数理解析研究所講究録
第 1554 巻 2007 年 171-177

Confluence of Length Preserving String Rewriting Systems is Undecidable

Yi Wang, Masahiko Sakai, Naoki Nishida,
Toshiki Sakabe, Keiichirou Kusakari

Graduate School of Information Science, Nagoya University
{wangyi@trs.cm., sakai@, nishida@, sakabe@, kusakari@}is.nagoya-u.ac.jp

Abstract

This paper shows the undecidability of confluence for length preserving string rewriting systems. It is proven by reducing the Post's correspondence problem (PCP), which is known to be undecidable, to confluence problem for length preserving string rewriting systems. More precisely, we designed a reduction algorithm having the property that the existence of a solution for a given instance of PCP coincides with the non-confluence of the string rewriting system obtained from the reduction algorithm.

Keywords Post's Correspondence Problem

1 Introduction

String rewriting systems (SRSs) are said to be length preserving if the left-hand side and the right-hand side of each rule have the same length. Caron showed that its termination is an undecidable property[1]. This paper shows its confluence is also an undecidable property, although both of the reachability problem and confluence of given strings are easily known to be decidable.

Confluence is generally undecidable for term rewriting systems (TRSs) and for string rewriting systems. Hence several decidable classes on confluence have been studied: terminating TRSs[6], ground TRSs[9], linear shallow TRSs[3], shallow right-linear TRSs[4]. There are also results on undecidable classes on confluence: semi-constructor TRSs[7] and flat TRSs[5, 8].

In this paper, we show the undecidability of confluence for length preserving SRSs and prove it by reducing the Post's correspondence problem (PCP), which is known to be undecidable, to confluence problem for length preserving string rewriting systems. More precisely, we designed a reduction algorithm having the property that the existence of a solution for a given instance of PCP coincides with the non-confluence of the SRS obtained from the reduction algorithm.

2 Preliminaries

Let Σ be an alphabet. A *string rewrite rule* is a pair of strings l, $r \in \Sigma^*$, denoted by $l \to r$. A finite set of string rewrite rules is called a *string rewriting system* (SRS). An SRS $\mathcal{R}$ induces a *rewrite step* relation $\to_{\mathcal{R}}$ defined by $s \to_{\mathcal{R}} t$ if there are $u, v \in \Sigma^*$ and a rule $l \to r$ in $\mathcal{R}$ such that $s = ulv$ and $t = urv$. We use $\xrightarrow[\mathcal{R}]{+}$ for the transitive closure of $\to_{\mathcal{R}}$ and $\xrightarrow[\mathcal{R}]{*}$ for the reflexive-transitive closure of $\to_{\mathcal{R}}$. We say that strings s_1 and s_2 are *joinable* if $s_1 \xrightarrow[\mathcal{R}]{*} s \xleftarrow[\mathcal{R}]{*} s_2$ for some s, denoted by $s_1 \downarrow_{\mathcal{R}} s_2$. A string s is *confluent* if $s_1 \downarrow_{\mathcal{R}} s_2$ for any $s_1 \xleftarrow[\mathcal{R}]{*} s \xrightarrow[\mathcal{R}]{*} s_2$. An SRS $\mathcal{R}$ is *confluent* if all strings are confluent.

In this paper, the notation $|u|$ represents the length of string u. The notation $\underbrace{a \cdots a}_{m}$ denotes the string that consists of m symbols of a. We refer $\{r \to l \mid l \to r \in \mathcal{R}\}$ by $\mathcal{R}^{-1}$.

3 Length preserving SRSs and undecidability of their confluence

Definition 3.1 *An SRS $\mathcal{R}$ is said to be length preserving if $|l| = |r|$ for every rule $l \to r$ in $\mathcal{R}$.*

Since rules are finite, symbols appearing in rules are also finite. Hence strings composed of n such symols are finite. Thus the decidability of the following problems for length preserving SRSs are trivially follows:

1. *Reachability problem* is a problem to decide $s \xrightarrow[\mathcal{R}]{*} t$ for given strings s and t and a SRS $\mathcal{R}$.
2. *String-confluence problem* is a problem to decide confluence of s for a given string s and a SRS $\mathcal{R}$.

Now we recall Post's correspondence problem, which is known to be undecidable.

Definition 3.2 *An instance of PCP is a set $P \subseteq \mathcal{A}^* \times \mathcal{A}^*$ of finite pairs of strings over an alphabet $\mathcal{A}$ with at least two symbols. A solution of P is a string w such that*

$$w = u_1 \cdots u_k = v_1 \cdots v_k$$

for some $(u_i, v_i) \in P$. The Post's correspondence problem (PCP) is a problem to decide whether such a solution exists or not.

Example 3.3 $P = \{(aba, a), (aab, abab), (bb, babba)\}$ is an instance of PCP. P has a solution $ababbaababa$ with $(u_1, v_1) = (aba, a)$, $(u_2, v_2) = (bb, babba)$, $(u_3, v_3) = (aab, abab)$ and $(u_4, v_4) = (aba, a)$.

Theorem 3.4 ([10]) *PCP is undecidable.*

As a preparation of the algorithm that transform an instance of PCP to an SRS, we introduce a kind of null symbol $-$ and an equal length representation of each pair in instances of PCP. Let $P = (u_1, v_1), \ldots, (u_n, v_n)$ be an instance of PCP over $\mathcal{A}$.

$$\begin{aligned} \overline{P} = \ & \{(u, v\underbrace{- \cdots -}_{m}) \mid (u, v) \in P \text{ and } |u| - |v| = m \geq 0\} \\ & \cup \{(u\underbrace{- \cdots -}_{m}, v) \mid (u, v) \in P \text{ and } |u| - |v| = m < 0\} \end{aligned}$$

We use $\overline{\mathcal{A}}$ for $\mathcal{A} \cup \{-\}$.

Example 3.5 For instance $P = \{(ab, a), (a, ba)\}$ of PCP over $\{a, b\}$, we have

$$\overline{P} = \{(ab, a-), (a-, ba)\}$$

We use symbols like $\overset{a}{\underset{b'}{X^{b}_{a'}}}$. For an easy handling of strings that consist of such symbols, we introduce a notation defined as follows:

$$(X_1\cdots X_n)^{\overset{a_1\cdots a_n}{b_1\cdots b_n}}_{\underset{b'_1\cdots b'_n}{a'_1\cdots a'_n}} = \overset{a_1}{\underset{b'_1}{X_1{}^{b_1}_{a'_1}}}\cdots\overset{a_n}{\underset{b'_n}{X_n{}^{b_n}_{a'_n}}}$$

For example $(eee)^{\overset{a_1a_2a_3}{b_1b_2b_3}}_{\underset{d_1d_2d_3}{c_1c_2c_3}}$ denotes $\overset{a_1}{\underset{d_1}{e^{b_1}_{c_1}}}\overset{a_2}{\underset{d_2}{e^{b_2}_{c_2}}}\overset{a_3}{\underset{d_3}{e^{b_3}_{c_3}}}$. Note that the length of the strings in the superscripts and subscripts are the same when we use this notation. We call X the *tag* of $\overset{a}{\underset{b'}{X^{b}_{a'}}}$.

Definition 3.6 *Let P be an instance of PCP over $\mathcal{A}$. The SRS $\mathcal{R}_P$ over Σ obtained from P is defined as follows:*

$$\begin{array}{rcl}
\Sigma &=& \{\Xi_0,\Xi_1,\Xi_2,\Psi_0,\Psi_1,\Psi_2\}\cup\Sigma_c\\
\Sigma_c &=& \left\{\overset{x_1}{\underset{x_4}{c^{x_2}_{x_3}}},\overset{x_1}{\underset{x_4}{\tilde{c}^{x_2}_{x_3}}},\overset{x_1}{\underset{x_4}{p^{x_2}_{x_3}}},\overset{x_1}{\underset{x_4}{\tilde{p}^{x_2}_{x_3}}},\overset{x_1}{\underset{x_4}{e^{x_2}_{x_3}}},\overset{x_1}{\underset{x_4}{\tilde{e}^{x_2}_{x_3}}} \;\middle|\; x_i\in\overline{\mathcal{A}}\right\}
\end{array}$$

$$\begin{array}{rcl}
\mathcal{R}_P &=& \Theta\cup\Theta^{-1}\cup\Phi\\
\Theta &=& \delta\cup\alpha_1\cup\beta_1\cup\alpha_2\cup\beta_2\\
\Phi &=& \gamma_1\cup\gamma_2\\
\delta &=& \left\{\overset{x_1}{\underset{x_4}{c^{x_2}_{-}}}\overset{y_1}{\underset{y_4}{c^{y_2}_{z}}}\to\overset{x_1}{\underset{x_4}{c^{x_2}_{z}}}\overset{y_1}{\underset{y_4}{c^{y_2}_{-}}},\ \overset{x_1}{\underset{-}{c^{x_2}_{x_3}}}\overset{y_1}{\underset{z}{c^{y_2}_{y_3}}}\to\overset{x_1}{\underset{z}{c^{x_2}_{x_3}}}\overset{y_1}{\underset{-}{c^{y_2}_{y_3}}} \;\middle|\; x_j,y_j\in\overline{\mathcal{A}},z\in\mathcal{A}\right\}\\
\alpha_1 &=& \left\{(cc\cdots c)^{\overset{u}{v}}_{\underset{v}{u}}\Psi_0\to(p\tilde{c}\cdots\tilde{c})^{\overset{u}{v}}_{\underset{v}{u}}\Psi_1 \;\middle|\; (u,v)\in\overline{P}\right\}\\
\beta_1 &=& \left\{(cc\cdots c)^{\overset{u}{v}}_{\underset{v}{u}}\overset{x_1}{\underset{x_2}{p^{x_2}_{x_1}}}\to(p\tilde{c}\cdots\tilde{c})^{\overset{u}{v}}_{\underset{v}{u}}\overset{x_1}{\underset{x_2}{\tilde{p}^{x_2}_{x_1}}} \;\middle|\; (u,v)\in\overline{P},x_j\in\overline{\mathcal{A}}\right\}\\
\gamma_1 &=& \left\{\Xi_0\overset{x_1}{\underset{x_2}{p^{x_2}_{x_1}}}\to\Xi_1\overset{x_1}{\underset{x_2}{\tilde{p}^{x_2}_{x_1}}} \;\middle|\; x_j\in\overline{\mathcal{A}}\right\}\\
\alpha_2 &=& \left\{\overset{x_1}{\underset{x_3}{c^{x_2}_{x_3}}}\Psi_0\to\overset{x_1}{\underset{x_3}{e^{x_2}_{x_3}}}\Psi_2 \;\middle|\; x_j,y_j\in\overline{\mathcal{A}}\right\}\\
\beta_2 &=& \left\{\overset{x_1}{\underset{x_3}{c^{x_2}_{x_3}}}\overset{y_1}{\underset{y_3}{e^{y_2}_{y_3}}}\to\overset{x_1}{\underset{x_3}{e^{x_2}_{x_3}}}\overset{y_1}{\underset{y_3}{\tilde{e}^{y_2}_{y_3}}} \;\middle|\; x_j,y_j\in\overline{\mathcal{A}}\right\}\\
\gamma_2 &=& \left\{\Xi_0\overset{x_1}{\underset{x_3}{e^{x_2}_{x_3}}}\to\Xi_2\overset{x_1}{\underset{x_3}{\tilde{e}^{x_2}_{x_3}}} \;\middle|\; x_j\in\overline{\mathcal{A}}\right\}
\end{array}$$

Example 3.7 Let $P=\{(a,ba),(ab,a)\}$ be an instance of PCP, where P has a solution aba. In $\mathcal{R}_P$, rules in α_1 and β_1 depends on P and the other rules depend only on the signature $\mathcal{A}$.

$$\begin{array}{rcl}
\alpha_1 &=& \left\{\overset{a}{\underset{b}{c^{b}_{a}}}\overset{-}{\underset{a}{c^{a}_{-}}}\Psi_0\to\overset{a}{\underset{b}{p^{b}_{a}}}\overset{-}{\underset{a}{\tilde{c}^{a}_{-}}}\Psi_1,\ \overset{a}{\underset{a}{c^{a}_{a}}}\overset{b}{\underset{-}{c^{-}_{b}}}\Psi_0\to\overset{a}{\underset{a}{p^{a}_{a}}}\overset{b}{\underset{-}{\tilde{c}^{-}_{b}}}\Psi_1\right\}\\
\beta_1 &=& \left\{\overset{a}{\underset{b}{c^{b}_{a}}}\overset{-}{\underset{a}{c^{a}_{-}}}\overset{x_1}{\underset{x_2}{p^{x_2}_{x_1}}}\to\overset{a}{\underset{b}{p^{b}_{a}}}\overset{-}{\underset{a}{\tilde{c}^{a}_{-}}}\overset{x_1}{\underset{x_2}{\tilde{p}^{x_2}_{x_1}}},\ \overset{a}{\underset{a}{c^{a}_{a}}}\overset{b}{\underset{-}{c^{-}_{b}}}\overset{x_1}{\underset{x_2}{p^{x_2}_{x_1}}}\to\overset{a}{\underset{a}{p^{a}_{a}}}\overset{b}{\underset{-}{\tilde{c}^{-}_{b}}}\overset{x_1}{\underset{x_2}{\tilde{p}^{x_2}_{x_1}}} \;\middle|\; x_i\in\overline{\mathcal{A}}\right\}
\end{array}$$

$\mathcal{R}_P$ is not confluent since we have the following reduction sequences:

$$\Sigma_0\overset{a}{\underset{a}{c^{a}_{a}}}\overset{b}{\underset{-}{c^{-}_{b}}}\overset{a}{\underset{b}{c^{b}_{a}}}\overset{-}{\underset{a}{c^{a}_{-}}}\Psi_0\underset{\alpha_1}{\to}\Sigma_0\overset{a}{\underset{a}{c^{a}_{a}}}\overset{b}{\underset{-}{c^{-}_{b}}}\overset{a}{\underset{b}{p^{b}_{a}}}\overset{-}{\underset{a}{\tilde{c}^{a}_{-}}}\Psi_1\underset{\beta_1}{\to}\Sigma_0\overset{a}{\underset{a}{p^{a}_{a}}}\overset{b}{\underset{-}{\tilde{c}^{-}_{b}}}\overset{a}{\underset{b}{\tilde{p}^{b}_{a}}}\overset{-}{\underset{a}{\tilde{c}^{a}_{-}}}\Psi_1\underset{\gamma_1}{\to}\Sigma_1\overset{a}{\underset{a}{\tilde{p}^{a}_{a}}}\overset{b}{\underset{-}{\tilde{c}^{-}_{b}}}\overset{a}{\underset{b}{\tilde{p}^{b}_{a}}}\overset{-}{\underset{a}{\tilde{c}^{a}_{-}}}\Psi_1\ ,$$

$$\Sigma_0\overset{a}{\underset{a}{c^{a}_{a}}}\overset{b}{\underset{-}{c^{-}_{b}}}\overset{a}{\underset{b}{c^{b}_{a}}}\overset{-}{\underset{a}{c^{a}_{-}}}\Psi_0\underset{\delta}{\overset{*}{\to}}\Sigma_0\overset{a}{\underset{a}{c^{a}_{a}}}\overset{b}{\underset{b}{c^{-}_{b}}}\overset{a}{\underset{a}{c^{b}_{a}}}\overset{-}{\underset{-}{c^{a}_{-}}}\Psi_0\underset{\alpha_2}{\to}\Sigma_0\overset{a}{\underset{a}{c^{a}_{a}}}\overset{b}{\underset{b}{c^{-}_{b}}}\overset{a}{\underset{a}{c^{b}_{a}}}\overset{-}{\underset{-}{e^{a}_{-}}}\Psi_2\underset{\beta_2}{\overset{*}{\to}}\Sigma_0\overset{a}{\underset{a}{e^{a}_{a}}}\overset{b}{\underset{b}{\tilde{e}^{-}_{b}}}\overset{a}{\underset{a}{\tilde{e}^{b}_{a}}}\overset{-}{\underset{-}{\tilde{e}^{a}_{-}}}\Psi_2\underset{\gamma_2}{\to}\Sigma_2\overset{a}{\underset{a}{\tilde{e}^{a}_{a}}}\overset{b}{\underset{b}{\tilde{e}^{-}_{b}}}\overset{a}{\underset{a}{\tilde{e}^{b}_{a}}}\overset{-}{\underset{-}{\tilde{e}^{a}_{-}}}\Psi_2$$

and their last step by γ_i rules are one way.

Obviously $\mathcal{R}_P$ is length preserving. The proof of the following main lemma is found in the next section.

Lemma 3.8 *Let P be an instance of PCP. Then, P has a solution if and only if $\mathcal{R}_P$ is not confluent.*

Theorem 3.9 *Confluence of length preserving SRSs is undecidable.*

Proof. We assume that confluence of length preserving SRSs is decidable. Then it follows from Lemma 3.8 that PCP is decidable, which contradicts to Theorem 3.4. □

Strings can be regarded as terms over unary functions and a variable. For example, a string abc corresponds to a term $a(b(c(x)))$. Hence we can consider classes of TRSs that contains all length preserving SRSs. Then, the undecidability of confluence for such classes is a corollary of the above theorem.

Structure preserving TRSs are TRSs in which the left-hand side and right-hand side of each rule have the same tree structure and the same variable occurrences. The tree structures are stable against reductions in this class of TRSs. Hence we have the following corollary.

Corollary 3.10 *Confluence of structure preserving TRSs is undecidable.*

4 Proof of Lemma 3.8

We use notion of persistency to simplify proofs as done in [2].

Theorem 4.1 (persistence of confluence[11]) *A well-typed (many-sorted) term rewriting system is confluent if and only if its underlined (untyped) term rewriting system is confluent.*

Now we apply Theorem 4.1 to $\mathcal{R}_P$.

Lemma 4.2 *Let $\mathcal{R}_P$ be the SRS over $\Sigma = \{\Xi_0, \Xi_1, \Xi_2, \Psi_0, \Psi_1, \Psi_2\} \cup \Sigma_c$ obtained from an instance P of PCP. Then $\mathcal{R}_P$ is confluent if and only if w is confluent for every w in either of the following three forms:*

(p1) $\Xi_i\chi$

(p2) $\chi\Psi_j$,

(p3) $\Xi_i\chi\Psi_j$,

where $\chi \in (\Sigma_c)^*$, $i, j \in \{0, 1, 2\}$.

Proof. Considering following typing to R_P, the lemma follows from the persistency (Theorem 4.1).

$$\begin{array}{ll} \Psi_i : \mathcal{T}'' \to \mathcal{T} & \text{for each } i \in \{0,1,2\} \\ X : \mathcal{T} \to \mathcal{T} & \text{for each } X \in \Sigma_c \\ \Xi_i : \mathcal{T} \to \mathcal{T}' & \text{for each } i \in \{0,1,2\} \end{array}$$

□

In the sequel, we analyze the confluent property for $\mathcal{R}_P = \Theta \cup \Theta^{-1} \cup \Phi$ obtained from an instance P of PCP.

We define an equivalence relation $\sim \subseteq (\overline{\mathcal{A}})^* \times (\overline{\mathcal{A}})^*$ as identity relation with ignoring all null symbols $-$, that is $u \sim v$ if and only if $\tilde{u} = \tilde{v}$ where $\tilde{u}$ and $\tilde{v}$ denote the strings obtained from u and v by omitting all $-$s respectively.

For a string $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0$, rules in α_1 and β_1 are to check that $u = u'$ and $v = v'$ and that (u, v) consists of a list of pairs in $\overline{P}$. Rules in δ gather null symbols in subscripts u' and v' backward. From these observation, the following lemma holds.

Lemma 4.3 *Let $u, u', v, v' \in (\overline{\mathcal{A}})^*$. Then, $u' \sim u = u_1 \cdots u_k$ and $v' \sim v = v_1 \cdots v_k$ for some k and $(u_i, v_i) \in \overline{P}$ if and only if $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0 \xrightarrow[\mathcal{R}_P]{*} \Xi_0 p^{\substack{x_1\\x_2}}_{\substack{x_1\\x_2}}\chi\Psi_1$ for some $\chi \in (\Sigma_c)^*$.*

Proof. ($\Rightarrow$) We have a reduction sequence $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0 \xrightarrow[\delta\cup\delta^{-1}]{*} \Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u\\v}}\Psi_0$ since $u \sim u'$ and $v \sim v'$. As shown in Example 3.5, we have a reduction sequence $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u\\v}}\Psi_0 \xrightarrow[\alpha_1\cup\beta_1]{*} \Xi_0 p^{\substack{x_1\\x_2}}_{\substack{x_1\\x_2}}\chi\Psi_1$ for some $x_i \in \overline{\mathcal{A}}$ since $u = u_1 \cdots u_n$ and $v = v_1 \cdots v_n$ for some $(u_i, v_i) \in \overline{P}$.

($\Leftarrow$) Let $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0 \xrightarrow[\mathcal{R}_P]{*} \Xi_0 p^{\substack{x_1\\x_2}}_{\substack{x_1\\x_2}}\chi\Psi_1$. Then rules α_1 and β_1 must be used in the reduction and all tags of χ are p' or c'. Since u and v cannot be modified by any rule in $\mathcal{R}_P$ and any possible reductions have no harmful branches from the construction of $\mathcal{R}_P$, the string $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u\\v}}\Psi_0$ must appears in the reduction. Thus we have $u = u_1 \cdots u_k$ and $v = v_1 \cdots v_k$ for some k and $(u_i, v_i) \in \overline{P}$ from the construction of α_1 and β_1 rules. We also have $u \sim u'$ and $v \sim v'$ from the construction of δ rules. $\square$

For a string $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0$, rules in α_2 and β_2 are to check that $u' = v'$. From this observation, the following lemma holds.

Lemma 4.4 *Let $u, u', v, v' \in (\overline{\mathcal{A}})^*$ such that $|u| = |u'| = |v| = |v'|$. Then, $u' \sim v'$ if and only if $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0 \xrightarrow[\mathcal{R}_P]{*} \Xi_0 e^{\substack{x_1\\x_2}}_{\substack{x_3\\x_3}}\chi\Psi_2$ for some $\chi \in (\Sigma_c)^*$.*

Proof. ($\Rightarrow$) We have a reduction sequence $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0 \xrightarrow[\delta\cup\delta^{-1}]{*} \Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\u'}}\Psi_0$ since $u' \sim v'$. As shown in Example 3.5, we have a reduction sequence $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\u'}}\Psi_0 \xrightarrow[\alpha_2\cup\beta_2]{*} \Xi_0 e^{\substack{x_1\\x_2}}_{\substack{x_1\\x_2}}\chi\Psi_2$ for some $x_i \in \overline{\mathcal{A}}$.

($\Leftarrow$) Let $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u'\\v'}}\Psi_0 \xrightarrow[\mathcal{R}_P]{*} \Xi_0 e^{\substack{x_1\\x_2}}_{\substack{x_1\\x_2}}\chi\Psi_2$. Then rules α_2 and β_2 must be used and all tags of χ are e'. Since any possible reductions have no harmful branches from the construction of $\mathcal{R}_P$, the string $\Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{w\\w}}\Psi_0$ must appears in the reduction for some string w. Thus we have $u' \sim w$ and $v' \sim w$ from the construction of δ rules. $\square$

Lemma 4.5 *Let P be an instance of PCP.*

(a) If P has a solution, then $\Xi_0 p^{\substack{x_1\\x_2}}_{\substack{x_1\\x_2}}\chi\Psi_1 \xleftarrow[\mathcal{R}_P]{*} \Xi_0(c\cdots c)^{\substack{u\\v}}_{\substack{u\\v}}\Psi_0 \xrightarrow[\mathcal{R}_P]{*} \Xi_0 e^{\substack{x_1\\x_2}}_{\substack{x_3\\x_3}}\chi'\Psi_2$ for some $\chi, \chi' \in (\Sigma_c)^*$.

(b) If $\Xi_0 p^{\substack{x_1\\x_2}}_{\substack{x_1\\x_2}}\chi\Psi_1 \xleftrightarrow[\mathcal{R}_P]{*} \Xi_0 e^{\substack{x_1\\x_2}}_{\substack{x_3\\x_3}}\chi'\Psi_2$ for some $\chi, \chi' \in (\Sigma_c)^*$, then P has a solution.

Proof. (a) Let P has a solution. Then we have $u = u_1 \cdots u_k \sim v_1 \cdots v_k = v$ for some k and $(u_i, v_i) \in \overline{P}$. Hence the claim follows from Lemma 4.3, Lemma 4.4.

(b) Let $\Xi_0 \overset{x_1}{\underset{x_2}{p^{x_2}_{x_1}}} \chi \Psi_1 \xleftrightarrow[\mathcal{R}_P]{*} \Xi_0 \overset{x_1}{\underset{x_3}{e^{x_2}_{x_3}}} \chi' \Psi_2$. Then it is easy to see that a string $\Xi_0 \overset{u}{\underset{v'}{(c \cdots c)^{v}_{u'}}} \Psi_0$ must appear in this reduction. From Lemma 4.3 and Lemma 4.4, we have $u = u_1 \cdots u_k \sim v_1 \cdots v_k = v$ for some k and $(u_i, v_i) \in \overline{P}$, which means P has a solution $\tilde{u}$. □

The following proposition obviously follows since $(\mathcal{R}_P \backslash (\gamma_1 \cup \gamma_2))^{-1} \subseteq \mathcal{R}_P$.

Proposition 4.6 *Let $S_1, S_2 \in \Sigma^*$. Then,*

(1) $S_1 \xleftrightarrow[\mathcal{R}_P \backslash \gamma_1]{*} S_2$ implies $S_1 \downarrow_{\mathcal{R}_P} S_2$, and

(2) $S_1 \xleftrightarrow[\mathcal{R}_P \backslash \gamma_2]{*} S_2$ implies $S_1 \downarrow_{\mathcal{R}_P} S_2$.

Proof. Proof by induction on the number of reductions by γ rules. □

Proof for Lemma 3.8

($\Rightarrow$): Let P has a solution. Then we have $\Xi_0 \overset{x_1}{\underset{x_2}{p^{x_2}_{x_1}}} \chi \Psi_1 \xleftarrow[\mathcal{R}_P]{*} \Xi_0 \overset{u}{\underset{v'}{(c \cdots c)^{v}_{u'}}} \Psi_0 \xrightarrow[\mathcal{R}_P]{*} \Xi_0 \overset{x_1}{\underset{x_3}{e^{x_2}_{x_3}}} \chi' \Psi_2$ for some $\chi, \chi' \in (\Sigma_c)^*$ by Lemma 4.5(a). Hence we have $\Xi_1 \overset{x_1}{\underset{x_2}{\tilde{p}^{x_2}_{x_1}}} \chi \Psi_1 \xleftarrow[\mathcal{R}_P]{*} \Xi_0 \overset{u}{\underset{v'}{(c \cdots c)^{v}_{u'}}} \Psi_0 \xrightarrow[\mathcal{R}_P]{*} \Xi_2 \overset{x_1}{\underset{x_3}{\tilde{e}^{x_2}_{x_3}}} \chi' \Psi_2$ by using rules γ_1 and γ_2, which leads non-confluence of R_P.

($\Leftarrow$): Let P has no solution. Let's show that $\mathcal{R}_P$ is confluent. Let $S_1 \xleftarrow[\mathcal{R}_P]{*} S_0 \xrightarrow[\mathcal{R}_P]{*} S_2$. From Lemma 4.2, it is enough to consider three kind of forms (p1), (p2) and (p3) as S_0.

- Consider the case that S_0 starts with Ξ_0 and ends with Ψ_i for some $i \in \{0, 1, 2\}$. Assume that both of γ_1 and γ_2 are applied in the reduction sequence. Then P must have a solution by Lemma 4.5(b), which is a contradiction. Hence at least one of γ_1 or γ_2 rules cannot be applied in the reduction sequence.

- In either of the other cases:

  - The case that S_0 ends with Ψ_i for some $i \in \{0, 1, 2\}$ and all other symbols are of Σ_c,
  - The case that S_0 starts with Ξ_1 of Ξ_2, and
  - The case that S_0 starts with Ξ_0 and all other symbols are of Σ_c,

  It is easy to see that at least one of γ_1 or γ_2 rules cannot be applied in the reduction sequence.

In any of the above cases, we have $S_1 \downarrow_{R_P} S_2$ by Proposition 4.6. □

References

[1] A.-C. Caron. *Linear Bounded Automata and Rewrite Systems: Influence of Initial Configuration on Decision Properties*, Proc. of the Colloquium on Trees in Algebra and Programming(CAAP 91), LNCS, *493*, pp.74–89, 1991.

[2] A. Geser, A. Middeldorp, E. Ohlebusch, H. Zantema. *Relative Undecidability in Term Rewriting (Part 2: The confluence Hierarchy)*, Information and Computation, *178(1)*, pp.132–148, 2002.

[3] G. Godoy, A. Tiwari, R. Verma. *On the Confluence of Linear Shallow Term Rewriting Systems*, Proc. of 20th Intl. Symposium on Theoretical Aspects of Computer Science (STACS 2003), Lecture Notes in Computer Science, 2507, pp.85–96, 2003.

[4] G. Godoy, A. Tiwari. *Confluence of Shallow Right-Linear Rewrite Systems*, Proc. of 14th Annual Conf. on Computer Science Logic (CSL 2005), LNCS, *3634*, pp.541–556, 2005.

[5] F. Jacquemard. *Reachability and Confluence are Undecidable for Flat Term Rewriting Systems*, Information Processing Letters *87(5)*, pp.265–270, 2003.

[6] K. E. Knuth, P. B. Bendix. *Computational Problems in Abstract Algebra*, Pergamon Press, Oxford, pp.263–297, 1970.

[7] I. Mitsuhashi, M. Oyamguchi, Y. Ohta, and T. Yamada. *The Joinability and Related Decision Problems for Confluent Semi-Constructor TRSs*, Transactions of Information Processing Society of Japan, *47(5)*, pp.1502-1514, 2006.

[8] I. Mitsuhashi, M. Oyamaguchi and F. Jacquemard. *The Confluence Problem for Flat TRSs*, Proc. of 8th Intl. Conf. on Artificial Intelligence and Symbolic Computation (AISC'06), LNAI *4120*, pp.68-81, 2006.

[9] M. Oyamaguchi, *The Church-Rosser Property for ground term rewriting systems is Decidable*, Theoretical Computer Science, *49*, pp.43–79, 1987.

[10] E. Post. *A Variant of a Recursively Unsolvable Problem*. Bulletin of the American Mathematical Society, *52*, pp.264–268, 1946.

[11] H. Zantema, *Termination of Term Rewriting: Interpretation and type elimination*, Journal of Symbolic Computation, *17*, pp.23–50, 1994.