}(\alpha) \mid \alpha \to \beta \in R\}$. A term s is *semi-constructor* if, for every subterm t of s such that $\mathrm{root}(t)$ is a defined symbol, t is ground.

Definition 1 A rule $\alpha \to \beta$ is *ground* if $\alpha, \beta \in G$, *right-ground* if $\beta \in G$, *semi-constructor* if β is semi-constructor, and *linear* if $|\mathcal{O}_x(\alpha)| \leq 1$ and $|\mathcal{O}_x(\beta)| \leq 1$ for every $x \in X$.

Example 2 *Let* $R_e = \{\mathsf{nand}(x,x) \to \mathsf{not}(x),\ \mathsf{nand}(\mathsf{not}(x),x) \to \mathsf{t},\ \mathsf{t} \to \mathsf{nand}(\mathsf{f},\mathsf{f}),\ \mathsf{f} \to \mathsf{nand}(\mathsf{t},\mathsf{t})\}$.

R_e is semi-constructor, non-terminating and confluent [3]. We will use this R_e in examples given in Section 3.

Definition 3 [11] An equation is a pair of terms s and t denoted by $s \approx t$. An equation $s \approx t$ is *unifiable modulo* a TRS R (or simply *R-unifiable*) if there exists a substitution θ and a rewrite sequence γ such that $\gamma\colon s\theta \leftrightarrow^* t\theta$. Such θ and γ are called an *R-unifier* and a *proof* of $s \approx t$, respectively. This notion is extended to sets of term pairs: for $\Gamma \subseteq T \times T$, θ is an R-unifier of Γ if θ is an R-unifier of every pair in Γ. In this case, Γ is R-unifiable. As a special case of R-unifiability, $s \approx t$ is $\emptyset$-unifiable if there exists a substitution θ such that $s\theta = t\theta$, i.e., $\emptyset$-unifiability coincides with the usual unifiability. If $s \downarrow t$ then $s \approx t$ is *joinable.* If $s \to^* t$ then $s \approx t$ is *reachable.*

Definition 4 TRSs R and R' are *equivalent* if $\leftrightarrow^*_R = \leftrightarrow^*_{R'}$.

3 Joinability

First, we show that the joinability and reachability problems for (non-confluent) semi-constructor TRSs are undecidable.

Theorem 5 The joinability and reachability problems for linear semi-constructor term rewriting systems are undecidable. **Proof** [sketch] The proof is by a reduction from the Post's correspondence problem (PCP). Let $P = \{\langle u_i, v_i\rangle \in \Sigma^* \times \Sigma^* \mid 1 \leq i \leq k\}$ be an instance of the PCP. The corresponding TRS R_P is constructed as follows: Let $F_0 = \{\mathsf{c},\mathsf{d},\$\}, F_1 = \Sigma \cup \{\mathsf{f},\mathsf{h}\}, F_2 = \{\mathsf{g}\}$; $R_P = \{\mathsf{c} \to \mathsf{h}(\mathsf{c}), \mathsf{c} \to \mathsf{d}, \mathsf{d} \to \mathsf{f}(\mathsf{d})\} \cup \{\mathsf{d} \to \mathsf{g}(u_i(\$), v_i(\$)), \mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(u_i(x), v_i(y)) \mid 1 \leq i \leq k\} \cup \{\mathsf{h}(\mathsf{g}(a(x),a(y))) \to \mathsf{g}(x,y) \mid a \in \Sigma\}$. $u(x)$ is an abbreviation for $a_1(a_2(\cdots a_k(x)))$ where $u = a_1 a_2 \cdots a_k$ with $a_1,\cdots,a_k \in \Sigma$. R_P is linear and semi-constructor. For R_P, the following three propositions (1)–(3) are equivalent: (1) $\mathsf{c} \downarrow \mathsf{g}(\$,\$)$, (2) $\mathsf{c} \to^* \mathsf{g}(\$,\$)$, and (3) PCP P has a solution. □

3.1 Standard Semi-Constructor TRSs

From now on, we consider only confluent semi-constructor TRSs, for which joinability is shown to be decidable. In order to facilitate the decidability proof, we transform a TRS into a simpler equivalent one.

Definition 6 For TRS R, we use R_{rg} and $\overline{R_{\mathrm{rg}}}$ to denote the sets of right-ground and non-right-ground rewrite rules in R, respectively.

If R is clear from the context, we write $\to_{\mathrm{rg}}$ instead of $\to_{R_{\mathrm{rg}}}$.

Definition 7 A TRS R is *standard* if the following condition holds: for every $\alpha \to \beta \in R$ either $\alpha \in F_0$ and $\mathrm{h}(\beta) \leq 1$ or $\alpha \notin F_0$ and for every $u \in \mathcal{O}(\beta)$ if $\beta_{|u} \in G$ then $\beta_{|u} \in F_0$.

Let R_0 be a confluent semi-constructor TRS. The corresponding standard TRS $R^{(\mathrm{i})}$ is constructed as follows. First, we choose $\alpha \to \beta \in R_k (k \geq 0)$ that does not satisfy the standardness condition. If $\alpha \in F_0$ then let $\{u_1, \cdots, u_m\} = \mathrm{Min}(\mathcal{O}_G(\beta) \setminus \mathcal{O}_{F_0}(\beta) \setminus \{\varepsilon\})$, else let $\{u_1, \cdots, u_m\} = \mathrm{Min}(\mathcal{O}_G(\beta) \setminus \mathcal{O}_{F_0}(\beta))$. Let $R_{k+1} = R_k \setminus \{\alpha \to \beta\} \cup \{\alpha \to \beta[d_1, \cdots, d_m]_{(u_1, \cdots, u_m)}\} \cup \{d_i \to \beta_{|u_i} \mid 1 \leq i \leq m\}$ where $d_1, \cdots, d_m$ are new pairwise distinct constants which do not appear in R_k. This procedure is applied repeatedly until the TRS satisfies the condition of standardness. The resulting TRS is denoted by $R^{(\mathrm{i})}$. For example, $\{\mathsf{f}_1(x) \to \mathsf{g}(x, \mathsf{g}(\mathsf{a}, \mathsf{b})), \mathsf{f}_2(x) \to \mathsf{f}_2(\mathsf{g}(\mathsf{c}, \mathsf{d}))\}$ is transformed to $\{\mathsf{f}_1(x) \to \mathsf{g}(x, \mathsf{d}_1), \mathsf{d}_1 \to \mathsf{g}(\mathsf{a}, \mathsf{b}), \mathsf{f}_2(x) \to \mathsf{d}_2, \mathsf{d}_2 \to \mathsf{f}_2(\mathsf{d}_3), \mathsf{d}_3 \to \mathsf{g}(\mathsf{c}, \mathsf{d})\}$. This transformation preserves confluence, joinability and unifiability.

Lemma 8

(1) $R^{(\mathrm{i})}$ is confluent.

(2) For any terms s, t which do not contain new constants, $s \downarrow_{R_0} t$ iff $s \downarrow_{R^{(\mathrm{i})}} t$.

(3) For any terms s, t which do not contain new constants, $s \approx t$ is R_0-unifiable iff $s \approx t$ is $R^{(\mathrm{i})}$-unifiable.

The proof is straightforward, since R_0 is confluent. By this lemma, we can assume that a given confluent semi-constructor TRS is standardized without loss of generality. By standardization, for any $\alpha \to \beta \in R_{\mathrm{rg}}$, $\alpha \in F_0$ or $\beta \in F_0$ holds and $\mathrm{h}(\beta) \leq 1$. However, by the transformation algorithm given in Section 3.2, the heights of the right-hand sides of ground rules (called R_C type rules later) may increase. This is the only exceptional case.

3.2 Adding Ground Rules

The joinability for right-ground TRSs is decidable [10]. In this paper, we show that the joinability for confluent semi-constructor TRSs is decidable, by reducing to the joinability for right-ground TRSs.

Let R_1 be a confluent TRS and R_2 be such a TRS that $\to_{R_2} \subseteq \downarrow_{R_1}$. Then, obviously $R_1 \cup R_2$ is equivalent to R_1 and confluent. Thus, even if we add pairs of joinable terms of R_1 to R_1 as new rewrite rules (called shortcuts), confluence, joinability and unifiability properties are preserved. Note that reachability is not necessarily preserved. Now, we show that the joinability of confluent semi-constructor TRSs reduces to that of right-ground TRSs by adding new finite ground rules. For this purpose, we need some definitions.

Definition 9 A rule $\alpha \to \beta$ has *type C* if $\alpha \in F_0, \beta \notin F_0$ and $\mathcal{O}_{D \setminus F_0}(\beta) = \emptyset$, and has *type* F_0 if $\alpha, \beta \in F_0$. Let $R_\tau = \{\alpha \to \beta \in R \mid \alpha \to \beta \text{ has type } \tau\}$.

That is, R_C is the subset of R_{rg} satisfying that for every rule $\alpha \to \beta \in R_C$, α is a constant, and β is non-constant and contains no defined non-constant symbol. Henceforth, we assume that $R \setminus R_C$ is standard.

Definition 10

$$\mathrm{h_D}(s) = \begin{cases} w + \max\{\mathrm{h_D}(s_i) \mid 1 \leq i \leq n\} & (\text{if } s = f(s_1, \cdots, s_n), n > 0, f \in D) \\ 1 + \max\{\mathrm{h_D}(s_i) \mid 1 \leq i \leq n\} & (\text{if } s = f(s_1, \cdots, s_n), n > 0, f \notin D) \\ 0 & (\text{if } s \in \mathit{Leaf}) \end{cases}$$

where $w = 1 + 2\max\{\mathrm{h}(\beta) \mid \alpha \to \beta \in R\}$. Note that we give weight w to each defined non-constant symbol and 1 to each other non-constant symbol and define new heights derived from these weights. We define $\mathrm{H_D}(s) = \{\mathrm{h_D}(s_{|u}) \mid u \in \mathcal{O}(s)\}_{\mathrm{m}}$, which is a multiset of heights of all subterms of s. Here, we use $\{\cdots\}_{\mathrm{m}}$ to denote a multiset and $\sqcup$ to denote multiset union. For TRS R_{e} of Example 1, $w = 3$ and $\mathrm{H_D}(\mathsf{nand}(\mathsf{not}(x), x)) = \{0, 0, 1, 4\}_{\mathrm{m}}$.

Let $\ll$ be the multiset extension of the usual relation $<$ on $\mathbb{N}$ and let $\lll$ be $\ll \cup =$. Let $\#(s) = (\mathrm{H_D}(s), \mathrm{g}(s))$. Here, function $\mathrm{g}(s)$ returns a natural number corresponding to s uniquely, and we assume that the ordering derived by this function is closed under context, i.e., for any terms r, s, t and any position $u \in \mathcal{O}(r)$, if $\mathrm{g}(s) < \mathrm{g}(t)$ then $\mathrm{g}(r[s]_u) < \mathrm{g}(r[t]_u)$. Such a function g is effectively computable. In order to compare $\#(s)$ and $\#(t)$, we use lexicographic order $<_{\mathrm{lex}}$. Note that $<_{\mathrm{lex}}$ is a total order. *A term* s_0 *is minimum in set* Δ iff $s_0 \in \Delta$ and $\#(s_0) = \mathrm{Min}(\{\#(s') \mid s' \in \Delta\})$.

Definition 11

(1) Function linearize(s) linearizes non-linear term s in the following manner. For each variable occurring more than once in s, the first occurrence is not renamed, and the other ones are replaced by new pairwise distinct variables. For example, linearize($\mathsf{nand}(x, x)$) = $\mathsf{nand}(x, x_1)$. If function linearize replaces x by x_1, then we use $x \equiv x_1$ to denote the replacement relation.

(2) For set $\Delta \subseteq T$, $\mathrm{Psub}(\Delta) = \{s_{|u} \mid s \in \Delta, u \in \mathcal{O}(s) \setminus \{\varepsilon\}\}$.

(3) For set $\Delta \subseteq T$, $\mathrm{Bud}(\Delta, R_C) = F_0 \cup \mathrm{Psub}(\Delta \cup \{\beta \mid \alpha \to \beta \in R_C\})$. Note that if $\Delta \subseteq F_0$ then $\mathrm{Bud}(\Delta, R_C) = \mathrm{Bud}(\emptyset, R_C)$.

(4) Substitution σ is *joinability preserving under relation* $\equiv$ *for TRS* R_{rg} if $x\sigma \downarrow_{R_{\mathrm{rg}}} x'\sigma$ whenever $x \equiv x'$. In this case, we write $\sigma \in\downarrow (\equiv, R_{\mathrm{rg}})$.

(5) For TRS R and term α, $R(\alpha) = \{\beta \mid \alpha \to \beta \in R\}$.

(6) Let $\{s_1, \cdots, s_m\} = R_C(d)$ and $\{u_1, \cdots, u_n\} = \mathrm{Min}(\cup_{1 \le i \le m} \mathcal{O}_{F_0}(s_i))$. Let d_j be the minimum term in $\{s_{i|u_j} \in F_0 \mid 1 \le i \le m\}$, $1 \le j \le n$. Then we define $\mathrm{Normalize}(d, R_C) = \{d \to s_1[d_1, \cdots, d_n]_{(u_1, \cdots, u_n)}\} \cup \{d_j \to s_{i|u_j} \mid 1 \le i \le m, 1 \le j \le n, d_j \ne s_{i|u_j}\}$. For example, Normalize(t, {t → not(not(t)), t → not(f)}) = {t → not(f), f → not(t)}.

Lemma 12 Let $R \backslash R_C$ be standard. Let $\alpha \to \beta \in \overline{R_{\mathrm{rg}}}$, $\theta : X \to T$ and $s \to^*_{R_{\mathrm{rg}}} \alpha\theta$. Let $\alpha' = \mathrm{linearize}(\alpha)$. Then, there exists a substitution $\sigma : \mathrm{V}(\alpha') \to \mathrm{Bud}(\{s\}, R_C)$ such that $s \to^*_{R_{\mathrm{rg}}} \alpha'\sigma \to^*_{R_{\mathrm{rg}}} \alpha\theta$, $\beta\sigma \to^*_{R_{\mathrm{rg}}} \beta\theta$ and $\sigma \in\downarrow (\equiv, R_{\mathrm{rg}})$.

By Lemma 12, for a rewrite sequence $d \to^*_{R_{\mathrm{rg}}} \alpha\theta \to \beta\theta$, there exists $\alpha'\sigma$ such that $d \to^*_{R_{\mathrm{ru}}} \alpha'\sigma \to^*_{R_{\mathrm{rg}}} \alpha\theta$ and $\beta\sigma \to^*_{R_{\mathrm{rg}}} \beta\theta$. So, if we add a new ground rule $d \to \beta\sigma$ to R, then we have $d \to^*_{R'} \beta\theta$ for $R' = R_{\mathrm{rg}} \cup \{d \to \beta\sigma\}$. Thus, by adding shortcut rules such as $d \to \beta\sigma$, we can omit applications of $\alpha \to \beta$ which is a non-right-ground rule. Using this technique, the following algorithm takes as input a standard semi-constructor TRS $R^{(\mathrm{i})}$ and produces as output an equivalent semi-constructor TRS $R^{(\mathrm{f})}$ satisfying that if $d \to^*_{R^{(\mathrm{i})}} s$ then $d \to^*_{R^{(\mathrm{f})}_{\mathrm{rg}}} s$. We call $R^{(\mathrm{f})}$ a *quasi-right-ground TRS*, hereafter.

function MakeQuasiRightGround(R)
  $R :=$ Determinize(R);
  repeat
    $R' := R$;
    $R :=$ Determinize(AddShortcuts(R'))
  until $R = R'$;
  return R

function AddShortcuts(R)
  $R' := R$;
  for each $\alpha \to \beta \in \overline{R_{\mathrm{rg}}}$ **do**
    $\alpha' :=$ linearize(α);
    for each $d \in F_0, \sigma : \mathrm{V}(\alpha') \to \mathrm{Bud}(\emptyset, R_C)$ such that $\sigma \in\downarrow (\equiv, R_{\mathrm{rg}})$ **do**
      if $d \to^*_{R_{\mathrm{rg}}} \alpha'\sigma$ **then** $R' := R' \cup \{d \to \beta\sigma\}$
  return R'

function Determinize(R)
  while there exists d such that $|R_C(d)| > 1$ **do**
    $R := R \cup \mathrm{Normalize}(d, R_C) \setminus \{d \to s \mid d \to s \in R_C\}$
  return R

Example 13 *For TRS* R_e *of Example 1,* MakeQuasiRightGround(R_e) *first computes* Determinize(R_e). *It returns the same* R_e *as output. Next,* AddShortcuts(R_e) *is called. Since* t → nand(f, f), nand(x, x) → not(x) ∈ R_e*, a new shortcut rule* t → not(f) *is added to* R_e*. Similarly,* f → not(t) *is added. Thus,* AddShortcuts(R_e) = R' *where* $R' = R_e \cup \{$t → not(f), f → not(t)$\}$*. Next,* Determinize(R') *is called and* *returns the same* R' *as output. Then,* AddShortcuts(R') *is called. Note that* $R'_C = \{\mathsf{t} \to \mathsf{not(f)}, \mathsf{f} \to \mathsf{not(t)}\}$. AddShortcuts($R'$) *returns the same* R' *and also calls* Determinize(R'). *Then, the algorithm halts. Let* $R_e^{(f)}$ *be this result:* $R_e^{(f)} = R_e \cup \{\mathsf{t} \to \mathsf{not(f)}, \mathsf{f} \to \mathsf{not(t)}\}$, *which will be used in later examples.*

We apply this algorithm to standard TRS. But by an application of this algorithm, the heights of some right-hand side terms of type C rules may become greater than 1. This algorithm satisfies the following lemmata.

Lemma 14 MakeQuasiRightGround is terminating.

Lemma 15 Let $R^{(f)} = \text{MakeQuasiRightGround}(R^{(i)})$.

(1) If $d \to^*_{R^{(i)}} s$ then $d \to^*_{R^{(f)}_{rg}} s$.

(2) $\to_{R^{(f)}} \subseteq \downarrow_{R^{(i)}}$.

Corollary 16

(1) $R^{(f)}$ is confluent (since $R^{(i)}$ is confluent).

(2) $c \downarrow_{R^{(i)}} d$ iff $c \downarrow_{R^{(f)}_{rg}} d$.

(3) $s \approx t$ is $R^{(i)}$-unifiable iff $s \approx t$ is $R^{(f)}$-unifiable.

3.3 Auxiliary Terms

We have shown that all rewrite sequences from every constant in $R^{(i)}$ (i.e., $d \to^*_{R^{(i)}} s$) can be obtained by using only right-ground rules (i.e., $d \to^*_{R^{(f)}_{rg}} s$). Now, we want to extend this result to that for rewrite sequences from any term. For this purpose, we need the notion of auxiliary terms. For $\Delta \subseteq G$

function Aux(Δ)
  repeat
    $\Delta' := \Delta$;
    $\Delta :=$ AddTerms(Δ')
  until $\Delta = \Delta'$;
  return Δ

function AddTerms(Δ)
  $\Delta' := \Delta$;
  for each $\alpha \to \beta \in \overline{R^{(f)}_{rg}}$ **do**
    $\alpha' :=$ linearize(α);
    for each $s \in \Delta, p \in \mathcal{O}_{D \setminus F_0}(s)$,
        $\sigma : \mathrm{V}(\alpha') \to \mathrm{Bud}(\{s_{|p}\}, R^{(f)}_C)$ such that $\sigma \in\downarrow (\equiv, R^{(f)}_{rg})$ **do**
      if $s_{|p} \to^*_{R^{(f)}_{rg}} \alpha'\sigma$ **then** $\Delta' := \Delta' \cup \{s[\beta\sigma]_p\}$
  return Δ'

Example 17 *In TRS* $R_e^{(f)}$ *of Example 2,*
Aux({not(nand(t, t))}) = AddTerms({not(nand(t, t))}) = {not(nand(t, t)), not(not(t))}.

Lemma 18 For any ground term s,

(1) For any $s' \in \text{Aux}(\{s\})$, $\text{Aux}(\{s'\}) \subseteq \text{Aux}(\{s\})$.

(2) Aux($\{s\}$) is finite and computable.

(3) For any $s' \in \text{Aux}(\{s\})$, $s' \downarrow_{R^{(f)}} s$.

(4) If $s \to^*_{R^{(i)}} t$ then there exists $s' \in \text{Aux}(\{s\})$ such that $s' \to^*_{R^{(f)}_{rg}} t$.

We call s' in Lemma 18(4) an *auxiliary term* of (s, t). This will be used to transform non-right-ground rewrite sequence to right-ground rewrite sequence.

Example 19 *For rewrite sequence* not(nand(t, t)) $\to^*_{rg}$ not(nand(not(f), not(f))) $\to$ not(not(not(f)), *we can choose* not(not(t)) $\in$ Aux({not(nand(t, t))}) *and* not(not(t)) $\to_{rg}$ not(not(not(f))).

3.4 Joinability for Confluent Semi-Constructor TRSs

Lemma 20 For any ground terms s and t, $s \downarrow_{R^{(i)}} t$ iff there exists $s' \in \mathrm{Aux}(\{s\}), t' \in \mathrm{Aux}(\{t\})$ such that $s' \downarrow_{R^{(i)}_{\mathrm{rg}}} t'$.

By Lemma 18(2) and decidablity of $s' \downarrow_{R^{(i)}_{\mathrm{rg}}} t'$ [10], $s \downarrow_{R^{(i)}} t$ is decidable for ground terms s and t. If s or t is non-ground, $s \downarrow_{R^{(i)}} t$ is equivalent to $s\sigma \downarrow_{R^{(i)}} t\sigma$ where $\sigma : \mathrm{V}(s) \cup \mathrm{V}(t) \to F'_0$ is a bijection and F'_0 is a set of new pairwise distinct constants which do not appear in $R^{(i)}$. Thus, we have the following theorem.

Theorem 21 The joinability for confluent semi-constructor term rewriting systems is decidable.

By confluence, we have the following corollary too.

Corollary 22 The word problem for confluent semi-constructor term rewriting systems is decidable.

4 R-Unification

By using Theorem 21, we have the following theorem.

Theorem 23 The unification problem for confluent semi-constructor term rewriting systems is decidable.

5 Conclusion

In this paper, we have shown that the joinability and unification problems for confluent semi-constructor TRSs are decidable. But, reachability remains open. Obviously, the class of semi-constructor TRSs is a subclass of strongly weight-preserving TRSs, for which several sufficient conditions to ensure confluence are given in [3].

Acknowledgements

This work was supported in part by Grand-in-Aid for Scientific Research 15500009 from Japan Society for the Promotion of Science.

References

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

[2] H. Comon, M. Haberstrau, and J.-P. Jouannaud. Syntacticness, cycle-syntacticness and shallow theories. *Information and Computation*, 111(1):154–191, 1994.

[3] H. Gomi, M. Oyamaguchi, and Y. Ohta. On the Church-Rosser property of root-E-overlapping and strongly depth-preserving term rewriting systems. *Transactions of Information Processing Society of Japan*, 39(4):992–1005, 1998.

[4] J.-M. Hullot. Canonical forms and unification. In *Proc. 5th Conf. on Automated Deduction*, pages 318–334. LNCS 87, 1980.

[5] F. Jacquemard, C. Meyer, and C. Weidenbach. Unification in extensions of shallow equational theories. In *Proc. 9th Rewriting Techniques and Applications*, pages 76–90. LNCS 1379, 1998.

[6] A. Martelli and G. Rossi. Efficient unification with infinite terms in logic programming. In *Proc. 5th Generation Computer Systems*, pages 202–209, 1984.

[7] I. Mitsuhashi, M. Oyamaguchi, Y. Ohta, and T. Yamada. On the unification problem for confluent monadic term rewriting systems. *IPSJ Transactions on Programming*, 44(SIG 4(PRO 17)):54–66, 2003.

[8] R. Nieuwenhuis. Basic paramodulation and decidable theories. In *Proc. 11th IEEE Symp. Logic in Computer Science*, pages 473–482, 1996.

[9] M. Oyamaguchi. On the word problem for right-ground term-rewriting systems. *Trans. IEICE*, E73(5):718–723, 1990.

[10] M. Oyamaguchi. The reachability and joinability problems for right-ground term-rewriting systems. *Journal of Information Processing*, 13(3):347–354, 1990.

[11] M. Oyamaguchi and Y. Ohta. The unification problem for confluent right-ground term rewriting systems. *Information and Computation*, 183(2):187–211, 2003.

[12] K. Salomaa. Deterministic tree pushdown automata and monadic tree rewriting systems. *Jornal of Computer and System Sciences*, 37:367–394, 1988.

[13] T. Takai, Y. Kaji, and H. Seki. Right-linear finite path overlapping term rewriting systems effectively preserve recognizability. In *Proc. 11th Rewriting Techniques and Applications*, pages 246–260.

[14] Terese. *Term Rewriting Systems*. Cambridge University Press, 2003.