Transformation for Refining Unraveled Conditional Term Rewriting Systems

Naoki Nishida, Tomohiro Mizutani, and Masahiko Sakai

Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
{nishida@, mizutani@sakabe.i., sakai@}is.nagoya-u.ac.jp

Abstract. Unravelings, which transform conditional term rewriting systems (CTRSs) into unconditional term rewriting systems, are useful for analyzing properties of CTRSs. To compute reduction sequences of CTRSs, the restriction by a particular context-sensitive and membership condition is imposed on reductions of the unraveled CTRSs. The condition is determined by extra function symbols introduced due to the unravelings. In this paper, we propose a method to weaken the restriction, that is, to reduce the number of extra symbols. We first improve the unraveling for deterministic CTRSs, and then propose a transformation that folds two successively used rewrite rules in the unraveled CTRSs, which satisfy a condition, to a rewrite rule that simulates reductions by the two rules.

1 Introduction

Unravelings are transformations from conditional term rewriting systems (for short, CTRSs) into unconditional term rewriting systems (TRSs). They are useful for analyzing properties of CTRSs. For example, 'effective termination', in which CTRSs are terminating and the recursive reduction of the instantiated conditional parts also terminates, is an important property of CTRSs and it can be guaranteed by termination of the unraveled CTRSs [6, 11]. An unraveling for normal CTRSs was investigated by Bergstra and Klop [3]. This concept was revisited by Marchiori who discussed its properties such as syntactic ones, termination, modularity, and so on [6]. He also proposed the unraveling for join CTRSs. Ohlebusch proposed an unraveling for deterministic 3-CTRSs to prove termination of logic programs [10]. A variant of Ohlebusch's unraveling is used in several papers [4, 7–9].

It is well-known that reductions of CTRSs are much more complicated than those of TRSs. One of the reasons is that the recursive reduction is necessary to evaluate instantiated conditional parts. To compute reduction sequences of CTRSs, unravelings appear attractive. An unraveling is said to be *simulation-complete* for a CTRS over a signature if both *reachability* and *unreachability* of terms over the signature are preserved by the unraveling [7–9]. In general, unravelings are not simulation-complete for arbitrary target CTRSs because the

$$\rho : l \to r \Leftarrow s_1 \to t_1 \wedge \cdots \wedge s_k \to t_k$$
$$\Downarrow \mathbb{U}$$
$$\{\ l \to \mathtt{u}_1^\rho(s_1, \overrightarrow{x_1}),\ \mathtt{u}_1^\rho(t_1, \overrightarrow{x_1}) \to \mathtt{u}_2^\rho(s_2, \overrightarrow{x_2}),\ \cdots,\ \mathtt{u}_k^\rho(t_k, \overrightarrow{x_k}) \to r\ \}$$

Fig. 1. Outline of the unraveling for deterministic CTRSs.

unraveled CTRSs are simple approximations of the original CTRSs [6, 11]. However, it was shown that the restriction by a particular *context-sensitive* and *membership* condition to reductions of the unraveled CTRSs preserves unreachability of the original CTRSs, that is, simulation-completeness of the unravelings [8].

Unravelings are generally done by decomposing each conditional rule to some unconditional rules that are supposed to be used in a fixed order (see Fig. 1). A reduction from $l\sigma$ to $r\sigma$ by the conditional rule ρ is simulated by a reduction sequence by the corresponding unconditional rules; the sequence starts from $l\sigma$; in the sequence, each extra function symbol $\mathtt{u}_i^\rho$ (called a *U symbol*) not in the original signature checks sequentially reachability from $s_i\sigma$ to $t_i\sigma$ (evaluates the condition $s_i \to t_i$ with σ); the sequence ends at $r\sigma$ after all conditions are evaluated successfully. We are sure that the unravelings preserve reachability on terms over the original signatures. On the other hand, the unravelings do not preserve unreachability for all CTRSs because unexpected reduction sequences are sometimes caused by disobeying the application order of rules whose left-hand sides are rooted with the U symbols [6, 11]. To avoid this, a restriction to reductions of the unraveled CTRSs is required, which prohibits reductions associated with the following redexes:

– (context-sensitive condition) redexes that occur strictly below U symbols, except for the first arguments of the U symbols , or
– (membership condition) redexes that contains a U symbol in their proper subterms.

In this way, the restriction by the above context-sensitive and membership condition is imposed on reductions of the unraveled CTRSs to maintain simulation-completeness [8]. As another approach to simulation-completeness, it was shown that the unraveled CTRSs are simulation-complete for the original CTRSs if the unraveled ones are either left-linear or both right-linear and non-erasing [7].

In this paper, we try to construct unconditional TRSs that are simulation-complete for the original CTRSs without the context-sensitive and membership condition. We first improve the unraveling for deterministic CTRSs so that the number of unraveled rules is less than those with the ordinary unraveling. We then propose a transformation, which is applied to the unraveled CTRSs, to remove the U symbols as many as possible from the unraveled CTRSs. The transformation folds two rules used successively in reduction sequences into one rule (see Fig. 2). We show a delicate condition that U symbols to be removed should satisfy, and we tighten it to maintain an advantage of CTRSs associated with the 'let' structure of functional programs. Removing U symbols leads to the relaxation of the restriction by the context-sensitive and membership condition

$$\left(\begin{array}{l} S \cup \{\; l_1 \to \mathtt{u}_i^{\rho}(t_{i,1}\delta, \ldots, t_{i,m_i}\delta, \overrightarrow{x_i}), \\ \qquad\qquad\quad \mathtt{u}_i^{\rho}(t_{i,1}, \ldots, t_{i,m_i}, \overrightarrow{x_i}) \quad \to r_2 \;\} \end{array}, \mu\right) \quad \Longrightarrow^{\mathbb{T}} \quad (S \cup \{\; l_1 \to r_2\delta \;\}, \mu')$$
where μ is updated to μ' w.r.t. $\mathtt{root}(r_2)$

Fig. 2. Outline for removing U symbols by the transformation $\mathbb{T}$.

because the condition depends on the existence of U symbols. We also show correctness of the transformation, and show that the composition of the unraveling and the transformation is also an unraveling. In the case that all U symbols are removed, we require no longer any context-sensitive and membership condition for simulation-completeness. We also show that the transformation preserves confluence of CTRSs.

Unfortunately, the transformation often fails to remove all U symbols. However, we have some advantages even if not all U symbols are removed.

- The context-sensitive condition is sometimes removed.
- Non-termination of CTRSs is preserved by the transformation. Thus, by showing termination of the unraveled CTRSs, we can prove 'effective termination' of the original CTRSs.

There are some cases in which the improvement in this paper increases the effect of the transformation (see Section 4). If we succeed in removing all U symbols, there are furthermore advantages as follows.

- The context-sensitive and membership condition is not necessary.
- Confluence of CTRSs is preserved. Accordingly, to prove confluence of the CTRSs, we can use many techniques for proving confluence of TRSs.

Therefore, the transformation is always harmless and we can sometimes obtain some advantages.

The unraveling for deterministic CTRSs is used in the *inversion compilers* proposed in [8, 9]. The compilers transform a given constructor TRS into a CTRS that computes (partial) inverse images of functions defined in the TRS. The compilers then unravel the CTRS to a TRS whose rules may have extra variables. Since inverse images are not mappings in general, CTRSs obtained by the compilers are not always confluent. From this reason, this paper does not assume confluence for CTRSs. The transformation in this paper is sometimes useful for simplifying TRSs obtained by the compilers. We will show an example at the end of this paper.

This paper is organized as follows. In Section 2, we give notations of term rewriting. In Section 3, we improve the unraveling for deterministic CTRSs. In Section 4, we propose a transformation that removes extra function symbols introduced due to the improved unraveling. In Section 5, we discuss confluence of CTRSs and the unraveled CTRSs. In Section 6, we enhance the condition for removing the extra function symbols in the transformation. In Section 7, we offer some concluding remarks.

2 Preliminaries

This paper follows the basic notions of term rewriting [2, 11]. In this section we outline the basic notations.

Through this paper, we use $\mathcal{V}$ as a countably infinite set of *variables*. The set of all *terms* over a *signature* $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F},\mathcal{V})$. The set of all variables appearing in either of terms $t_1,\ldots,t_n$ is represented by $\mathcal{V}ar(t_1,\ldots,t_n)$. The *identity* of terms s and t is denoted by $s \equiv t$. The notation $t|_p$ represents the subterm of t at a position p. The function symbol at the *root position* ε of t is denoted by $\mathtt{root}(t)$. The notation $\mathcal{C}[t_1,\ldots,t_n]_{p_1,\ldots,p_n}$ represents the term obtained by replacing $\Box$ at position p_i of an n-hole *context* C with term t_i for $1 \leq i \leq n$. The *domain* and *range* of a *substitution* σ are denoted by $\mathcal{D}om(\sigma)$ and $\mathcal{R}an(\sigma)$, respectively. The *composition* $\sigma\theta$ of substitutions σ and θ is defined as $\sigma\theta(x) = \theta(\sigma(x))$.

An *(oriented) conditional rewrite rule* over a signature $\mathcal{F}$ is a triple $(l, r, \mathcal{C}nd)$, denoted by $l \to r \Leftarrow \mathcal{C}nd$, such that the *left-hand side* (*lhs*) l is a non-variable term in $\mathcal{T}(\mathcal{F},\mathcal{V})$, the *right-hand side* (*rhs*) r is a term in $\mathcal{T}(\mathcal{F},\mathcal{V})$, and the *conditional part* $\mathcal{C}nd$ is in form of $s_1 \to t_1 \wedge \cdots \wedge s_n \to t_n$ $(n \geq 0)$ of terms s_i and t_i in $\mathcal{T}(\mathcal{F},\mathcal{V})$. In particular, the conditional rewrite rule $l \to r \Leftarrow \mathcal{C}nd$ is said to be an *(unconditional) rewrite rule* if $n = 0$, and we may abbreviate it to $l \to r$. We say that a binary relation $\approx$ and a substitution σ *satisfy* the conditional part $\mathcal{C}nd$, written by $\mathcal{C}nd(\sigma, \approx)$, if $s_i\sigma \approx t_i\sigma$ for $1 \leq i \leq n$. We denote $l \to r \Leftarrow \mathcal{C}nd$ with a unique label ρ by $\rho : l \to r \Leftarrow \mathcal{C}nd$. To simplify notations, we may write labels instead of the corresponding rules. For a conditional rewrite rule $\rho : l \to r \Leftarrow \mathcal{C}nd$, variables occurring not in l but in either r or $\mathcal{C}nd$ are called *extra variables* of ρ. The set of all extra variables of ρ is denoted by $\mathcal{EV}ar(\rho)$.

Let R be a finite set of conditional rewrite rules over a signature $\mathcal{F}$. The *n-level rewrite relation* $\xrightarrow[n]{}_R$ of R is defined inductively as follows: $\xrightarrow[0]{}_R = \emptyset$ and $\xrightarrow[n+1]{}_R = \{(C[l\sigma]_p, C[r\sigma]_p) \mid \rho : l \to r \Leftarrow \mathcal{C}nd \in R, \mathcal{C}nd(\sigma, \xrightarrow[n]{*}_R)\}$. The *rewrite relation* $\to_R$ of R is defined as $\to_R = \bigcup_{n\geq 0} \xrightarrow[n]{}_R$. To specify the position p and the rule ρ, we write $s \to_R^p t$ or $s \to_R^{[p,\rho]} t$. An *(oriented) conditional rewriting system* (*CTRS*) over a signature $\mathcal{F}$ is an abstract reduction system $(\mathcal{T}(\mathcal{F},\mathcal{V}), \to_R)$ of $\mathcal{T}(\mathcal{F},\mathcal{V})$ and the rewrite relation of a finite set R of conditional rewrite rules over $\mathcal{F}$. We use the set R of rules to denote the CTRS $(\mathcal{T}(\mathcal{F},\mathcal{V}), \to_R)$. A CTRS is called a *term rewriting system with extra variables* (*EV-TRS*) if it contains only unconditional rewrite rules. Specifically, it is a *term rewriting system* (*TRS*) if $\mathcal{V}ar(l) \supseteq \mathcal{V}ar(r)$ for every its rule $l \to r$.

A CTRS R is called a *1-CTRS* if every rule in R has no extra variable, a *2-CTRS* if every rule in R has no extra variable in its right-hand side, a *3-CTRS* if for every rule in R all extra variables of the rule appear in the conditional part, and a *4-CTRS* if no restriction is imposed. A conditional rewrite rule $\rho : l \to r \Leftarrow s_1 \to t_1 \cdots s_k \to t_k$ is called *deterministic* if $\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l, t_1, \ldots, t_{i-1})$ for $1 \leq i \leq k$. A CTRS is called *normal* if every its rule $l \to r \Leftarrow s_1 \to t_1 \wedge \cdots \wedge s_k \to t_k$ satisfies that $t_1,\ldots,t_k$ are ground normal forms of $R_u = \{\, l \to r \mid l \to r \Leftarrow \mathcal{C}nd \in R \,\}$.

We use the notion of *context-sensitive reduction* in [5]. Let $\mathcal{F}$ be a signature. A *context-sensitive condition* (*replacement mapping*) μ is a mapping from $\mathcal{F}$ to a set of integer lists such that $\mu(f) \subseteq \{1, \ldots, n\}$ for n-ary symbols f *in* $\mathcal{F}$. When $\mu(f)$ is not defined explicitly, we assume that $\mu(f) = \{1, \ldots, n\}$. The set $\mathcal{O}_\mu(t)$ of *replacing* (*active*) *positions* of a term t is defined inductively as follows: $\mathcal{O}_\mu(x) = \emptyset$ if $x \in \mathcal{V}$, and $\mathcal{O}_\mu(f(t_1, \ldots, t_n)) = \{ip \mid f \in \mathcal{F}, i \in \mu(f), p \in \mathcal{O}_\mu(t_i)\}$. The *context-sensitive reduction* of an EV-TRS R with μ is defined as $\to_{(R,\mu)} = \{(s,t) \mid s \to^p_R t, p \in \mathcal{O}_\mu(s)\}$. An abstract reduction system $(\mathcal{T}(\mathcal{F}, \mathcal{V}), \to_{(R,\mu)})$, denoted by (R, μ), is called a *context-sensitive reduction system* (*CS-TRS*).

In this paper we use a simple variant of *membership-conditional systems* [13]. For an EV-TRS R, the *membership-conditional reduction* of $\to_R$ by a *membership condition* $\in T$ (where $T \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$) is defined as $\xrightarrow[\in T]{}_R = \{(C[l\sigma]_p, C[r\sigma]_p) \mid l \to r \in R, (\forall x \in \mathcal{V}ar(l,r), x\sigma \in T)\}$. The membership-conditional reduction for $\to_{(R,\mu)}$ is defined similarly as $\xrightarrow[\in T]{}_{(R,\mu)}$.

3 Improvement of Unraveling for Deterministic CTRSs

In this section, we improve the unraveling (denoted by $\mathbb{U}_O$ in this paper) for deterministic CTRSs, which is proposed in [4, 7–9]. The unraveling $\mathbb{U}_O$ is a variant of Ohlebusch's unraveling [10]. The idea for this improvement is based on the unraveling for normal CTRSs [6], which is denoted by $\mathbb{U}_N$.

We first explain the intuitive idea of our improvement method. The unraveling $\mathbb{U}_O$ decomposes each conditional rewrite rule ρ having k conditions into $k+1$ unconditional rewrite rules that are used to evaluate the conditions in left-to-right order, introducing 'fresh' extra function symbols, called *U symbols* (see Fig. 1). For example, the conditional rewrite rule

$$\rho_1 : \mathsf{f}(x, y) \to z \Leftarrow \mathsf{g}(x) \to w \wedge \mathsf{g}(y) \to z \wedge \mathsf{h}(w, x) \to z$$

is unraveled into the following four unconditional rewrite rules, by introducing U symbols u_1, u_2 and u_3:

$$\mathbb{U}_O(\rho_1) = \left\{ \begin{array}{ll} \mathsf{f}(x,y) \to \mathsf{u}_1(\mathsf{g}(x), x, y), & \mathsf{u}_1(w, x, y) \to \mathsf{u}_2(\mathsf{g}(y), w, x), \\ \mathsf{u}_2(z, w, x) \to \mathsf{u}_3(\mathsf{h}(w, x), z), & \mathsf{u}_3(z, z) \to z \end{array} \right\}.$$

The application order of these rules in a reduction sequence corresponds exactly to the order of evaluating the conditions. However, the order between u_1 and u_2 is not necessary because the first and second conditions $\mathsf{g}(x) \to w$ and $\mathsf{g}(y) \to z$ can be evaluated in parallel. The reason is that all variables x, y used in the evaluation already appear in the lhs $\mathsf{f}(x, y)$ of the conditional rule. From this fact, we can combine u_1 and u_2 into one symbol u'_1 as follows:

$$\mathsf{f}(x,y) \to \mathsf{u}'_1(\mathsf{g}(x), \mathsf{g}(y), x) \text{ and } \mathsf{u}'_1(w, z, x) \to \mathsf{u}_3(\mathsf{h}(w, x), z).$$

Thus, to allow simultaneous evaluation of conditions that can be evaluated in parallel, we improve the ordinary unraveling $\mathbb{U}_O$ so that some conditional rules are decomposed to less unconditional rules. This idea comes from the unraveling $\mathbb{U}_N$ for normal CTRSs [6].

This improvement is formalized as follows. Here, we denote by $\overrightarrow{T}$ the sequence of the elements (in some fixed order) in the finite set T of terms, and denote $\bigcup_{t \in T} \mathcal{V}ar(t)$ by $\mathcal{V}ar(T)$.

Definition 1. *Let R be a deterministic CTRS over a signature $\mathcal{F}$. We consider a conditional rewrite rule $\rho : l \to r \Leftarrow \bigwedge_{j=1}^{m_1} s_{1,j} \to t_{1,j} \wedge \cdots \wedge \bigwedge_{j=1}^{m_k} s_{k,j} \to t_{k,j} \in R$ [1] such that $\mathcal{V}ar(s_{i,j}) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(T_1) \cup \cdots \cup T_{i-1})$ for all i and j, where $T_i = \{t_{i,1}, \ldots, t_{i,m_i}\}$. For every conditional rewrite rule ρ in the above form, let $|\rho|$ denote the number of groups of conditions in ρ (that is, $|\rho| = k$), and we need k 'fresh' function symbols $\mathtt{u}_1^\rho, \ldots, \mathtt{u}_k^\rho$, called* U symbols*, in the transformation. We transform ρ into a set $\mathbb{U}(\rho)$ of $k+1$ unconditional rewrite rules as follows:*

$$\mathbb{U}(\rho) = \begin{cases} \hfill l \to \mathtt{u}_1^\rho(s_{1,1}, \ldots, s_{1,m_1}, \overrightarrow{X_1}), \\ \mathtt{u}_1^\rho(t_{1,1}. \ldots, t_{1,m_1}, \overrightarrow{X_1}) \to \mathtt{u}_2^\rho(s_{2,1}, \ldots, s_{2,m_2}, \overrightarrow{X_2}), \\ \hfill \vdots \hfill \\ \mathtt{u}_k^\rho(t_{k,1}, \ldots, t_{k,m_k}, \overrightarrow{X_k}) \to r \end{cases}$$

where $S_i = \{s_{i,1}, \ldots, s_{i,m_i}\}$ and $X_i = (\mathcal{V}ar(l) \cup \mathcal{V}ar(T_1 \cup \cdots \cup T_{i-1})) \cap (\mathcal{V}ar(T_i) \cup \mathcal{V}ar(S_{i+1} \cup T_{i+1} \cup \cdots \cup S_k \cup T_k) \cup \mathcal{V}ar(r))$ for $1 \le i \le k$. The set $\mathbb{U}(R) = \bigcup_{\rho \in R} \mathbb{U}(\rho)$ is an EV-TRS over the extended signature $\mathcal{F}_{\mathbb{U}(R)} = \mathcal{F} \cup \{\mathtt{u}_i^\rho \mid \rho \in R, 1 \le i \le |\rho|\}$.

The set X_i in the above definition plays the role of delivering values to the later conditions; these values are obtained via variables in either l, T_1, $\cdots$ or T_{i-1}, and they are used in either r, $S_{i+1}, \ldots, S_k$ or $T_i, \ldots, T_k$. The above unraveling $\mathbb{U}$ is based on the unraveling $\mathbb{U}_O$ [4, 7–9], in which the definition of X_i is different from the original definition [10]. For this reason, all results in this paper or [7–9] do not hold for the original unraveling. In the above definition, one can freely divide a conditional part into groups of conditions that satisfy the variable-occurrence condition. The set $\mathbb{U}(\rho)$ is equal to $\mathbb{U}_O(\rho)$ if $m_i = 1$ for every i, and it is equal to $\mathbb{U}_N(\rho)$ if $k = 1$. Thus, $\mathbb{U}_O$ and $\mathbb{U}_N$ are special cases of $\mathbb{U}$. For the purpose of reducing the number of unconditional rules, this paper assumes that ρ in the above definition satisfies $\mathcal{V}ar(s_{i,j}) \not\subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(T_1 \cup \cdots \cup T_{i-2})$ for $1 < i \le k$ and $1 \le j \le m_i$. Under this assumption, $\mathbb{U}(\rho)$ is determined uniquely.

Example 2. The conditional rule ρ_1 is unraveled by $\mathbb{U}$ into $\mathbb{U}(\rho_1) = \{\, \mathsf{f}(x, y) \to \mathtt{u}_1'(\mathsf{g}(x), \mathsf{g}(y), x), \mathtt{u}_1'(w, z, x) \to \mathtt{u}_3(\mathsf{h}(w, x), z), \mathtt{u}_3(z, z) \to z\}$. The number of rules obtained by $\mathbb{U}$ is five while that obtained by $\mathbb{U}_O$ is six.

Next, we give the notion of *simulation-completeness* based on completeness of *ultra-properties* [6].

Definition 3. *Let U be an unraveling and R be a CTRS over a signature $\mathcal{F}$.*

- *U is said to be* $\xrightarrow{*}_R$-preserving *for R if U preserves reachability of R, that is, for all terms s and $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, $s \xrightarrow{*}_R t$ implies $s \xrightarrow{*}_{U(R)} t$.*

[1] It is clear that every deterministic conditional rewrite rule can be expressed like this.

– *U is* simulation-sound *for R if U is sound for unreachability of R, that is, for all s and $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$, $s \xrightarrow{*}_R t$ if $s \xrightarrow{*}_{U(R)} t$.*
– *U is* simulation-complete *for R if U is complete ($\xrightarrow{*}_R$-preserving and sound for $\xrightarrow{*}_R$), that is, for all s and $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$, $s \xrightarrow{*}_R t$ if and only if $s \xrightarrow{*}_{U(R)} t$.*

We similarly define these properties for the unraveled system $U(R)$.

The definition of simulation-completeness in [7–9] is different from that used in this paper. More precisely, simulation-completeness in [7–9] corresponds to simulation-soundness in this paper. However, discussions on the simulation-completeness in those papers are essentially equivalent because $\xrightarrow{*}_R$-preserving holds for all CTRSs.

All proposed unravelings are $\xrightarrow{*}_R$-preserving for every target CTRS because $\xrightarrow{*}_R$-preserving is a necessary condition for a transformation that is an 'unraveling'. On the other hand, in general, they are not simulation-sound for all target CTRSs, and hence are simulation-incomplete. The cause is that the unraveled CTRSs are approximations of the original CTRSs. In [6], we can find a counterexample against simulation-completeness of $\mathbb{U}_N$, $\mathbb{U}_O$ and Ohlebusch's unraveling.

A restriction to reductions of the unraveled CTRSs for avoiding this difficulty on simulation-incompleteness of $\mathbb{U}_O$ is shown in [8], which is done by a particular *context-sensitive* and *membership* condition that prohibits reductions associated with the following redexes:

– redexes that occur strictly below U symbols, except for the first arguments of the U symbols, or
– redexes that contain a U symbol in their proper subterms.

The context-sensitive condition μ_ρ for ρ in Definition 1 and the membership condition become as follows:

– $\mu_\rho(\mathtt{u}_i^\rho) = \{1, \ldots, m_i\}$ for every $\mathtt{u}_i^\rho$, and
– the membership condition is "$\in \mathcal{T}(\mathcal{F},\mathcal{V})$".

The context-sensitive condition μ_R for R is defined as $\mu_R(\mathtt{u}_i^\rho) = \mu_\rho(\mathtt{u}_i^\rho)$ (and $\mu(f) = \{1, \ldots, n\}$ for all n-ary symbols $f \in \mathcal{F}$). For $\mathbb{U}(\rho_1)$ in Example 2, the context-sensitive condition μ_{ρ_1} is specified as $\mu_{\rho_1}(\mathtt{u}'_1) = \{1, 2\}$ and $\mu_{\rho_1}(\mathtt{u}_3) = \{1\}$. We denote the CS-TRSs $(\mathbb{U}(\rho), \mu_\rho)$, $(\mathbb{U}(R), \mu_R)$ and $(\mathbb{U}_O(R), \mu_R)$ by $\mathbb{U}_\mu(\rho)$, $\mathbb{U}_\mu(R)$ and $\mathbb{U}_{O\mu}(R)$, respectively. We consider $\mathbb{U}_\mu$ and $\mathbb{U}_{O\mu}$ as unravelings from CTRSs to CS-TRSs.

Theorem 4 ([8]). *For every deterministic CTRS R over a signature $\mathcal{F}$, $\mathbb{U}_{O\mu}$ is simulation-complete (with respect to the membership-condition "$\in \mathcal{T}(\mathcal{F},\mathcal{V})$"), that is, for all s and $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$, $s \xrightarrow{*}_R t$ if and only if $s \xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{\mathbb{U}_{O\mu}(R)} t$.*

In the rest of this paper, we assume that the membership condition "$\in \mathcal{T}(\mathcal{F},\mathcal{V})$" is imposed on reductions.

Similarly to other unravelings, $\mathbb{U}$ is not simulation-complete for all CTRSs while $\mathbb{U}$ is $\xrightarrow{*}_R$-preserving. However, $\mathbb{U}_\mu$ is always simulation-complete for R with respect to $\xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{}_{\mathbb{U}_\mu(R)}$.

Theorem 5. *Theorem 4 also holds for $\mathbb{U}_\mu$.*

Proof (Sketch). We only show that the CS-TRS $\mathbb{U}_\mu(R)$ is simulation-sound for R, that is, for all s and t in $\mathcal{T}(\mathcal{F},\mathcal{V})$, $s \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{\mathbb{U}_\mu(R)} t$ implies $s \xrightarrow{*}_R t$. This claim can be straightforwardly proved by induction on the lexicographic products of term structure and steps k of $s \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{k}_{\mathbb{U}_\mu(R)} t$.

Another approach to this proof is to construct the following rule from ρ in Definition 1; $\rho' : l \to r \Leftarrow \mathsf{tp}_{m_1}(s_{1,1},\ldots,s_{1,m_1}) \to \mathsf{tp}_{m_1}(t_{1,1},\ldots,t_{1,m_1}) \wedge \cdots \wedge \mathsf{tp}_{m_k}(s_{k,1},\ldots,s_{k,m_k}) \to \mathsf{tp}_{m_k}(t_{k,1},\ldots,t_{k,m_k})$ where tp_j is a fresh constructor not in $\mathcal{F}$ that represents the tuple of j terms $t_1,\ldots,t_j$. This ρ' is deterministic and satisfies that $\mathbb{U}_O(\rho') = \mathbb{U}(\rho')$ and $\mu_{\rho'}(\mathtt{u}_i^{\rho'}) = \{1\}$. Let R' be a CTRS obtained by the above transformation of the rules in R; then it is clear that $\to_R \, = \, \to_{R'}$ and $\xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{(\mathbb{U}_\mu(R))} = \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{(\mathbb{U}_O(R'),\bigcup_{\rho'\in R'}\mu_{\rho'})}$ on terms in $\mathcal{T}(\mathcal{F},\mathcal{V})$. It follows from Theorem 4 that $\xrightarrow{*}_{R'} = \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{(\mathbb{U}_O(R'),\bigcup_{\rho'\in R'}\mu_{\rho'})}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$. Therefore, we have $\xrightarrow{*}_R = \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{(\mathbb{U}_\mu(R))}$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$. $\square$

The transformation in the above proof is also adequate for our purpose. However, we proposed $\mathbb{U}$ because $\mathbb{U}$ helps us to describe the transformation proposed later.

4 Reducing Context-Sensitive and Membership Conditions

In this section, we propose a transformation to relax the context-sensitive and membership condition of $(\mathbb{U}(R),\mu_R)$. In fact, the transformation reduces the number of U symbols in $\mathbb{U}(R)$. This leads to the relaxation of the condition because the condition depends on the existence of U symbols. Simply speaking, the transformation folds two rules having the same U symbol into one rule, that is, the replacement of $l_1 \to l_2\delta$ and $l_2 \to r_2$ with $l_1 \to r_2\delta$ where $\mathtt{root}(l_2)$ is a U symbol (see Fig. 2). When all U symbols are removed from $\mathbb{U}(R)$, we can obtain an unconditional system that works equally for R without the context-sensitive and membership condition. There are some cases where the context-sensitive condition is not necessary even if U symbols are still remaining.

We first give examples showing our intuitive idea of the transformation process. For an EV-TRS R, we say that a context-sensitive condition μ is *ineffective* for R if $\mu(f) = \{1,\ldots,n\}$ for all n-ary symbols f that may be a U symbol. Let us consider a conditional rewrite rule $\rho_2 : \mathsf{f}(x,y) \to z \Leftarrow \mathsf{g}(x) \to w \wedge \mathsf{f}(w,y) \to z$. This is unraveled by $\mathbb{U}_\mu$ to $(\mathbb{U}(\rho_2),\mu_{\rho_2})$ where

$$\mathbb{U}(\rho_2) = \{\mathsf{f}(x,y) \to \mathtt{u}_4(\mathsf{g}(x),y),\quad \mathtt{u}_4(w,y) \to \mathtt{u}_5(\mathsf{f}(w,y)),\quad \mathtt{u}_5(z) \to z\}$$

and $\mu_{\rho_2}(\mathtt{u}_4) = \mu_{\rho_2}(\mathtt{u}_5) = \{1\}$. The first and second rules are used in order like "$\cdots \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*} \mathsf{f}(x,y)\sigma_1 \to \mathtt{u}_4(\mathsf{g}(x),y)\sigma_1 \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*} \mathtt{u}_4(w,y)\sigma_2 \to \mathtt{u}_5(\mathsf{f}(w,y)\sigma_2) \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*} \cdots$" where we ignore contexts over this sequence. This reduction sequence can be simulated by the rule $\mathsf{f}(x,y) \to \mathtt{u}_5(\mathsf{f}(\mathsf{g}(x),y))$ as like $\cdots \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*}$

$\mathsf{f}(x,y)\sigma_1 \to \mathsf{u}_5(\mathsf{f}(\mathsf{g}(x),y)\sigma_1) \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*} \mathsf{u}_5(\mathsf{f}(w,y)\sigma_2) \xrightarrow[\in\mathcal{T}(\mathcal{F},\mathcal{V})]{*} \cdots$. In a similar fashion, we also remove u_5 as follows:

$$\{\, \mathsf{f}(x,y) \to \mathsf{f}(\mathsf{g}(x),y) \,\}.$$

The above rule has no U symbol which means the context-sensitive and membership condition is not necessary.

Let us consider the more complicated case of the rule ρ_1. This rule is unraveled to $\mathbb{U}(\rho_1)$ in Example 2 with μ_{ρ_1}. Similarly to the previous example ρ_2, the first and second rules are replaced with $\mathsf{f}(x,y) \to \mathsf{u}_3(\mathsf{h}(\mathsf{g}(x),x),\mathsf{g}(y))$. At this time, possible reductions at position 2 of $\mathsf{u}'_1(\mathsf{g}(x),\mathsf{g}(y),x)$ must be done at position 2 of $\mathsf{u}_3(\mathsf{h}(\mathsf{g}(x),x),\mathsf{g}(y))$. To allow these reductions, the context-sensitive condition μ_{ρ_1} must be updated as $\mu'_{\rho_1}(\mathsf{u}_3) = \{1,2\}$. Since we have only one U symbol u_3, the context-sensitive condition μ'_{ρ_1} is ineffective. In this way, we reduce the number of U symbols from $\mathbb{U}(R)$, reducing and updating the context-sensitive conditions.

The transformation removing U symbols is formalized as follows:

Definition 6. *Let ρ be a deterministic conditional rewrite rule over a signature $\mathcal{F}$. We define pairs (S_i,μ_i) recursively as follows:*

1. $(S_0,\mu_0) := (\mathbb{U}(\rho),\mu_\rho)$ [2].
2. *Select a* removable *U symbol* u^ρ_j *from* S_i *such that* $S_i = \{l \to \mathsf{u}^\rho_j(t_1\delta,\ldots,t_m\delta),\ \mathsf{u}^\rho_j(t_1,\ldots,t_m) \to r\,\} \uplus R'$ [3] *for some substitution δ, that is,*
   - (guarding replacing positions) $t_k\delta \equiv t_k$ *for all* $k \notin \mu_i(\mathsf{u}^\rho_j)$ [4], *and*
   - (RMC) *if* $\mathtt{root}(r)$ *is a U symbol (let* $\mathtt{root}(r) = \mathsf{u}$*), then no variable in* $\mathcal{D}om(\delta)$ *is shared between terms at positions in* $\mu_i(\mathsf{u})$ *and at positions not in* $\mu_i(\mathsf{u})$ [5].

   We let $S_{i+1} := \{l \to r\delta\} \cup R'$, $\mu_{i+1}(f) := \mu_i(f)$ *for* $f \in \mathcal{F}_{\mathbb{U}(\rho)} \setminus \{\mathsf{u}^\rho_j\}$ *and*
   - (updating μ) *if* $\mathtt{root}(r)$ *is a U symbol, let* $\mathtt{root}(r) = \mathsf{u}$*, then* $\mu_{i+1}(\mathsf{u}) := \mu_i(\mathsf{u}) \cup \{\, k \mid 1 \le k \le m, r|_k \in \mathcal{D}om(\delta) \,\}$.

We denote (S_i,μ_i) by $\mathbb{T}_i(\mathbb{U}_\mu(\rho))$, and define $\mathbb{T}(\mathbb{U}_\mu(\rho)) = (S_{i'},\mu_{i'})$ where $(S_{i'},\mu_{i'}) = (S_{i'+1},\mu_{i'+1})$. For a deterministic CTRS R, we define $\mathbb{T}(\mathbb{U}_\mu(R)) = (\bigcup_{\rho\in R} R_\rho, \bigcup_{\rho\in R}\mu_\rho)$ where $\mathbb{T}(\mathbb{U}_\mu(\rho)) = (R_\rho,\mu_\rho)$. Note that $\bigcup_{\rho\in R}\mu_\rho$ is well-defined as a mapping because the domains of μ_ρs are disjoint.

The above transformation always terminates because the number of U symbols are finite and a U symbol is removed at every step, that is, i' is at most $|\rho|$.

Example 7. $\mathbb{U}_\mu(\rho_1)$ is transformed by $\mathbb{T}$ into $\mathbb{T}(\mathbb{U}_\mu(\rho_1)) = (R_1,\mu_{R_1})$ where $R_1 = \{\, \mathsf{f}(x,y) \to \mathsf{u}_3(\mathsf{h}(\mathsf{g}(x),x),\mathsf{g}(y)),\ \mathsf{u}_3(z,z) \to z \,\}$ and $\mu_{R_1}(\mathsf{u}_3) = \{1,2\}$. The membership condition is necessary for the above system because of the existence of U

[2] We write $\mu = \mu'$ if $\mu(f) = \mu'(f)$ for all f.

[3] These two rules are the only rules in S_i which contain u^ρ_j.

[4] More precisely, $\mathcal{D}om(\delta) \subseteq (\bigcup_{k\in\mu_i(\mathsf{u}^\rho_j)} \mathcal{V}ar(t_k)) \setminus (\bigcup_{k\notin\mu_i(\mathsf{u}^\rho_j)} \mathcal{V}ar(t_k))$.

[5] That is, $\mathcal{D}om(\delta) \cap (\bigcup_{k\in\mu_i(\mathsf{u})} \mathcal{V}ar(t_k) \cap \bigcup_{k\notin\mu_i(\mathsf{u})} \mathcal{V}ar(t_k)) = \emptyset$.

symbols u_3. On the other hand, the above μ_{R_1} is ineffective for R_1. Therefore, we succeed in removing the context-sensitive condition, although the membership condition still remains.

There are non-deterministic choices for selecting U symbols at the second step in Definition 6 because there are possibly some removable U symbols. This means that the final products of $\mathbb{T}$ for $\mathbb{U}_\mu(R)$ are not unique in general. For example, consider the conditional rule $\rho_3 : \mathsf{f}(x, x') \to z \Leftarrow \mathsf{g}(x) \to y \wedge \mathsf{g}(x') \to z \wedge \mathsf{g}(y) \to w \wedge \mathsf{h}(w, z) \to z$. Here, there are two results of $\mathbb{T}(\mathbb{U}_\mu(\rho_3))$ while they become unique if the fourth condition $\mathsf{f}(w, z) \to z$ is replaced with $\mathsf{f}(w, z) \to v$. The same is said of $\mathbb{U}_O(R)$. As another example, consider the rule $\rho_4 : \mathsf{f}(x, x') \to \mathsf{h}(y, w) \Leftarrow \mathsf{g}(x) \to y \wedge \mathsf{g}(x') \to z \wedge \mathsf{h}(y, z) \to w \wedge \mathsf{g}(y) \to \mathsf{b}$. There are two results of $\mathbb{T}(\mathbb{U}_{O\mu}(\rho_4))$ and they become unique if the fourth condition is removed from ρ_4. On the other hand, $\mathbb{T}(\mathbb{U}_\mu(\rho_4))$ is unique. This means that the improvement of $\mathbb{U}_O$ in Section 3 is effective for some cases. In this way, the result of $\mathbb{T}$ is not always unique. However, it is clear that the number of all possible results is finite. Therefore, one can select the most 'favorite' in all results, for instance, one of the results whose number of rules is the least. Note that the transformation $\mathbb{T}$ does not always succeed in removing all U symbols even if we search all possible results exhaustively. To determine $\mathbb{T}(\mathbb{U}_\mu(R))$ uniquely, in this paper, we select the u^ρ_j at every step of S_i, whose index j is the greatest in all removable U symbols of ρ.

The condition RMC in Definition 6 is necessary for preserving simulation-completeness. In other words, ignoring this condition leads to systems without simulation-completeness. For example, consider the CTRS $R_2 = \{\rho_3\} \cup R_3$ where $R_3 = \{\ \mathsf{g}(\mathsf{a}) \to \mathsf{b},\ \mathsf{g}(\mathsf{b}) \to \mathsf{c},\ \mathsf{h}(\mathsf{g}(x), \mathsf{g}(\mathsf{a})) \to \mathsf{b}\ \}$. The CTRS R_2 is unraveled by $\mathbb{U}$ and transformed by $\mathbb{T}$ into (R'_2, μ_2) where $R'_2 = R_3 \cup \{\ \mathsf{f}(x, x') \to \mathsf{u}_6(\mathsf{g}(\mathsf{g}(x)), \mathsf{g}(x')),\ \mathsf{u}_6(w, z) \to \mathsf{u}_7(\mathsf{h}(w, z), z),\ \mathsf{u}_7(z, z) \to z\ \}$ and $\mu_2(\mathsf{u}_7) = \{1\}$. Furthermore, consider the CS-TRS (R_4, μ_4) where $R_4 = R_3 \cup \{\ \mathsf{f}(x, x') \to \mathsf{u}_7(\mathsf{h}(\mathsf{g}(\mathsf{g}(x)), \mathsf{g}(x')), \mathsf{g}(x'))),\ \mathsf{u}_7(z, z) \to z\ \}$ and $\mu_4(\mathsf{u}_7) = \{1, 2\}$. The system (R_4, μ_4) is obtained by applying $\mathbb{T}$ to (R'_2, μ_2), ignoring RMC. This system is not simulation-complete for $\mathbb{U}_\mu(R_2)$ because we have $\mathsf{f}(\mathsf{a}, \mathsf{a}) \xrightarrow{*}_{(R_4, \mu_4)} \mathsf{b}$ but not $\mathsf{f}(\mathsf{a}, \mathsf{a}) \xrightarrow{*}_{\mathbb{U}_\mu(R_2)} \mathsf{b}$. The variable z at position 2 of the term $\mathsf{u}_6(\mathsf{h}(y, z), z)$ should be used only for delivering value. For this reason, this z should not be instantiated by $\mathbb{T}$ with any term that does not finish being evaluated. This observation brings the condition RMC to the transformation $\mathbb{T}$.

One may think that 'simplification' in completion procedures appear adequate. However, it is too powerful for folding rules and hence it does not always preserve simulation-completeness and it sometimes collapses the feature of the conditional rules that we will describe later. The reason is that applying 'simplification' ignores RMC. Thus, 'simplification' is not adequate for our purpose.

Finally, we show correctness of $\mathbb{T}$, that is, simulation-completeness of $\mathbb{T}$.

Lemma 8. *Let ρ be a conditional rewrite rule in a deterministic CTRS R over a signature $\mathcal{F}$, and s and t be terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. Suppose that $\mathbb{T}_i(\mathbb{U}_\mu(\rho))$*

$= (R_i, \mu_i)$, $\mathbb{T}_{i+1}((R_i, \mu_i)) = (R_{i+1}, \mu_{i+1})$ *and* $\mathbb{U}_\mu(R \setminus \{\rho\}) = (R', \mu')$*. Then* $s \xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{(R_i \cup R', \mu_i \cup \mu')} t$ *if and only if* $s \xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{(R_{i+1} \cup R', \mu_{i+1} \cup \mu')} t$.

Proof (Sketch). Since we can easily prove the case that r in Definition 6 is not rooted with a U symbol, we only consider the remaining case. Moreover, proving the *only-if* part is not difficult. Hence, we only prove the *if* part by induction on the lexicographic products of term structure and the length of the reduction sequences. To simplify this proof, we use underlines for active positions, and $\xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{}_{(R_i \cup R', \mu_i \cup \mu')}$ and $\xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{}_{(R_{i+1} \cup R', \mu_{i+1} \cup \mu')}$ are denoted by $\to_i$ and $\to_{i+1}$, respectively.

We can assume without loss of generality the following:

- $R_i \setminus R_{i+1} = \{\, l \to \mathtt{u}_j^\rho(\underline{f(u, u, u', y)}, z),\ \mathtt{u}_j^\rho(\underline{f(x, x, x', y)}, z) \to \mathtt{u}(\underline{s'}, x, y, z) \,\}$,
- $R_{i+1} \setminus R_i = \{\, l \to \mathtt{u}(\underline{s'\delta, x\delta}, y, z) \,\}$,
- $\mathtt{u}(\underline{t'}, x, y, z) \to r' \in R_i$ and $\mathtt{u}(\underline{t', x}, y, z) \to r' \in R_{i+1}$.

where $\delta = \{x \mapsto u, x' \mapsto u'\}$, $\mu_i(\mathtt{u}_j^\rho) = \mu_i(\mathtt{u}) = \{1\}$ and $\mu_{i+1}(\mathtt{u}) = \{1, 2\}$. It follows from Rmc that $x \notin \mathcal{V}ar(s')$. We only show the most difficult case. Suppose that $s \xrightarrow{*}_{i+1} l\sigma_1 \to_{i+1} \mathtt{u}(\underline{s'\delta, x\delta}, y, z)\sigma_1 \xrightarrow{*}_{i+1} \mathtt{u}(\underline{t', x}, y, z)\sigma_2 \to_{i+1} r'\sigma_2 \xrightarrow{*}_{i+1} t$ where $\mathcal{R}an(\sigma_1) \cup \mathcal{R}an(\sigma_2) \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$. Then, it follows from the context-sensitive condition that $y\sigma_1 \equiv y\sigma_2$ and $z\sigma_1 \equiv z\sigma_2$. By the induction hypothesis, we have $s \xrightarrow{*}_i l\sigma_1$, $s'\delta\sigma_1 \xrightarrow{*}_i t'\sigma_2$, $x\delta\sigma_1 \xrightarrow{*}_i x\sigma_2$, and $r'\sigma_2 \xrightarrow{*}_i t$. It follows from $x \notin \mathcal{V}ar(s')$ that $s'\delta\sigma_1 \equiv s'\sigma_1$. Let $\theta = \{x \mapsto x\sigma_2, x' \mapsto u'\sigma_1, y \mapsto y\sigma_2, z \mapsto z\sigma_2\}$. Therefore, we have $s \xrightarrow{*}_i l\sigma_1 \to_i \mathtt{u}_j^\rho(\underline{f(u, u, u', y)}, z)\sigma_1 \xrightarrow{*}_i \mathtt{u}_j^\rho(\underline{f(x\sigma_2, x\sigma_2, u'\sigma_1, y\sigma_1)}, z\sigma_1) \equiv \mathtt{u}_j^\rho(\underline{f(x, x, x', y)}, z)\theta \to_i \mathtt{u}(\underline{s'}, x, y, z)\theta \equiv \mathtt{u}(\underline{s'\sigma_1}, x\sigma_2, y\sigma_2, z\sigma_2) \xrightarrow{*}_i \mathtt{u}(\underline{t'\sigma_2}, x\sigma_2, y\sigma_2, z\sigma_2) \to_i r'\sigma_2 \xrightarrow{*}_i t$. □

Theorem 9. *Let R be a deterministic CTRS over a signature $\mathcal{F}$. For all $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, $s \xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{\mathbb{U}_\mu(R)} t$ if and only if $s \xrightarrow[\in \mathcal{T}(\mathcal{F},\mathcal{V})]{*}_{\mathbb{T}(\mathbb{U}_\mu(R))} t$.*

From Lemma 8 and Theorems 9 and 5, the composition $\mathbb{T}(\mathbb{U}_\mu(\cdot))$ of the transformations can be considered as an unraveling with simulation-completeness.

Corollary 10. *Theorem 4 also holds for $\mathbb{T}(\mathbb{U}_\mu(\cdot))$.*

5 On Confluence of CTRSs

To prove confluence of CTRSs, simulation-completeness of the unravelings enable us to use confluence of the unraveled CTRSs.

Theorem 11. *Let R be a deterministic CTRS over a signature $\mathcal{F}$. If $\mathbb{U}(R)$ is confluent, then R is confluent.*

On the other hand, confluence of CTRSs is not preserved by unravelings, that is, the converse of Proposition 11 does not always hold in general. Consider a normal form of a confluent CTRS over a signature, which are matched with the lhs of a conditional rule with at least a condition. The normal form sometimes becomes reducible on the unraveled CTRS to determine whether the original conditional part is satisfied, although the conditional part is not satisfied. The normal form is not reachable to any terms over the original signature, and hence it is reduced to a normal form containing a U symbol. Thus, we can see that terms containing U symbols prevent the unravelings from preserving confluence of CTRSs. For this observation, as far as terms without U symbols are concerned, confluence of CTRSs are preserved by the unravelings if simulation-completeness is preserved. The unraveling $\mathbb{U}_\mu$ and the transformation $\mathbb{T}$ preserve simulation-completeness. Moreover, $\mathbb{T}$ sometimes remove all U symbols. In such cases, confluence of the systems obtained by $\mathbb{T}(\mathbb{U}_\mu(\cdot))$ coincides with that of the original CTRSs.

Corollary 12. *A deterministic CTRS R over a signature $\mathcal{F}$ is confluent if and only if $\mathbb{U}_\mu(R)$ (respectively $\mathbb{T}(\mathbb{U}_\mu(R))$) is confluent on $\mathcal{T}(\mathcal{F},\mathcal{V})$* [6]*. Especially, let $(R',\mu') = \mathbb{T}(\mathbb{U}_\mu(R)))$ and suppose that R' has no U symbol, then $\xrightarrow{*}_R = \xrightarrow{*}_{R'}$ (more precisely, $\rightarrow_{R'} \subseteq \rightarrow_R \subseteq \xrightarrow{+}_{R'}$), that is, R is confluent if and only if R' is.*

As long as we know, there are no methods to show confluence of $\mathbb{U}_\mu(R)$ and $\mathbb{T}(\mathbb{U}_\mu(R))$ on $\mathcal{T}(\mathcal{F},\mathcal{V})$ if U symbols still remain. However, to decide confluence of R, we can use ordinary techniques for deciding confluence of $\mathbb{T}(\mathbb{U}(\mu(R)))$ if $\mathbb{T}$ removes all U symbols.

The method in this paper appears to counter the other approaches to confluence, such as Bergstra and Klop's method [3]. In fact, the unraveled CTRSs often lose confluence of the original CTRSs as described above. However, the transformation $\mathbb{T}$ recovers the confluence that is lost in the process of the unravelings if all U symbols are removed successfully. Therefore, the transformation $\mathbb{T}$ is sometimes effective for preserving confluence of CTRSs.

6 Refinement of the Condition for Removing U Symbols

It is probably impossible to relax the condition Rmc in Definition 6. To the contrary, we should tighten Rmc for maintaining a feature of conditional rules associated with efficiency of reductions. Consider the following 'ML' program.

```
fun twofib 0 = (0,1)
  | twofib n = let val m = twofib (n-1)
               in (#2 m, (#2 m) + (#1 m) ) end;
```

It is known that the function `twofib` efficiently computes pairs of two continuous Fibonacci numbers. Such efficiency comes from the 'let' structure, and the first part of the 'let' structure can be considered as a conditional part. From this observation, the above program is regarded as the following CTRS:

$$R_5 = \begin{cases} \mathsf{twofib}(\mathsf{0}) \to \mathsf{tp}_2(\mathsf{0},\mathsf{s}(\mathsf{0})), \\ \mathsf{twofib}(\mathsf{s}(n)) \to \mathsf{tp}_2(\mathsf{\#2}(m),\mathsf{add}(\mathsf{\#2}(m),\mathsf{\#1}(m))) \Leftarrow \mathsf{twofib}(n) \to m, \\ \quad\vdots \end{cases}$$

[6] For all $s,t_1,t_2 \in \mathcal{T}(\mathcal{F},\mathcal{V})$, if $s \xrightarrow{*}_{(\mathbb{U}(R),\mu_R)} t_1$ and $s \xrightarrow{*}_{(\mathbb{U}(R),\mu_R)} t_2$ then there exists a term $u \in \mathcal{T}(\mathcal{F},\mathcal{V})$ such that $t_1 \xrightarrow{*}_{(\mathbb{U}(R),\mu_R)} u$ and $t_2 \xrightarrow{*}_{(\mathbb{U}(R),\mu_R)} u$.

where $\mathsf{tp}_2(t_1, t_2)$ denotes the pair of two terms t_1 and t_2. The second rule is unraveled into the system (R_6, μ_{R_6}) where

$$R_6 = \{\mathsf{twofib}(\mathsf{s}(n)) \to \mathtt{u}_8(\mathsf{twofib}(n)),\ \mathtt{u}_8(m) \to \mathsf{tp}_2(\mathtt{\#2}(m), \mathsf{add}(\mathtt{\#2}(m), \mathtt{\#1}(m)))\}$$

and $\mu_{R_6}(\mathtt{u}_8) = \{1\}$. Under innermost reduction strategy, efficiency is still alive in (R_6, μ_{R_6}). The system (R_6, μ_{R_6}) can be transformed by $\mathbb{T}$ as follows:

$$\mathsf{twofib}(\mathsf{s}(n)) \to \mathsf{tp}_2(\mathtt{\#2}(\mathsf{twofib}(n)), \mathsf{add}(\mathtt{\#2}(\mathsf{twofib}(n)), \mathtt{\#1}(\mathsf{twofib}(n)))).$$

$\mathbb{T}$ succeeded in removing all U symbols from (R_6, μ_{R_5}). This corresponds to the following 'ML' program.

```
fun twofib2 0 = (0,1)
  | twofib2 n = ( (#1 (twofib2 (n-1))),
                  (#2 (twofib2 (n-1)))+(#1 (twofib2 (n-1))) );
```

However, the above 'ML' program loses efficiency.

The 'let' structure provides a facility that separates the parallel evaluations of terms that are identical into one. For example, `twofib2 (n-1)` is evaluated once in the first 'ML' program and three times in the second 'ML' program. The advantage of coming from the 'let' structure is lost in the transformation $\mathbb{T}$, by instantiating variable m in $\mathsf{tp}_2(\mathtt{\#2}(m), \mathsf{add}(\mathtt{\#2}(m), \mathtt{\#1}(m)))$, whose occurrence is non-linear, with $\mathsf{twofib}(n)$. In order to prevent such instantiation in these cases, we enhance the condition Rmc as follows:

(Rmc′) *r is linear with respect to $\mathcal{D}om(\delta)$.*

It is clear that Rmc′ implies Rmc. The enhanced condition Rmc′ does not cause the target systems to lose the essential advantage of the original CTRSs, such as efficiency that comes from 'let' structure. For confluent CTRSs, simulation-completeness holds without Rmc. However, Rmc′ should not be ignored because of the points outlined in the above discussion.

7 Concluding Remarks and Related Works

We firstly show an application of our method. Consider the following rule obtained by the inversion compiler [8], from the TRS that computes multiplication:

$$\rho_{\mathsf{div}} : \mathsf{div}(\mathsf{s}(z), \mathsf{s}(y)) \to \mathsf{tp}_1(\mathsf{s}(x)) \Leftarrow \mathsf{sub}(z, y) \to \mathsf{tp}_1(w) \land \mathsf{div}(w, \mathsf{s}(y)) \to \mathsf{tp}_1(x),$$

where div and sub compute division and subtraction of natural numbers, respectively, and $\mathsf{tp}_i(t_1, \ldots, t_i)$ denotes the tuple of i terms $t_1, \ldots, t_i$. Since we can consider $\mathsf{tp}_1(t)$ as t similarly to several functional languages, we can easily see that the following rule seems to be similar to the above rule in the sense of computing division [7]:

$$\rho'_{\mathsf{div}} : \mathsf{div}(\mathsf{s}(z), \mathsf{s}(y)) \to \mathsf{s}(x) \Leftarrow \mathsf{sub}(z, y) \to w \land \mathsf{div}(w, \mathsf{s}(y)) \to x.$$

[7] Note that $\mathsf{tp}_1(t)$ cannot be abbreviated to t in all cases.

This rule is transformed by $\mathbb{T}(\mathbb{U}_\mu(\cdot))$ into the following rule:

$$\mathsf{div}(\mathsf{s}(z), \mathsf{s}(y)) \to \mathsf{s}(\mathsf{div}(\mathsf{sub}(z, y), y)).$$

Using $\mathbb{T}$, we succeeded in removing all U symbols from $\mathbb{U}(\rho'_{\mathsf{div}})$, and the above rule coincides with the typical rewrite rule of division $\mathsf{s}(x) \div \mathsf{s}(y) \to \mathsf{s}((x - y) \div \mathsf{s}(y))$. This tells us that the program generated by the compiler seems to be correct, in comparison with the handmade program.

Finally, we briefly offer some extra remarks.

- Two syntactic conditions to preserve simulation-completeness without the context-sensitive and membership condition [7] also hold for $\mathbb{U}$ and $\mathbb{T}(\mathbb{U}(\cdot))$. Neither of the two syntactic conditions are sufficient and necessary condition for removing all U symbols successfully.
- 'Effective termination' of CTRSs is preserved by $\mathbb{U}_\mu$ and $\mathbb{T}$. Thus, termination of $\mathbb{T}(\mathbb{U}_\mu(R))$ guarantees 'effective termination' of R. When $\mathbb{T}(\mathbb{U}_\mu(R))$ has no U symbols, termination of $\mathbb{T}(\mathbb{U}_\mu(R))$ coincides with 'effective termination' of R. Therefore, several methods of proving termination of TRSs are applicable for proving 'effective termination' of R.
- Given a conditional rule, the recursive reduction of the conditional part that is not terminating sometimes become terminating. Consider the CTRS $R_7 = \{\ \mathsf{f}(x, y) \to z \Leftarrow \mathsf{g}(x) \to z,\ \mathsf{a} \to \mathsf{g}(\mathsf{a})\ \}$. This CTRS R_7 is transformed by $\mathbb{T}(\mathbb{U}_\mu(\cdot))$ into $R'_7 = \{\ \mathsf{f}(x, y) \to \mathsf{g}(x),\ \mathsf{a} \to \mathsf{g}(\mathsf{a})\ \}$. When $\mathsf{f}(x, y) \to z \Leftarrow \mathsf{g}(x) \to z$ is applied to $\mathsf{f}(\mathsf{a}, \mathsf{a})$, the recursive reduction of the instantiated condition $\mathsf{g}(\mathsf{a})$ does not terminate. On the other hand, in the case of applying $\mathsf{f}(x, y) \to \mathsf{g}(x)$, the conditional part is no longer concerned, that is, the reduction of the condition does terminate.
- It is clear that all CS-TRSs in the process of $\mathbb{T}$ can be considered as the unraveled systems for some CTRSs. For example, R_1 corresponds to the conditional rule $\mathsf{f}(x, y) \to z \Leftarrow \mathsf{g}(y) \to z \wedge \mathsf{h}(\mathsf{g}(x), x) \to z$.

As another approach to CTRSs, Viry proposed the transformation of normal or join CTRSs into TRSs [14]. Unlike unravelings, his transformation does not introduce U symbols but extends the arity of defined symbols. Similarly to unravelings, his transformation is not simulation-complete for all CTRSs. The example in [6] is also a counterexample against simulation-completeness of his transformation. Antoy, Brassel, and Hanus applied Viry's transformation to conditional narrowing of constructor-based CTRSs that are restricted normal CTRSs. [1]. Rosu proposed the transformation of join CTRSs for implementing an efficient conditional rewriting engine [12]. His transformation seems to produce unconditional systems that are simulation-complete. However, the main part to evaluate conditional parts is not defined by rewrite rules but implemented. Thus, his transformation is not suitable for analyzing ultra-properties of CTRSs. Moreover, neither of Viry's and Rosu's transformations are applicable to deterministic 3-CTRSs.

Acknowledgement. This work is partly supported by MEXT. KAKENHI #17700009 and #18500011. We thank Professor Yoshihito Toyama for pointing out some interesting issues about confluence.

References

1. Antoy, S., Brassel, B., Hanus, M.: Conditional narrowing without conditions. In: Proceedings of the 5th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP'03), ACM (2003) 20–31
2. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press (1998)
3. Bergstra, J.A., Klop, J.W.: Conditional rewrite rules: Confluence and termination. Journal of Computer and System Sciences **32** (1986) 323–362
4. Durán, F., Lucas, S., Meseguer, J., Marché, C., Urbain, X.: Proving termination of membership equational programs. In Heintze, N., Sestoft, P., eds.: Proceedings of the 2004 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-based Program Manipulation (PEPM), ACM (2004) 147–158
5. Lucas, S.: Context-sensitive computations in functional and functional logic programs. Journal of Functional and Logic Programming **1998** (1998)
6. Marchiori, M.: Unravelings and ultra-properties. In: Proceedings of the 5th International Conference on Algebraic and Logic Programming (ALP'96). Volume 1139 of Lecture Notes in Computer Science, Springer (1996) 107–121
7. Nishida, N., Sakai, M., Sakabe, T.: On simulation-completeness of unraveling for conditional term rewriting systems. IEICE Technical Report SS2004-18, the Institute of Electronics, Information and Communication Engineers (IEICE) (2004) Vol. 104, No. 243, pp. 25–30.
8. Nishida, N., Sakai, M., Sakabe, T.: Partial inversion of constructor term rewriting systems. In: Proceedings of the 16th International Conference on Rewriting Techniques and Applications. Volume 3467 of Lecture Notes in Computer Science, Springer (2005) 264–278
9. Nishida, N., Sakai, M., Sakabe, T.: Generation of inverse computation programs of constructor term rewriting systems. The IEICE Transactions on Information and Systems **J88-D-I** (2005) 1171–1183 (in Japanese).
10. Ohlebusch, E.: Termination of logic programs: Transformational methods revisited. Applicable Algebra in Engineering, Communication and Computing **12** (2001) 73–116
11. Ohlebusch, E.: Advanced Topics in Term Rewriting. Springer (2002)
12. Rosu, G.: From conditional to unconditional rewriting. In: Revised selected papers of the 17th International Workshop on Recent Trends in Algebraic Development Techniques (WADT 2004). Volume 3423 of Lecture Notes in Computer Science, Springer (2005) 218–233
13. Toyama, Y.: Confluent term rewriting systems with membership conditions. In: Proceedings of the 1st International Workshop on Conditional Term Rewriting Systems (CTRS). Volume 308 of Lecture Notes in Computer Science, Springer (1987) 228–241
14. Viry, P.: Elimination of conditions. Journal of Symbolic Computation **28**(3) (1999) 381–401