社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

準構成子項書換え系における停止性の決定問題

王　　易†　酒井　正彦††　西田　直樹††　草刈　圭一朗††　坂部　俊樹††

†,†† 名古屋大学大学院情報科学研究科
〒 464-8603 名古屋市千種区不老町
E-mail: †ywang80@trs.cm.is.nagoya-u.ac.jp, ††{sakai,nishida,kusakari,sakabe}@is.nagoya-u.ac.jp

あらまし　本稿では，右基礎項書換え系の停止性が決定可能であるという既存の結果を準構成子項書換え系に一般化する．ここで，項書換え系の規則の右辺のすべての被定義記号のどの引数も基礎項であるときに，準構成子項書換え系という．準構成子項書換え系の無限依存鎖の存在性が決定可能であることを示すことにより，準構成子項書換え系の停止性が決定可能であることを証明する．

キーワード　依存対，右基礎，決定手続き

Decidability of Termination for Semi-Constructor Term Rewriting Systems

Yi WANG†, Masahiko SAKAI††, Naoki NISHIDA††,
Keiichiro KUSAKARI††, and Toshiki SAKABE††

†,†† Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
E-mail: †ywang80@trs.cm.is.nagoya-u.ac.jp, ††{sakai,nishida,kusakari,sakabe}@is.nagoya-u.ac.jp

Abstract　It is known that termination is decidable for right-ground term rewriting systems (TRSs for short). In this paper, we show that termination is also decidable for semi-constructor TRSs, class of which is a super class of right-ground case. Here, a TRS is a semi-constructor system if every defined symbol in the right-hand sides of rules takes ground terms as its arguments. The proof is constructed by showing the decidability of the absence of the infinite dependency chains for semi-constructor TRSs, which implies the termination of the TRSs.

Key words　dependency pair, right-ground, decision procedure

1. Introduction

55% %

Termination is an important property of term rewriting systems (TRSs for short). A TRS terminates if it does not admit any infinite rewrite sequences; conversely, it does not terminate if admitting an infinite rewrite sequence. Unfortunately, termination is an undecidable property of TRSs in general. This is true even if one allows for only unary function symbols in the rules, or for only one rewrite rule. In some restricted case of TRSs, termination becomes a decidable property. The decidability result of right-ground TRSs can be found in the reference [6]. It is the obvious generalization to the right-ground case of Huet and Lankford's result [3] for ground rewrite systems. A detailed proof of the decidability for right-ground TRSs can also be found in the reference [2].

In this paper, we consider the decidability problem of termination for semi-constructor TRSs. Here, a TRS is a semi-constructor system if every defined symbol in the right-hand sides of rules takes ground terms as its arguments. We give a positive answer to this problem that termination of semi-constructor TRSs is decidable.

Concerning automated termination proofs for term rewriting systems, the dependency pair approach proposed in the reference [8] is one of the most powerful techniques. Based on the notion of dependency pair, we prove our results by showing the decidability of the absence of the infinite dependency chains for semi-constructor TRSs, which implies the termination of the TRSs. As the class of right-ground TRSs forms a proper subclass of semi-constructor TRSs, we point out that our result can be viewed as the generalization to the semi-constructor TRSs of the result for right-ground case.

The organization of this paper is as follows: in section 2 we review the preliminary definitions of term rewriting systems; in section 3 we give the detailed termination decision procedure for semi-constructor TRSs; in section 4 we first give some related results concerning loop, head-loop and cycles, then prove the correctness of the decision procedure proposed in the last section; in section 5 we compare our results with the existing results.

2. Preliminaries

We assume the reader is familiar with the standard definitions of term rewriting systems [2] and here we just review the main notations used in this paper.

A *signature* $\mathcal{F}$ is a set of function symbols, where every $f \in \mathcal{F}$ is associated with a non-negative integer number by an arity function: $arity\colon \mathcal{F} \to \mathbb{N}(= \{0, 1, 2, \ldots\})$. Function symbols of arity 0 are called *constant symbols*. The set $\mathcal{T}(\mathcal{F}, \mathcal{V})$ of all *terms* built from a signature $\mathcal{F}$ and a countable infinite set $\mathcal{V}$ of *variables* such that $\mathcal{F} \cap \mathcal{V} = \emptyset$, is represented by $\mathcal{T}(\mathcal{F}, \mathcal{V})$. The set of *ground terms* is denoted by $\mathcal{T}(\mathcal{F}, \emptyset)$ ($\mathcal{T}(\mathcal{F})$ for short). We write $s = t$ when two terms s and t are identical. The *root symbol* of a term t is denoted by $root(t)$.

The set of all *positions* in a term t is denoted by $\mathcal{P}os(t)$. Here ε represents the root position. Positions are partially ordered by the prefix order $\leqq$, that is, $p \leqq q$ if there exists an r such that $p \cdot r = q$. We write $p < q$ if $p \leqq q$ and $p \neq q$. If $p \in \mathcal{P}os(t)$, then $t|_p$ denotes the *subterm* of t at position p, and $t[s]_p$ denotes the term that is obtained from t by replacing the subterm at position p by the term s. We denote by $\trianglerighteq$ the *subterm ordering*, that is, $t \trianglerighteq u$ if u is a subterm of t, and $t \triangleright u$ if $t \trianglerighteq u$ and $t \neq u$.

Let C be a *context* with a hole $\square$. We write $C[t]$ for the term obtained from C by replacing $\square$ with a term t.

A *substitution* θ is a mapping from $\mathcal{V}$ to $\mathcal{T}(\mathcal{F}, \mathcal{V})$ such that the set $\{x \in \mathcal{V} \mid \theta(x) \neq x\}$ called the *domain* of θ and denoted by $\mathcal{D}om(\theta)$ is finite. We usually identify a substitution θ with the set $\{x \mapsto \theta(x) \mid x \in \mathcal{D}om(\theta)\}$ of variable bindings. We denote $\mathcal{R}an(\theta) = \{\theta(x) \mid x \in \mathcal{D}om(\theta)\}$ as the *range* of θ. Substitutions are naturally extended to homomorphisms from $\mathcal{T}(\mathcal{F}, \mathcal{V})$ to $\mathcal{T}(\mathcal{F}, \mathcal{V})$. In the following, we write $t\theta$ instead of $\theta(t)$. The composition $\theta_1\theta_2$ of two substitutions θ_1 and θ_2 is defined by $x(\theta_1\theta_2) = (x\theta_1)\theta_2$ for all $x \in \mathcal{V}$.

A *rewrite rule* $l \to r$ is a directed equation which satisfies $l \notin \mathcal{V}$ and $\mathcal{V}ar(r) \subseteqq \mathcal{V}ar(l)$. We call l the *left-hand side* and r the *right-hand side* of the rewrite rule. A *term rewriting system* TRS is a set of finite rewrite rules. The *rewrite relation* $\to_R \subseteqq \mathcal{T}(\mathcal{F}, \mathcal{V}) \times \mathcal{T}(\mathcal{F}, \mathcal{V})$ associated with a TRS R is defined as follows: $s \to_R t$ if there exist a rewrite rule $l \to r \in R$, a substitution θ, and a position $p \in \mathcal{P}os(s)$ such that $s|_p = l\theta$ and $t = s[r\theta]_p$. The subterm $l\theta$ of s is called a *redex* and we say that s rewrites to t by contracting redex $l\theta$. We say that p is a redex position. The transitive closure of $\to_R$ is denoted by $\to_R^+$. The transitive and reflexive closure of $\to_R$ is denoted by $\to_R^*$. If $s \to_R^* t$, then we say that there is a *rewrite sequence* starting from s, which reduces to t. We use $\xrightarrow{\varepsilon}_R$ to denote the root rewrite step and $\xrightarrow{>\varepsilon}_R$ to denote the rewrite step where redex position occurs below the root. A term without redexes is called a *normal form*. We say that a term t *has* a normal form if there exists a rewrite sequence starting from t that reduces to a normal form. A TRS R is called *right-ground* if the right-hand sides of all the rules in R are ground terms, and *ground* if both sides of all the rules are ground terms.

For a TRS R, a term $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ *terminates* if there is no infinite rewrite sequences starting from t. R terminates over $T \subseteqq \mathcal{T}(\mathcal{F}, \mathcal{V})$ if all terms in T terminate. In the case that $T = \mathcal{T}(\mathcal{F}, \mathcal{V})$, we say that R terminates.

For a TRS R, a function symbol $f \in \mathcal{F}$ is a *defined symbol* of R if $f = root(l)$ for some rewrite rule $l \to r \in R$. The set of all the defined symbols of R is denoted by $D_R = \{root(l) \mid \exists l \to r \in R\}$. We write C_R for the set of all the *constructor symbols* of R which is defined as $\mathcal{F} \backslash D_R$. A term t has a *defined root symbol* if $root(t) \in D_R$.

3. Decision Procedure

In this section, we first give the definition of semi-constructor TRSs proposed in the reference [5]. Then we propose the decision procedure of termination for semi-constructor TRSs in details. Then we give a theorem which assures the correctness of the procedure.

Definition 3.1 (semi-constructor TRS) *For a TRS R, a term t is a* semi-constructor *term if s is ground for every subterms t of t having root defined symbol. R is a semi-constructor system if for every rule $l \to r \in R$, r is a semi-constructor term.*

Example 3.2 TRS $R_1 = \{f(x) \to h(x, f(g(a))), g(x) \to g(h(a, a))\}$ is a semi-constructor system.

For a set $T \subseteqq \mathcal{T}(\mathcal{F}, \mathcal{V})$, we define $\to_R T = \{s \mid \exists t \in T, t \to_R s\}$. The decision procedure of termination for semi-constructor TRSs is shown as follows.

Decision Procedure SC_T
function SC_T
/* Input: semi-constructor TRS $R = \{l_i \to r_i \mid 1 \leqq i \leqq n\}$ */
/* Output: R terminates or does not terminate */
  begin function

```
        T_i^1 := {t | r_i ⊵ t and root(t) ∈ D_R} for every i;
        k := 1;
        while (∃i. T_i^k ≠ ∅) do
            if(∃i. →_{l_i → r_i} T_i^k ≠ ∅) then
              return "R does not terminate";
            else
              T_i^{k+1} := →_R T_i^k for every i;
              k := k + 1;
        end while
        return "R terminates";
    end function
```

The following theorem assures the correctness of the decision procedure SC_T.

Theorem 3.3 *Concerning the decision procedure SC_T, we have the following statements:*

(1) SC_T eventually terminates.

(2) SC_T outputs "R (does not) terminate" if and only if R (does not) terminate.

The proof of Theorem 3.3 will be found in the next section.

Corollary 3.4 *Termination of semi-constructor TRSs is decidable.*

4. Correctness of Decidable Procedure

At the beginning of this section, we first redefine loop, head-loop and cycle [4] and recall the definition of the dependency pair approach [7, 8]. Then we give some results concerning loop, head-loop and cycle which are related with the proof for correctness of SC_T. At the last part of this section, we give the detailed proof of Theorem 3.3.

Given a TRS R, a non-terminating term t *loops* in a rewrite sequence if the sequence contains $t \rightarrow_R^* t' \rightarrow_R^+ C[t'\theta]$ for some context C, substitution θ and term t'. A term t *head-loops* if containing $t \rightarrow_R^* t' \rightarrow_R^+ t'\theta$, and t *cycles* if containing $t \rightarrow_R^* t' \rightarrow_R^+ t'$. Cycling of t implies its head-looping, and head-looping of t implies its looping. We say a TRS admits a loop, head-loop and cycle, if there exists a term t such that t loops, head-loops and cycles, respectively. A TRS does not terminate if it admits one of these properties.

Example 4.1 Let a TRS $R_2 = \{f(x) \rightarrow h(f(g(a))),\ g(x) \rightarrow g(h(x))\}$ and a term $t = f(x)$. Then we have the following rewrite sequence

$$f(x) \rightarrow h(f(g(a))) \rightarrow h(h(f(g(a)))) \rightarrow \cdots$$

in which t loops of the form $t \rightarrow^+ C[t\theta]$ for a substitution $\theta = \{x \mapsto g(a)\}$ and context $C[\Box] = h(\Box)$.

Let R be a TRS over a signature $\mathcal{F}$. $\mathcal{F}^\sharp$ denotes the union of $\mathcal{F}$ and $D_R^\sharp = \{f^\sharp \mid f \in D_R\}$ where $\mathcal{F} \cap D_R^\sharp = \emptyset$ and $f^\sharp$ has the same arity as f. We call these new symbols *dependency pair symbols.* Given a term $t = f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ with f defined, we write $t^\sharp$ for the term $f^\sharp(t_1, \ldots, t_n)$. If $l \rightarrow r \in R$ and u is a subterm of r with a defined root symbol, then the rewrite rule $l^\sharp \rightarrow u^\sharp$ is called a *dependency pair* of R. The set of all dependency pairs of R is denoted by DP(R). For any subset $T \subseteqq \mathcal{T}(\mathcal{F}, \mathcal{V})$ consisting of terms with a defined root symbol, we denote the set $\{t^\sharp \mid t \in T\}$ by $T^\sharp$.

For a TRS R, a (possibly infinite) sequence of dependency pairs $s_1^\sharp \rightarrow t_1^\sharp$, $s_2^\sharp \rightarrow t_2^\sharp$, ... is a *dependency chain* if there exist substitutions $\tau_1, \tau_2, \ldots$ such that $t_i^\sharp \tau_i \rightarrow_R^* s_{i+1}^\sharp \tau_{i+1}$ holds for every two consecutive dependency pairs $s_i^\sharp \rightarrow t_i^\sharp$ and $s_{i+1}^\sharp \rightarrow t_{i+1}^\sharp$ in the sequence.

Proposition 4.2 *For a TRS R, if there exists an infinite dependency chain, then $R \cup DP(R)$ does not terminate over $\mathcal{T}^\sharp(\mathcal{F}, \mathcal{V})$.*

Example 4.3 For TRS R_2 in Example 4.1, $D_{R_2} = \{f, g\}$, DP(R_2) $= \{f^\sharp(x) \rightarrow f^\sharp(g(a)),\ f^\sharp(x) \rightarrow g^\sharp(a),\ g^\sharp(x) \rightarrow g^\sharp(h(x))\}$.

One of the benefits of the dependency pair approach is that we can follow the essential part in an infinite rewrite sequence as dependency chain.

Given a TRS, $\mathcal{T}_\infty$ denotes the set of all the *minimal non-terminating terms*, here "minimal" is in the sense that all its proper subterms terminate.

Definition 4.4 ($\mathcal{C}$-min) *For a TRS R, let $\mathcal{C} \subseteqq DP(R)$. An infinite rewrite sequence in $R \cup \mathcal{C}$ of the form*

$$t_1^\sharp \rightarrow_R^* t_2^\sharp \rightarrow_{\mathcal{C}} t_3^\sharp \rightarrow_R^* t_4^\sharp \rightarrow_{\mathcal{C}} \cdots$$

with $t_i \in \mathcal{T}_\infty$ for all $i \geqq 1$ is called $\mathcal{C}$-min. We use $\mathcal{C}_{min}(t^\sharp)$ to denote the set of all the $\mathcal{C}$-min rewrite sequences starting from $t^\sharp$.

Then we conclude some properties concerning $\mathcal{C}$-min in the following proposition.

Proposition 4.5 ([7, 8]) *Given a TRS R, a term $t \in \mathcal{T}_\infty$ and $\mathcal{C} \subseteqq DP(R)$, we have the following statements:*

(1) For every $t \in \mathcal{T}_\infty$, there exists a $\mathcal{C} \subseteqq DP(R)$ such that $\mathcal{C}_{min}(t^\sharp) \neq \emptyset$.

(2) For every sequence in $\mathcal{C}_{min}(t^\sharp)$, every reduction $\rightarrow_R$ takes place below the root while every reduction $\rightarrow_{\mathcal{C}}$ takes place at the root.

(3) For every sequence in $\mathcal{C}_{min}(t^\sharp)$, the rewrite subsequence $t_i^\sharp \rightarrow_R^* t_{i+1}^\sharp$ has a finite length.

(4) $s^\sharp \rightarrow_{R \cup \mathcal{C}}^* t^\sharp$ implies $s \rightarrow_R^* C[t]$ for some context C.

(5) There is a $\mathcal{C}$-min if and only if there is a corresponding dependency chain.

(6) For every sequence in $\mathcal{C}_{min}(t^\sharp)$, there is some rule $l^\sharp \rightarrow u^\sharp \in \mathcal{C}$ which is applied for infinite times.

Theorem 4.6 ([7, 8]) *For a TRS R, R does not terminate if and only if there exists an infinite dependency chain.*

The following lemma is a consequence of Theorem 4.6.

Lemma 4.7 *A TRS R does not terminate over $\mathcal{T}(\mathcal{F},\mathcal{V})$ if and only if the TRS $R \cup DP(R)$ does not terminate over $\mathcal{T}^\sharp(\mathcal{F},\mathcal{V})$.*

Proof. If R does not terminate, by Theorem 4.6, there exists an infinite dependency chain, thus by Proposition 4.2, $R\cup \mathrm{DP}(R)$ does not terminate over $\mathcal{T}^\sharp(\mathcal{F},\mathcal{V})$. If $R\cup\mathrm{DP}(R)$ does not terminate over $\mathcal{T}^\sharp(\mathcal{F},\mathcal{V})$, then there exists an infinite rewrite sequence starting from some term $t^\sharp \in \mathcal{T}^\sharp(\mathcal{F},\mathcal{V})$ of the form $t^\sharp \xrightarrow{>\varepsilon}{}^*_R t^\sharp_1 \xrightarrow{\varepsilon}_{\mathrm{DP}(R)} t^\sharp_2 \xrightarrow{>\varepsilon}{}^*_R \cdots$. In the sequence, if the root reduction $\rightarrow_{\mathrm{DP}(R)}$ is applied infinite times, then there exists an infinite dependency chain, thus by Theorem 4.6, R does not terminate, else there is an index k, the subsequence starting from $t^\sharp_k$ only applies reduction $\xrightarrow{>\varepsilon}_R$ for infinite times, thus term t_k does not terminate with respect to R, therefore R does not terminate. □

Next, we talk about the relation of looping between a TRS R and the corresponding TRS $R\cup\mathrm{DP}(R)$.

Lemma 4.8 *For a TRS R, a term $t \in \mathcal{T}_\infty$ and a sequence $sq \in \mathcal{C}_{min}(t^\sharp)$, if $t^\sharp$ loops in sq, then $t^\sharp$ head-loops in sq.*

Proof. Since $t^\sharp$ loops in sq with respect to $R \cup \mathcal{C}$ where $\mathcal{C}$ is a subset of $\mathrm{DP}(R)$, we can find a rewrite subsequence $t^\sharp \rightarrow^*_{R\cup\mathcal{C}} t^\sharp_k \rightarrow^+_{R\cup\mathcal{C}} C[t^\sharp_k\theta]$ in sq for some index k, context C and substitution θ. Since the dependency pair symbol only occurs at the root position, $C[t^\sharp_k\theta] = t^\sharp_k\theta$. Therefore, we have the rewrite subsequence $t^\sharp \rightarrow^*_{R\cup\mathcal{C}} t^\sharp_k \rightarrow^+_{R\cup\mathcal{C}} t^\sharp_k\theta$ in sq, which shows that $t^\sharp$ head-loops in sq. □

As the comment of Lemma 4.8, note that "$C[t^\sharp_k\theta] = t^\sharp_k\theta$" in the proof can not be substituted by "$C[t^\sharp_k\theta] = t^\sharp_k$", that is, $t^\sharp$ loops in a sequence $sq \in \mathcal{C}_{min}(t^\sharp)$ can not imply that $t^\sharp$ cycles in sq. To see this point, we have the following example.

Example 4.9 For the looping TRS R_2 in Example 4.1, $g(x)$ is a minimal non-terminating term of R_2. Let $t^\sharp = g^\sharp(x) \in \mathcal{T}^\sharp_\infty$, we can easily find an infinite rewrite sequence in $\mathcal{C}_{min}(g^\sharp(x))$ for $\mathcal{C} = \{g^\sharp(x) \rightarrow g^\sharp(h(x))\}$:

$$g^\sharp(x) \rightarrow g^\sharp(h(x)) \rightarrow g^\sharp(h(h(x))) \rightarrow \cdots$$

in which $g^\sharp(x)$ head-loops, but does not cycle.

Lemma 4.10 *For a TRS R, a term $t \in \mathcal{T}_\infty$ and a sequence $sq \in \mathcal{C}_{min}(t^\sharp)$, if $t^\sharp$ loops in sq, then there exists a term $t^\sharp_k$ in sq such that t_k loops with respect to R.*

Proof. By Lemma 4.8, $t^\sharp$ head-loops in sq with respect to $R \cup \mathcal{C}$ where $\mathcal{C}$ is a subset of $\mathrm{DP}(R)$. Thus we can find a rewrite subsequence $t^\sharp \rightarrow^*_{R\cup\mathcal{C}} t^\sharp_k \rightarrow^+_{R\cup\mathcal{C}} t^\sharp_k\theta$ in sq for some index k and substitution θ. From Proposition 4.5 – (4), there exists a context C' such that $t_k \rightarrow^+_R C'[t_k\theta]$, that is, t_k loops with respect to R. □

From the proof of Lemma 4.10, if it is the case that $t^\sharp$ cycles of the form $t^\sharp \rightarrow^*_{R\cup\mathrm{DP}(R)} \cdot \rightarrow_{l^\sharp\rightarrow u^\sharp} t^\sharp$ then obviously we have the following corollary.

Corollary 4.11 *For a TRS R, a term $t \in \mathcal{T}_\infty$ and a sequence $sq \in \mathcal{C}_{min}(t^\sharp)$, if $t^\sharp$ cycles of the form $t^\sharp \rightarrow^*_{R\cup DP(R)} \cdot \rightarrow_{l^\sharp\rightarrow u^\sharp} t^\sharp$ for some $l^\sharp \rightarrow u^\sharp \in \mathcal{C} \subseteqq DP(R)$ in sq, then t loops of the form $t \rightarrow^*_R \cdot \rightarrow_{\{l\rightarrow r\}} C'[t]$ for some context C' and $l \rightarrow r \in R$ such that $r \trianglerighteq u$.*

Conversely, it is not true for the proposition that for a TRS R, if there exists a term $t \in \mathcal{T}_\infty$ such that t loops with respect to R, then $t^\sharp$ loops in some sequence $sq \in \mathcal{C}_{min}(t^\sharp)$ with respect to $R\cup\mathrm{DP}(R)$, as shown in the following example.

Example 4.12 From Example 4.1, the term $t = f(x) \in \mathcal{T}_\infty$ loops with respect to R_2. However, we have that the following sequence is the only one in $\mathcal{C}_{min}(f^\sharp(x))$ for $\mathcal{C} = \mathrm{DP}(R_2)$:

$$f^\sharp(x) \rightarrow g^\sharp(a) \rightarrow g^\sharp(h(a)) \rightarrow g^\sharp(h(h(a))) \rightarrow \cdots$$

in which $f^\sharp(x)$ does not loop with respect to $R\cup\mathrm{DP}(R)$.

There are many unknown properties concerning loop, head-loop and cycle. For example, observe that without the condition "in some sequence $sq \in \mathcal{C}_{min}(t^\sharp)$", then in the example above, $f^\sharp(x)$ loops of the form

$$f^\sharp(x) \rightarrow f^\sharp(g(a)) \rightarrow f^\sharp(g(a))) \rightarrow \cdots$$

with respect to $R\cup\mathrm{DP}(R)$. Thus, we have the following conjecture.

Conjecture 4.13 *For a TRS R, if there exists a term $t \in \mathcal{T}_\infty$ such that t loops with respect to R, then $t^\sharp$ loops with respect to $R \cup DP(R)$.*

Next, we first give some lemmas on semi-constructor TRSs and then prove Theorem 3.3 by using these lemmas.

The following fact holds obviously.

Proposition 4.14 *For a semi-constructor TRS R, $DP(R)$ is right-ground.*

Lemma 4.15 *For a semi-constructor TRS R, the following statements are equivalent:*

(1) *R does not terminate.*

(2) *There exists a pair $l^\sharp \rightarrow u^\sharp \in DP(R)$ such that $u \in \mathcal{T}_\infty$ and $u^\sharp$ cycles of the form $u^\sharp \rightarrow^*_{R\cup DP(R)} \cdot \rightarrow_{\{l^\sharp\rightarrow u^\sharp\}} u^\sharp$ in some sequence $sq \in \mathcal{C}_{min}(u^\sharp)$ for some $\mathcal{C} \subseteqq DP(R)$.*

Proof.

(2)⇒(1): It is obvious by Corollary 4.11.

(1)⇒(2): Since R does not terminate, there exists a $t \in \mathcal{T}_\infty$, by (1) of Proposition 4.5, there exists a corresponding sequence $sq \in \mathcal{C}_{min}(t^\sharp)$ for some $\mathcal{C} \subseteqq \mathrm{DP}(R)$. Thus by Proposition 4.5 – (2) and (6), there exists a pair $l^\sharp \rightarrow u^\sharp \in \mathrm{DP}(R)$

such that it is applied for root reduction in sq for infinite times. Since $u^\sharp$ is ground, by Proposition 4.14 we have $u \in \mathcal{T}_\infty$ and $u^\sharp$ cycles of the form $u^\sharp \to^*_{R\cup \mathrm{DP}(R)} \cdot \to_{\{l^\sharp \to u^\sharp\}} u^\sharp$ in sq. □

We give the proof of Theorem 3.3 as follows.

Proof. **(Theorem 3.3)**
It is easily known that $s \in T_i^k$ if and only if $\exists t.r_i \trianglerighteq t$ and $root(t) \in D_R$ and $t \to_R^{k-1} s$.
Firstly, consider the case that R terminates. Then the condition in the 9th line will never hold, otherwise we have $t \to_R^{k-1} s \to_{\{l_i \to r_i\}} t'$ and $t' \trianglerighteq t$ for some $t \in T_i^1$, which leads to a contradiction. Also $T_1^k, \ldots, T_n^k$ will be empty for some k, since R terminates. Hence, the execution of the while loop that begins from the 8th line will end and SC_T terminates with the output "R terminates".
Secondly, consider the case that R does not terminate. Then the condition in the 8th line always holds. From Lemma 4.15 and Corollary 4.11, we have $u \to_R^* \cdot \to_{\{l_i \to r_i\}} C'[u]$ for some index i, term u and context C' such that $r_i \trianglerighteq u$ and $root(u) \in D_R$. Thus, $\to_{\{l_i \to r_i\}} T_i^k$ is not empty for some k and SC_T produces the output "R does not terminate".
The *only if* part of (2) is trivial. □

5. Comparison

In this section, we briefly compare our result with the existing related results, then we give a result concerning the dependency graph of semi-constructor TRSs and compare it with the result in [1] for right-ground case.

We have shown the decidability of termination for semi-constructor TRSs by using the notion of dependency pair. Note that the class of semi-constructor TRSs is a superclass of right-ground case.

Next, we talk about the dependency graph of semi-constructor TRSs.

The nodes of the *dependency graph* DG(R) are the dependency pairs of R and there is an arrow from a pair $s^\sharp \to t^\sharp$ to $u^\sharp \to v^\sharp$ if and only if there exist substitutions σ and τ such that $t^\sharp\sigma \to_R^* u^\sharp\tau$. We write SCS for a strongly connected subgraph of DG(R)[1]. Note that the dependency graph is not computable in general.

For a semi-constructor TRS R, we can give the following lemma.

Lemma 5.1 *For a semi-constructor TRS R, the following statements are equivalent:*

(1) R does not terminate.

(2) DG(R) contains at least one SCS.

Proof. If R does not terminate, then by Lemma 4.15, there exists a pair $l^\sharp \to u^\sharp \in \mathrm{DP}(R)$ such that $u \in \mathcal{T}_\infty$ and $u^\sharp$ cycles of the form $u^\sharp \to^+_{R\cup \mathrm{DP}(R)} u^\sharp$, thus there is a corresponding SCS in DG(R). If there is a SCS in DG(R), then from Proposition 4.14 we have the following sequence:
$l_1^\sharp\tau_1 \to_{\{l_1^\sharp \to u_1^\sharp\}} u_1^\sharp\sigma_1 = u_1^\sharp \to_R^* l_2^\sharp\tau_2 \to_{\{l_2^\sharp \to u_2^\sharp\}} u_2^\sharp\sigma_2 = u_2^\sharp$
$\to_R^* \cdots l_k^\sharp\tau_k \to_{\{l_k^\sharp \to u_k^\sharp\}} u_k^\sharp\sigma_k = u_k^\sharp \to_R^* l_1^\sharp\tau_{k+1} \to_{\{l_1^\sharp \to u_1^\sharp\}} u_1^\sharp\sigma_{k+1} =$
$u_1^\sharp$
for substitutions τ_i, σ_i, $l_i^\sharp \to u_i^\sharp \in$ SCS and $1 \leqq i \leqq k$. Thus $u_1^\sharp$ does not terminate with respect to $R \cup \mathrm{DP}(R)$, thus by Lemma 4.7, R does not terminate. □

In [1], A. Middeldorp proposed an easy decision procedure for termination of right-ground TRSs which is dependency graph based. Denoting growing approximation dependency graph by DG$_g$(R), He showed that for every right-hand TRS R, DG(R) = DG$_g$(R), that is, the precise dependency graph of the right-ground TRS is computable. Thus, the decision procedure proposed is that: compute the dependency graph of R using the growing approximation and then determine whether there are any SCSs. For semi-constructor case, we also have Lemma 5.1 to assure that semi-constructor TRS terminates if and only if there is no SCSs in the dependency graph. However, the method above can not be directly applied to semi-constructor case so far, since normally we only have DG(R)$\subseteqq$DG$_g$(R) for semi-constructor TRS R. It is still unknown whether the dependency graph of the semi-constructor TRS is computable or not.

6. Conclusion

In this paper we have shown that termination is a decidable property for semi-constructor TRSs and hence we can automatically prove termination or non-terminating of any semi-constructor TRSs by using procedure SC_T.

Based on the notion of dependency pair, we accomplished our proof by showing the decidability of the absence of the infinite dependency chains for semi-constructor TRSs which implies the termination of the TRSs.

We have compared our result with the existing results in Section 5. The comparison with the decision procedure proposed in the reference [1] showed that the dependency graph based method is not directly applicable to the semi-constructor case. It is still unknown whether the dependency graph of the semi-constructor TRS is computable or not. Furthermore, It is also worthwhile to do further investigations on the classes of TRSs, the termination of which are equivalent to the absence of SCSs in dependency graph and the dependency graph of which are computable but the equivalence of termination and absence of SCSs do not hold.

Acknowledgment This work is partly supported by

[1] A graph is said to be strongly connected if each node is reachable from an arbitrary node.

MEXT. KAKENHI #15500007 and #16300005.

References

[1] Aart Middeldorp. Approximating Dependency Graphs using Tree Automata Techniques, Proceedings of the International Joint Conference on Automated Reasoning(IJCAR'01), volume 2083 of LNAI, pages 593–610, 2001.

[2] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*, Cambridge University Press, 1998.

[3] Gérard Huet and Dallas Lankford. On the uniform halting problem for term rewriting systems, Technical Report 283, IRIA, 1978.

[4] Hans Zantema and Alfons Geser. Non-looping Rewriting, Technical Report, UU-CS-1996-03, Utrecht University, January 1996.

[5] Ichiro Mitsuhashi, Michio Oyamaguchi, Yoshikatsu Ohta and Toshiyuki Yamada. The Joinability and Unification Problems for Confluent Semi-Constructor TRSs, Proceedings of the 15th International Conference on Rewriting Techniques and Applications, volume 3091 of LNCS, pages 285–300, 2004.

[6] Nachum Dershowitz. Termination of linear rewriting systems, in S. Even and O. Kariv, editors, Automata, Languages and Programming, volume 115 of LNCS, pages 448–458, Springer-Verlag, 1981.

[7] Nao Hirokawa and Aart Middeldorp. Dependency Pairs Revisited, Proceedings of the 15th International Conference on Rewriting Techniques and Applications, Aachen, volume 3091 of LNCS, pages 249–268, 2004.

[8] Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs, *Theoretical Computer Science*, pages 133–178, 2000.