

Transformation for Refining Unraveled Conditional Term Rewriting Systems

Naoki Nishida¹ Tomohiro Mizutani² Masahiko Sakai³

Graduate School of Information Science
Nagoya University
Nagoya, Japan

Abstract

Unravelings, transformations from conditional term rewriting systems (CTRSs, for short) into unconditional term rewriting systems, are valuable for analyzing properties of CTRSs. In order to completely simulate rewrite sequences of CTRSs, the restriction by a particular context-sensitive and membership condition that is determined by extra function symbols introduced due to the unravelings, must be imposed on the rewrite relations of the unraveled CTRSs. In this paper, in order to weaken the context-sensitive and membership condition, we propose a transformation applied to the unraveled CTRSs, that reduces the number of the extra symbols. In the transformation, updating the context-sensitive condition properly, we remove the extra symbols that satisfy a certain condition. If the transformation succeeds in removing all of the extra symbols, we obtain the TRSs that are computationally equivalent with the original CTRSs.

Keywords: unraveling, context-sensitive reduction, membership constraint, program transformation

1 Introduction

Unravelings are transformations from conditional term rewriting systems (CTRSs, for short) into unconditional term rewriting systems (TRSs, for short) [7]. They are useful for analyzing properties of CTRSs. For example, *operational termination* [6]⁴ is an important property of CTRSs and it can be guaranteed by termination of the unraveled CTRSs [6,7,13]. The first transformation from CTRSs into TRSs was investigated for normal CTRSs by J.A. Bergstra and J.W. Klop [3]. This concept was revisited as an *unraveling* by M. Marchiori who discussed its properties such as syntactic ones, termination, modularity, and so on [7]. He also proposed an

¹ Email: nishida@is.nagoya-u.ac.jp

² Email: mizutani@sakabe.i.is.nagoya-u.ac.jp

³ Email: sakai@is.nagoya-u.ac.jp

⁴ Intuitively, a terminating CTRS is *operationally terminating* if all reductions that are induced by conditional parts and that are needed for deciding whether given terms are reducible, also terminate. It was shown in [6] that *operational termination* is more practical and suitable notion for CTRS termination than “effective termination [7,13]”.

$$\begin{array}{c}
\rho : l \rightarrow r \Leftarrow s_1 \rightarrow t_1 \wedge \cdots \wedge s_k \rightarrow t_k \\
\Downarrow \mathbb{U} \\
\left\{ \begin{array}{l} l \rightarrow u_1^\rho(s_1, \vec{x}_1), \\ u_1^\rho(t_1, \vec{x}_1) \rightarrow u_2^\rho(s_2, \vec{x}_2), \\ \quad \quad \quad \cdots \quad \quad \quad u_k^\rho(t_k, \vec{x}_k) \rightarrow r \end{array} \right\}
\end{array}$$

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

unraveling for join CTRSs. E. Ohlebusch proposed an unraveling for deterministic 3-CTRSs to prove termination of logic programs [12]. A variant of Ohlebusch’s unraveling is used in several papers [4,9,10,11].

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. Unravelings appear to be attractive in order to compute reduction sequences of CTRSs. However, in general, they do not maintain either of the two important properties, irreducibility of normal forms of CTRSs (invariance of normal forms) or *simulation-completeness*. Note that normal forms of all CTRSs are not well-defined. Here, 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 [9,10,11]. Note that *simulation-completeness* in this paper coincides with “soundness and completeness” in [15] but slightly different from *computational equivalence* [15]. Intuitively, *computational equivalence* can be considered as “simulation-completeness and invariance of normal forms”. In general, unravelings are not simulation-complete for arbitrary target CTRSs because the unraveled CTRSs are over approximations of the original CTRSs [7,13]. However, it was shown that the restriction by a particular *context-sensitive and membership condition* to the rewrite relations of the unraveled CTRSs makes the unraveling for deterministic CTRSs preserve unreachability of the original CTRSs, that is, simulation-completeness of the unraveling with respect to the particular context-sensitive and membership reduction [11].

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). Let ρ be a conditional rewrite rule $l \rightarrow r \Leftarrow s_1 \rightarrow t_1 \wedge \cdots \wedge s_k \rightarrow t_k$ in a CTRS R and σ a substitution. The reduction step from $l\sigma$ to $r\sigma$ by ρ is simulated by a reduction sequence constructed from the corresponding unconditional rules as follows:

$$\begin{array}{l}
l\sigma \rightarrow_{\mathbb{U}(R)} u_1^\rho(s_1, \vec{x}_1)\sigma \xrightarrow{*}_{\mathbb{U}(R)} u_1^\rho(t_1, \vec{x}_1)\sigma \\
\rightarrow_{\mathbb{U}(R)} u_2^\rho(s_2, \vec{x}_2)\sigma \xrightarrow{*}_{\mathbb{U}(R)} \cdots \xrightarrow{*}_{\mathbb{U}(R)} u_k^\rho(t_k, \vec{x}_k)\sigma \rightarrow_{\mathbb{U}(R)} r\sigma.
\end{array}$$

The sequence starts from $l\sigma$. In the sequence, each extra function symbol u_i^ρ , called a *U symbol*, sequentially checks reachability from $s_i\sigma$ to $t_i\sigma$ (in other words, u_i^ρ evaluates the condition $s_i \rightarrow t_i$ with σ). The sequence ends at $r\sigma$ after all conditions are evaluated successfully, that is, $s_i\sigma \xrightarrow{*}_R t_i\sigma$.

We are sure that unravelings preserve reachability on terms over the original signatures. On the other hand, as described above, the unravelings do not preserve

$$\left(\left\{ \begin{array}{l} l \rightarrow \mathbf{u}_j^\rho(t_1\delta, \dots, t_m\delta), \\ \mathbf{u}_j^\rho(t_1, \dots, t_m) \rightarrow r \end{array} \right\} \cup S, \mu_i \right) \xRightarrow{\mathbb{T}} (\{ l \rightarrow r\delta \} \cup S, \mu_{i+1})$$

where \mathbf{u}_j^ρ is *removable* (satisfies the condition RMC described in Subsection 4.2) and the context-sensitive condition μ_i is updated to μ_{i+1} with respect to $\text{root}(r)$.

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

unreachability for all CTRSs because unexpected rewrite sequences are sometimes caused by disobeying the application order of rules whose left-hand sides are rooted with U symbols [7,13]. In order to avoid this, a restriction to the rewrite relations of the unraveled CTRSs is required, which prohibits reducing the following redexes:

- (CONTEXT-SENSITIVE) ones that occur strictly below U symbols, except for the first arguments of the U symbols (for instance, redexes in $\vec{x}_i\sigma$), or
- (MEMBERSHIP) ones that contain a U symbol in their proper subterms.

In this way, the restriction by the above context-sensitive and membership condition is imposed on the rewrite relations of the unraveled CTRSs to maintain simulation-completeness [11].

In this paper, we try to construct unconditional TRSs (from the unraveled CTRSs) that are simulation-complete for the corresponding CTRSs without the context-sensitive and membership condition. We first improve the unraveling for deterministic CTRSs slightly so that the number of unraveled rules is less than those with the ordinary unraveling. This improvement is done by grouping as many conditions as possible, that can be evaluated in parallel. We then propose a transformation that is applied to the unraveled CTRSs, in order to remove as many U symbols as possible from the unraveled CTRSs. Although the improvement is not so a novelty, there are some cases where it helps the transformation to reduce the number of U symbols (see Section 4). Each step of the transformation we propose is based on the “composition” of two rules, and it folds two rules used successively in reduction sequences into one rule (see Fig. 2). We show a certain delicate condition (RMC in Subsection 4.2) that U symbols to be removed should satisfy to maintain simulation-completeness, and we tighten it to keep an advantage of CTRSs associated with the **let** structure of functional programs (RMC' in Section 6). The “composition” is a rather trivial procedure, but removing U symbols leads to the relaxation of the restriction by the context-sensitive and membership condition 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. We also show that the transformation preserves confluence of CTRSs modulo the reduction strategy by the context-sensitive and membership conditions, on terms over the original signatures.

The transformation does not preserve non-“operational-termination” in all cases where non-“operational termination” means “non-termination or termination without operational termination”. This disadvantage makes it impossible to prove operational termination of CTRSs by termination of the corresponding unraveled CTRSs. In order to remove this disadvantage, we need to use a tightened RMC, that is, RMC''

in Section 7.

Unfortunately, not all U symbols can be removed, that is, the transformation sometimes fails in removing all U symbols. Nevertheless, we have some advantages even if not all U symbols are removed.

- The context-sensitive conditions are sometimes removed.
- The transformation helps us to simplify conditional rules.

If we succeed in removing all U symbols, there are the following further advantages.

- The context-sensitive and membership conditions disappear.
- Confluence of CTRSs is preserved. Accordingly, in order to prove confluence of the CTRSs, we can use several techniques for proving confluence of TRSs.
- Irreducibility of normal forms of CTRSs is preserved. This leads us to computationally equivalent TRSs with the original CTRSs.

Thus, the transformation we propose is harmless with respect to simulation-completeness, operational termination and non-“operational-termination” of CTRSs whenever it is based on both RMC' and RMC'' . The largest advantage of the transformation is that we obtain computationally equivalent TRSs with the original CTRSs if we succeed in removing all U symbols. Even when not all U symbols can be removed, we can obtain simplified CTRSs that are computationally equivalent with the original CTRSs.

The unraveling for deterministic CTRSs is used in the *inversion compilers* proposed in [10,11]. The compilers transform constructor TRSs into CTRSs that compute (partial) inverses of functions defined in the TRSs. The compilers then unravel the CTRSs into TRSs whose rules may have extra variables. Since terms representing inverse computation of many-to-one functions may have several normal forms as solutions of the computation, the CTRSs that are intermediate results of the compilers are not always confluent. For 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.

As another approach to simulation-completeness, it was shown that the unraveling for normal CTRSs is simulation-complete for all left-linear normal CTRSs [7]. It was also shown that the unraveling for deterministic CTRSs is simulation-complete for CTRSs if the unraveled CTRSs are either left-linear or both right-linear and non-erasing [9]. The approach in [9] is not applicable to all deterministic CTRSs while that in [11] is so.

This paper is organized as follows. In Section 2, we give notations of term rewriting. In Section 3, we introduce a variant of Ohlebusch’s unraveling on which our approach is based, and we give a notion of simulation-completeness and context-sensitive and membership conditions to preserve simulation-completeness. We also improve slightly the unraveling. In Section 4, we propose a transformation that removes extra function symbols introduced due to the improved unraveling from the unraveled CTRSs. In Section 5, we discuss confluence of CTRSs and the unraveled

CTRSs. In Section 6, we tighten the condition for removing the extra function symbols. In Section 7, we offer some concluding remarks and related works.

2 Preliminaries

This paper follows the basic notions of term rewriting [2,13]. 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, \dots, t_n is represented by $\text{Var}(t_1, \dots, t_n)$. The *identity* of terms s and t is denoted by $s \equiv t$. For a term t and a position p of t , the notation $t|_p$ represents the subterm of t at p . The function symbol at the *root position* ε of t is denoted by $\text{root}(t)$. The notation $C[t_1, \dots, t_n]_{p_1, \dots, p_n}$ represents the term obtained by replacing \square 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 $\text{Dom}(\sigma)$ and $\text{Ran}(\sigma)$, respectively. The application $\sigma(t)$ of σ to t is abbreviated to $t\sigma$. 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, c) , denoted by $l \rightarrow r \Leftarrow c$, such that the *left-hand side* l is a non-variable term in $\mathcal{T}(\mathcal{F}, \mathcal{V})$, the *right-hand side* r is a term in $\mathcal{T}(\mathcal{F}, \mathcal{V})$, and the *conditional part* c is in form of $s_1 \rightarrow t_1 \wedge \dots \wedge s_n \rightarrow t_n$ ($n \geq 0$) of terms s_i and t_i in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. In particular, the conditional rewrite rule $l \rightarrow r \Leftarrow c$ is said to be an (*unconditional*) *rewrite rule* if $n = 0$, and we may abbreviate it to $l \rightarrow r$. We say that a binary relation \rightarrow and a substitution σ *satisfy* the conditional part c , written by $c(\sigma, \rightarrow)$, if $s_i\sigma \rightarrow t_i\sigma$ for $1 \leq i \leq n$. We denote $l \rightarrow r \Leftarrow c$ with a unique label ρ by $\rho : l \rightarrow r \Leftarrow c$. To simplify notations, we may write labels instead of the corresponding rules. For a conditional rewrite rule $\rho : l \rightarrow r \Leftarrow c$, variables occurring not in l but in either r or c are called *extra variables* of ρ . The set of all extra variables of ρ is denoted by $\mathcal{EVar}(\rho)$.

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 $\xrightarrow{n+1}_R = \{(C[l\sigma]_p, C[r\sigma]_p) \mid \rho : l \rightarrow r \Leftarrow c \in R, c(\sigma, \xrightarrow{n}_R)\}$. The *rewrite relation* \rightarrow_R of R is defined as $\rightarrow_R = \bigcup_{n \geq 0} \xrightarrow{n}_R$. To specify the position p and the rule ρ , we write $s \xrightarrow{p}_R t$ or $s \xrightarrow{[p, \rho]}_R t$. An (*oriented*) *conditional rewriting system* (CTRS, for short) over a signature \mathcal{F} is an abstract reduction system $(\mathcal{T}(\mathcal{F}, \mathcal{V}), \rightarrow_R)$ 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 $(\mathcal{T}(\mathcal{F}, \mathcal{V}), \rightarrow_R)$. A CTRS is called a *term rewriting system with extra variables* (EV-TRS, for short) if it contains only unconditional rewrite rules. Note that rewrite sequences of EV-TRSs can be simulated by *narrowing* if each extra variable is substituted with a normal form in the rewrite sequences [8]. Specifically, it is a *term rewriting system* (TRS, for short) if $\text{Var}(l) \supseteq \text{Var}(r)$ for every rule $l \rightarrow r$ in it.

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 $\rho : l \rightarrow r \Leftarrow s_1 \rightarrow t_1 \cdots s_k \rightarrow t_k$ is called *deterministic* if $\text{Var}(s_i) \subseteq \text{Var}(l, t_1, \dots, t_{i-1})$ for $1 \leq i \leq k$. A CTRS is called *normal* if every of its rules $l \rightarrow r \Leftarrow s_1 \rightarrow t_1 \wedge \cdots \wedge s_k \rightarrow t_k$ satisfies that t_1, \dots, t_k are ground normal forms of $R_u = \{l \rightarrow r \mid l \rightarrow r \Leftarrow c \in R\}$.

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 natural numbers such that $\mu(f) \subseteq \{1, \dots, n\}$ for n -ary symbols f in \mathcal{F} . When $\mu(f)$ is not defined explicitly, we assume that $\mu(f) = \{1, \dots, n\}$. 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 $\mathcal{O}_\mu(f(t_1, \dots, t_n)) = \{ip \mid f \in \mathcal{F}, i \in \mu(f), p \in \mathcal{O}_\mu(t_i)\}$. The *context-sensitive reduction* of an EV-TRS R with μ is defined as $\rightarrow_{(R, \mu)} = \{(s, t) \mid s \rightarrow_R^p t, p \in \mathcal{O}_\mu(s)\}$. An abstract reduction system $(\mathcal{T}(\mathcal{F}, \mathcal{V}), \rightarrow_{(R, \mu)})$, 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* [16]. For an EV-TRS R , the *membership-conditional reduction* of \rightarrow_R by a *membership condition* $\in T$ (where $T \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$) is defined as $\xrightarrow[\in T]{\rightarrow} R = \{(C[l\sigma]_p, C[r\sigma]_p) \mid l \rightarrow r \in R, \text{Ran}(\sigma) \subseteq T\}$. The membership-conditional reduction for $\rightarrow_{(R, \mu)}$ is defined similarly as $\xrightarrow[\in T]{\rightarrow}_{(R, \mu)}$.

3 Unravelings for Deterministic CTRSs

In this section, we first introduce a variant \mathbb{U}_O of Ohlebusch's unraveling [12] for deterministic CTRSs and give a notion of *simulation-completeness* [9,10,11]. Then, we give *context-sensitive and membership conditions* [11] to preserve simulation-completeness. Finally, we slightly improve the unraveling \mathbb{U}_O . This improvement is based on the unraveling \mathbb{U}_N for normal CTRSs [7] and effective in reducing the number of the unraveled rules by the transformation we will show in Section 4 (see an example at the end of Subsection 4.3).

3.1 A Variant of Ohlebusch's Unraveling and Simulation-Completeness

We first define a variant of Ohlebusch's unraveling [12], which is proposed in [4,9,10,11]. Here, given a finite set T of terms, we denote by \vec{T} the sequence of the elements in T (in some fixed order), and we denote $\bigcup_{t \in T} \text{Var}(t)$ by $\text{Var}(T)$.

Definition 3.1 Let R be a deterministic CTRS over a signature \mathcal{F} . For every conditional rewrite rule $\rho : l \rightarrow r \Leftarrow s_1 \rightarrow t_1 \wedge \cdots \wedge s_k \rightarrow t_k$, let $|\rho|$ denote the number of conditions in ρ (that is, $|\rho| = k$), and we need k “fresh” function symbols $u_1^\rho, \dots, u_k^\rho$, called *U symbols*, in the transformation. Here, the word “fresh” means that $u_i^\rho \notin \mathcal{F}$ for $1 \leq i \leq k$. We transform ρ into a set $\mathbb{U}_O(\rho)$ of $k + 1$ unconditional rewrite rules as follows:

$$\mathbb{U}_O(\rho) = \{l \rightarrow u_1^\rho(s_1, \vec{X_1}), u_1^\rho(t_1, \vec{X_1}) \rightarrow u_2^\rho(s_2, \vec{X_2}), \dots, u_k^\rho(t_k, \vec{X_k}) \rightarrow r\}.$$

where $X_i = \text{Var}(l, t_1, \dots, t_{i-1}) \cap \text{Var}(t_i, s_{i+1}, t_{i+1}, \dots, s_k, t_k, r)$ ⁵ for $1 \leq i \leq k$.

⁵ X_i is the set of variables that appear in either l, t_1, \dots, t_{i-1} and also in either $t_i, s_{i+1}, t_{i+1}, \dots, s_k, t_k$

The system $\mathbb{U}_O(R) = \bigcup_{\rho \in R} \mathbb{U}_O(\rho)$ is an unconditional EV-TRS over the extended signature $\mathcal{F}_{\mathbb{U}_O(R)} = \mathcal{F} \cup \{u_i^\rho \mid \rho \in R, 1 \leq i \leq |\rho|\}$.

Note that $\mathbb{U}_O(R)$ is a TRS if R is a 3-CTRS. The revised point from the original definition is X_i .

Next, we give the notion of *simulation-completeness* that is based on completeness of *ultra-properties* proposed in [7].

Definition 3.2 Let U be a transformation from CTRSs to TRSs, and R be a CTRS over a signature \mathcal{F} .

- U is said to be *R -reachability-preserving* ($\xrightarrow{*}_R$ -preserving, for short) 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 $s \xrightarrow{*}_{U(R)} t$.
- U is *simulation-sound* for R if U preserves 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*⁶ for R if U is $\xrightarrow{*}_R$ -preserving and simulation-sound for 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 $U(R)$.

All proposed unravelings are $\xrightarrow{*}_R$ -preserving for every target CTRS because $\xrightarrow{*}_R$ -preserving is a necessary condition that transformations are “unraveling”s. On the other hand, in general, they are not simulation-sound for all target CTRSs, and hence not simulation-complete. The cause is that the unraveled CTRSs are over approximations of the original CTRSs. In [7], we can find a counterexample against simulation-completeness of \mathbb{U}_N , \mathbb{U}_O and Ohlebusch’s unraveling.

A restriction to the rewrite relations of the unraveled CTRSs for avoiding this difficulty on non-“simulation-completeness” of \mathbb{U}_O was shown in [11], which is done by the following *context-sensitive* and *membership* condition:

- the context-sensitive condition μ for ρ in Definition 3.1 such that $\mu(u_i^\rho) = \{1\}$ for every u_i^ρ , and
- the membership condition “ $\in \mathcal{T}(\mathcal{F}, \mathcal{V})$ ”.

Let the context-sensitive condition μ_R for R be defined as $\mu_R(u_i^\rho) = \mu_\rho(u_i^\rho)$. Then, we denote the CS-TRS $(\mathbb{U}_O(R), \mu_R)$ by $\mathbb{U}_{O\mu}(R)$. We consider $\mathbb{U}_{O\mu}$ as an unraveling from CTRSs to CS-TRSs.

Theorem 3.3 ([11]) *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 $s \xrightarrow[\in \mathcal{T}(\mathcal{F}, \mathcal{V})]{*} \mathbb{U}_{O\mu}(R) t$.*

or r . In the original definition by Ohlebusch, $\overrightarrow{X_i}$ is the variable sequence “ $\overrightarrow{\text{Var}(l)}, \overrightarrow{\text{Var}(t_1)}, \dots, \overrightarrow{\text{Var}(t_{i-1})}$ ”, and there are a redundancy for computation of terms because all variables are remaining in the argument of u_j^ρ even if some of them are not used in the computation of s_{i+1}, \dots, s_k, r .

⁶ The definition of “simulation-completeness” in [9,10,11] is different from that in this paper, and it does not coincide with “completeness” of ultra-properties. More precisely, “simulation-completeness” in [9,10,11] corresponds to simulation-soundness in this paper. However, discussions on simulation-completeness in those papers are essentially equivalent because $\xrightarrow{*}_R$ -preserving holds for all CTRSs.

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

3.2 Improvement of the Unraveling

There is room for improving the unraveling \mathbb{U}_O . We first explain the intuitive idea of our improvement. 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 \mathbb{U} symbols (see Definition 3.1). For example, the conditional rewrite rule

$$\rho_1 : f(x, y) \rightarrow z \Leftarrow g(x) \rightarrow w \wedge g(y) \rightarrow z \wedge h(w, x) \rightarrow z$$

is unraveled into the following four unconditional rewrite rules, introducing \mathbb{U} symbols u_1 , u_2 and u_3 :

$$\mathbb{U}_O(\rho_1) = \begin{cases} f(x, y) \rightarrow u_1(g(x), x, y), & u_1(w, x, y) \rightarrow u_2(g(y), w, x), \\ u_2(z, w, x) \rightarrow u_3(h(w, x), z), & u_3(z, z) \rightarrow z. \end{cases}$$

The application order of these rules in reduction sequences 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 $g(x) \rightarrow w$ and $g(y) \rightarrow z$ can be evaluated in parallel. The reason is that all the variables x, y used in the evaluation already appear in the left-hand side $f(x, y)$ of the conditional rule ρ_1 . From this fact, we can combine u_1 and u_2 into one symbol u'_1 as follows:

$$f(x, y) \rightarrow u'_1(g(x), g(y), x) \quad \text{and} \quad u'_1(w, z, x) \rightarrow u_3(h(w, x), z).$$

Thus, in order 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 [7]. As shown in the proof of Theorem 3.7 later, this improvement is not so a novelty. However, it is sometimes effective for reducing the number of the unraveled rules by the transformation we will show in Section 4 (see an example at the end of Subsection 4.3).

The idea of the above improvement is formalized as follows.

Definition 3.4 Let R be a deterministic CTRS over a signature \mathcal{F} . We consider a conditional rewrite rule

$$\rho : l \rightarrow r \Leftarrow \left(\bigwedge_{j=1}^{m_1} s_{1,j} \rightarrow t_{1,j} \right) \wedge \cdots \wedge \left(\bigwedge_{j=1}^{m_k} s_{k,j} \rightarrow t_{k,j} \right)$$

in R such that $\text{Var}(s_{i,j}) \subseteq \text{Var}(\{l\} \cup T_1 \cup \cdots \cup T_{i-1})$ for all i and j , where $T_i = \{t_{i,1}, \dots, t_{i,m_i}\}$. Note that every deterministic conditional rewrite rule can be expressed like this. For every conditional rewrite rule ρ in the above form, let $|\rho|$ denote the number of groups of conditions in ρ (that is, $|\rho| = k$), we need k \mathbb{U} symbols $u_1^\rho, \dots, u_k^\rho$. We transform ρ into a set $\mathbb{U}(\rho)$ of $k + 1$ unconditional rewrite

rules as follows:

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

where $S_i = \{s_{i,1}, \dots, s_{i,m_i}\}$ and

$$X_i = \text{Var}(\{l\} \cup T_1 \cup \dots \cup T_{i-1}) \cap \text{Var}(T_i \cup S_{i+1} \cup T_{i+1} \cup \dots \cup S_k \cup T_k \cup \{r\})$$

for $1 \leq i \leq k$. The set $\mathbb{U}(R) = \bigcup_{\rho \in R} \mathbb{U}(\rho)$ is an EV-TRS over the extended signature $\mathcal{F}_{\mathbb{U}(R)} = \mathcal{F} \cup \{\mathbf{u}_i^\rho \mid \rho \in R, 1 \leq i \leq |\rho|\}$.

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 , \dots or T_{i-1} , and used in either r , S_{i+1}, \dots, S_k or T_i, \dots, T_k .

In Definition 3.4, 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 $\mathbb{U}_O(\rho)$ upto some renaming of \mathbf{U} symbols if $m_i = 1$ for every i , and it is equal to $\mathbb{U}_N(\rho)$ upto the renaming 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 Definition 3.4 satisfies

$$\text{Var}(s_{i,j}) \not\subseteq \text{Var}(l) \cup \text{Var}(T_1 \cup \dots \cup T_{i-2}) \text{ for } 1 < i \leq k \text{ and } 1 \leq j \leq m_i.$$

Under this assumption, $\mathbb{U}(\rho)$ is determined uniquely.

Example 3.5 The conditional rule ρ_1 is unraveled by \mathbb{U} into

$$\mathbb{U}(\rho_1) = \begin{cases} f(x, y) \rightarrow \mathbf{u}_{1,2}(\mathbf{g}(x), \mathbf{g}(y), x), & \mathbf{u}_{1,2}(w, z, x) \rightarrow \mathbf{u}_3(\mathbf{h}(w, x), z), \\ \mathbf{u}_3(z, z) \rightarrow z. \end{cases}$$

The transformation \mathbb{U} produces three rules while \mathbb{U}_O does four.

Finally, we show that \mathbb{U} preserves simulation-completeness with respect to the context-sensitive membership reduction. Similarly to the case of \mathbb{U}_O , the context-sensitive condition μ_ρ for ρ in Definition 3.4 becomes as follows:

$$\mu_\rho(\mathbf{u}_i^\rho) = \{1, \dots, m_i\} \text{ for every } \mathbf{u}_i^\rho.$$

Let the context-sensitive condition μ_R for R be defined as $\mu_R(\mathbf{u}_i^\rho) = \mu_\rho(\mathbf{u}_i^\rho)$. Then, we denote the CS-TRSs $(\mathbb{U}(\rho), \mu_\rho)$ and $(\mathbb{U}(R), \mu_R)$ and by $\mathbb{U}_\mu(\rho)$ and $\mathbb{U}_\mu(R)$, respectively. Similarly to $\mathbb{U}_{O\mu}$, we can consider \mathbb{U}_μ as an unraveling from CTRSs to CS-TRSs.

Example 3.6 For $\mathbb{U}(\rho_1)$ in Example 3.5, the context-sensitive condition μ_{ρ_1} is specified as $\mu_{\rho_1}(\mathbf{u}_{1,2}) = \{1, 2\}$ and $\mu_{\rho_1}(\mathbf{u}_3) = \{1\}$, and $\mathbb{U}_\mu(\rho_1) = (\mathbb{U}(\rho_1), \mu_{\rho_1})$.

Similarly to other unravelings, \mathbb{U} is not simulation-complete for all CTRSs while \mathbb{U} is $\xrightarrow{*}_R$ -preserving (the example in [7] is also a counterexample against simulation-completeness of \mathbb{U}). However, \mathbb{U}_μ is always simulation-complete for R with respect to $\xrightarrow{\in \mathcal{T}(\mathcal{F}, \mathcal{V})} \mathbb{U}_\mu(R)$.

Theorem 3.7 *Theorem 3.3 also holds for \mathbb{U}_μ .*

Proof. We construct the following rule from ρ in Definition 3.4:

$$\rho' : l \rightarrow r \Leftarrow \bigwedge_{i=1}^k \mathbf{tp}_{m_i}(s_{i,1}, \dots, s_{i,m_i}) \rightarrow \mathbf{tp}_{m_i}(t_{i,1}, \dots, t_{i,m_i})$$

where \mathbf{tp}_j is a fresh constructor not in \mathcal{F} that represents the tuple of j terms t_1, \dots, t_j . It is clear that ρ' is deterministic and satisfies $\mathbb{U}_O(\rho') = \mathbb{U}(\rho')$ and $\mu_{\rho'}(\mathbf{u}_i^{\rho'}) = \{1\}$, and hence $\mathbb{U}_{O_\mu}(\rho') = \mathbb{U}_\mu(\rho')$. Let R' be a CTRS obtained by the above transformation of the rules in R . Then, it is clear that $\rightarrow_R = \rightarrow_{R'}$ and $\xrightarrow{*}_{\mathcal{T}(\mathcal{F}, \mathcal{V})} \mathbb{U}_\mu(R) = \xrightarrow{*}_{\mathcal{T}(\mathcal{F}, \mathcal{V})} \mathbb{U}_{O_\mu}(R')$ on terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. It follows from Theorem 3.3 that $\xrightarrow{*}_{R'} = \xrightarrow{*}_{\mathcal{T}(\mathcal{F}, \mathcal{V})} \mathbb{U}_{O_\mu}(R')$ on terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. Therefore, we have $\xrightarrow{*}_R = \xrightarrow{*}_{R'} = \xrightarrow{*}_{\mathcal{T}(\mathcal{F}, \mathcal{V})} \mathbb{U}_{O_\mu}(R') = \xrightarrow{*}_{\mathcal{T}(\mathcal{F}, \mathcal{V})} \mathbb{U}_\mu(R)$ on terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. \square

The transformation in the above proof is also adequate for reducing the number of \mathbb{U} symbols. However, we proposed \mathbb{U} in order to avoid introducing another extra symbols (such as \mathbf{tp}_j), and to simply describe the transformation proposed later.

Theorem 3.3 and 3.7 say that \mathbb{U}_{O_μ} and \mathbb{U}_μ are simulation-complete unravelings from CTRSs to CS-TRSs.

4 Transformation to Remove \mathbb{U} Symbols

In this section, maintaining simulation-completeness, we propose a transformation to relax the context-sensitive and membership condition of $(\mathbb{U}(R), \mu_R)$. More precisely, the transformation reduces the number of \mathbb{U} symbols in $\mathbb{U}(R)$, updating the context-sensitive condition. This leads to the relaxation of the condition because the condition depends on the existence of \mathbb{U} symbols. Simply speaking, each step of the transformation is based on the “composition” of two rules, and it folds two rules having the same \mathbb{U} symbol into one rule, that is, the replacement of $l \rightarrow l'\delta$ and $l' \rightarrow r$ with $l \rightarrow r\delta$ where $\mathbf{root}(l')$ is a \mathbb{U} symbol \mathbf{u}_j^ρ (see Fig. 2 in Section 1). When all \mathbb{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 \mathbb{U} symbols are still remaining.

4.1 Examples of Removing \mathbb{U} Symbols

We first give two examples showing our intuitive idea of the transformation process, that is, the “composition” appears to be attractive for removing \mathbb{U} symbols.

Simply speaking, the transformation we will propose is based on the following “composition” of two rewrite rules:

$$\text{COMPOSITION} \quad \frac{S \cup \{l_1 \rightarrow C[l_2\sigma], l_2 \rightarrow r_2\}}{S \cup \{l_1 \rightarrow C[r_2\sigma], l_2 \rightarrow r_2\}}.$$

We apply it to two rules that are connected by a \mathbb{U} symbol \mathbf{u}_j^ρ (that is, $C \equiv \square$ and $\mathbf{root}(l_2) = \mathbf{u}_j^\rho$), and we remove the rule $l_2 \rightarrow r_2$ (see Fig. 2). The reason for

removing $l_2 \rightarrow r_2$ is that u_j^ρ does not appear in S or $l_1 \rightarrow C[r_2\sigma]$ (no term over the original signature is reachable to any term containing the U symbol u_j^ρ) and then $l_2 \rightarrow r_2$ is no longer necessary.

Let us consider a conditional rewrite rule

$$\rho_2 : f(x, y) \rightarrow z \Leftarrow g(x) \rightarrow w \wedge f(w, y) \rightarrow z.$$

This is unraveled by \mathbb{U}_μ to $(\mathbb{U}(\rho_2), \mu_{\rho_2})$ where

$$\mathbb{U}(\rho_2) = \{ f(x, y) \rightarrow u_4(g(x), y), \quad u_4(w, y) \rightarrow u_5(f(w, y)), \quad u_5(z) \rightarrow z \}$$

and $\mu_{\rho_2}(u_4) = \mu_{\rho_2}(u_5) = \{1\}$. The first and second rules are used in order as follows:

$$f(x, y)\sigma_1 \rightarrow u_4(g(x), y)\sigma_1 \xrightarrow{*} u_4(w, y)\sigma_2 \rightarrow u_5(f(w, y)\sigma_2) \xrightarrow{*} u_5(z)\sigma_3 \rightarrow z\sigma_3$$

where $y\sigma_1 \equiv y\sigma_2$ by the context-sensitive condition. This reduction sequence can be simulated by the rule $f(x, y) \rightarrow u_5(f(g(x), y))$ as follows:

$$\dots \xrightarrow{*} f(x, y)\sigma_1 \rightarrow u_5(f(g(x), y)\sigma_1) \xrightarrow{*} u_5(f(w, y)\sigma_2) \xrightarrow{*} u_5(z)\sigma_3 \rightarrow z\sigma_3.$$

From this observation, we can replace the first and second rules $f(x, y) \rightarrow u_4(g(x), y)$ and $u_4(w, y) \rightarrow u_5(f(w, y))$ with the rule $f(x, y) \rightarrow u_5(f(g(x), y))$, and hence we have

$$\{ f(x, y) \rightarrow u_5(f(g(x), y)), \quad u_5(z) \rightarrow z \}$$

In a similar fashion, we also remove u_5 as follows:

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

The above set of rules contains 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 3.5. Similarly to the previous example, the first and second rules in $\mathbb{U}(\rho_1)$ are replaced with $f(x, y) \rightarrow u_3(h(g(x), x), g(y))$, and hence we have

$$\{ f(x, y) \rightarrow u_3(h(g(x), x), g(y)), \quad u_3(z, z) \rightarrow z \}.$$

At this time, possible reductions at position 2 of $u_{1,2}(g(x), g(y), x)$ must be done at position 2 of $u_3(h(g(x), x), g(y))$. In order to allow these reductions, the context-sensitive condition μ_{ρ_1} must be updated as $\mu'_{\rho_1}(u_3) = \{1, 2\}$. Since we have only one U symbol u_3 , the context-sensitive condition μ'_{ρ_1} is *ineffective*. Here, given an EV-TRS R , we say that a context-sensitive condition μ is *ineffective* for R if $\mu(f) = \{1, \dots, n\}$ for all n -ary symbols f that may be a U symbol. In this way, by using the above replacements, we succeeded in reducing the number of U symbols from $\mathbb{U}(R)$, removing U symbols and updating the context-sensitive conditions.

Fortunately, in the above two examples, folding rules and updating context-sensitive conditions do not entail problems, that is, simulation-completeness is preserved although we did not concern about it. In the next subsection, we will show a sufficiency of a side condition, named RMC, for the transformation to preserve simulation-completeness.

4.2 Removable U Symbols

In the previous subsection, all replacements preserve simulation-completeness. However, such replacements do not preserve simulation-completeness in all cases. Con-

sider the CTRS $R_1 = \{\rho_3\} \cup R_2$ where

$$\rho_3 : f(x, x') \rightarrow z \Leftarrow g(x) \rightarrow y \wedge g(x') \rightarrow z \wedge g(y) \rightarrow w \wedge h(w, z) \rightarrow z$$

and $R_2 = \{ g(a) \rightarrow b, g(b) \rightarrow c, h(g(x), g(a)) \rightarrow b \}$. The CTRS R_1 is unraveled by \mathbb{U} and transformed by the “composition” into

$$R'_1 = R_2 \cup \left\{ \begin{array}{l} f(x, x') \rightarrow u_6(g(g(x)), g(x')), \quad u_6(w, z) \rightarrow u_7(h(w, z), z), \\ u_7(z, z) \rightarrow z, \end{array} \right.$$

and $\mu_1(u_7) = \{1\}$. Moreover, the CS-TRS (R_3, μ_3) is obtained by applying the “composition” to (R'_1, μ_1) where

$$R_3 = R_2 \cup \{ f(x, x') \rightarrow u_7(h(g(g(x)), g(x')), g(x')), \quad u_7(z, z) \rightarrow z \}$$

and $\mu_3(u_7) = \{1, 2\}$. This CS-TRS (R_3, μ_3) is not simulation-complete for $\mathbb{U}_\mu(R_1)$ because we have $f(a, a) \xrightarrow{*}_{(R_3, \mu_3)} b$ but not $f(a, a) \xrightarrow{*}_{\mathbb{U}_\mu(R_1)} b$. The variable z at position 2 of the term $u_6(h(y, z), z)$ should be used only for delivering value. For this reason, this z should not be instantiated by the “composition” with any term that does not finish being evaluated although z is used in the other position 1.2. This observation brings the following condition RMC to the transformation as a side condition (see Fig. 2):

(RMC) if $\text{root}(r)$ is a U symbol u , then no variable in $\text{Dom}(\delta)$ is shared between terms at active and inactive positions (in $\mu_i(u)$ and not in $\mu_i(u)$)⁷.

In summary, the “composition” preserves simulation-completeness if it is applied to two rewrite rules connected by a U symbol satisfying the above condition RMC. Concerning the upper case in Fig. 2, we say that a U symbol u_i^ρ is *removable from* $\{ l \rightarrow u_j^\rho(t_1\delta, \dots, t_m\delta), u_j^\rho(t_1, \dots, t_m) \rightarrow r \} \cup S$ if RMC is satisfied.

4.3 Transformation for Removing U Symbols

The transformation for removing U symbols is formalized as follows. Given context-sensitive conditions μ and μ' , we write $\mu = \mu'$ if $\mu(f) = \mu'(f)$ for all f .

Definition 4.1 Let ρ be a deterministic conditional rewrite rule over a signature \mathcal{F} . We define pairs (S_i, μ_i) recursively as follows.

- (i) $(S_0, \mu_0) := (\mathbb{U}(\rho), \mu_\rho)$.
- (ii) Select a *removable* U symbol u_j^ρ from

$$S_i = \{ l \rightarrow u_j^\rho(t_1\delta, \dots, t_m\delta), \quad u_j^\rho(t_1, \dots, t_m) \rightarrow r \} \uplus S^8$$

where δ is a substitution, and then let

$$S_{i+1} := \{ l \rightarrow r\delta \} \cup S \quad \text{and} \quad \mu_{i+1} := \mu_i.$$

Moreover, if $\text{root}(r)$ is a U symbol u , then update μ_{i+1} as follows:

$$\mu_{i+1}(u) := \mu_i(u) \cup \{ k \mid 1 \leq k \leq m, r|_k \in \text{Dom}(\delta) \}.$$

⁷ That is, $\text{Dom}(\delta) \cap \left(\bigcup_{k \in \mu_i(u)} \text{Var}(t_k) \right) \cap \left(\bigcup_{k \notin \mu_i(u)} \text{Var}(t_k) \right) = \emptyset$.

⁸ These two rules are the only rules in S_i which contain u_j^ρ .

We 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_ρ s are disjoint.

Proposition 4.2 *At (ii) in Definition 4.1, both of the following hold:*

- if $\text{root}(r)$ is a U symbol \mathbf{u} , then $r|_p$ is a variable for all inactive positions $p \notin \mu_{i+1}(\mathbf{u})$, and
- $t_p \delta \equiv t_p \in \mathcal{V}$ for all inactive positions $p \notin \mu_i(\mathbf{u}_j^\rho)$ ⁹.

Proof (Sketch) It is clear that $\mathbb{U}(\rho)$ and μ_ρ satisfy the first claim. By definition, all inactive positions $p \notin \mu_i(\mathbf{u})$ are added to the set $\mu_{i+1}(\mathbf{u})$ of active positions if variables at the inactive positions are instantiated by δ . From this observation, the first claim holds for all (S_i, μ_i) s.

It follows from the first claim that all terms at inactive positions are variables, that is, $t_p \delta$ in $\mathbf{u}_j^\rho(t_1 \delta, \dots, t_m \delta)$ is a variable if $p \notin \mu_i(\mathbf{u}_j^\rho)$. Therefore, the second claim holds for all (S_i, μ_i) s. \square

The above transformation \mathbb{T} 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 4.3 $\mathbb{U}_\mu(\rho_1)$ is transformed by \mathbb{T} into $\mathbb{T}(\mathbb{U}_\mu(\rho_1)) = (R_4, \mu_{R_4})$ where

$$R_4 = \{ f(x, y) \rightarrow \mathbf{u}_3(\mathbf{h}(\mathbf{g}(x), x), \mathbf{g}(y)), \quad \mathbf{u}_3(z, z) \rightarrow z \}$$

and $\mu_{R_4}(\mathbf{u}_3) = \{1, 2\}$. The membership condition is still necessary for the above system because of the existence of the U symbol \mathbf{u}_3 . On the other hand, the above μ_{R_4} is ineffective for R_4 . Therefore, we succeed in removing the context-sensitive condition, while the membership condition is still necessary.

In general, we have finitely many results of \mathbb{T} from an unraveled CTRS because there are possibly some removable U symbols at the second step in Definition 4.1 and then there are non-deterministic choices for selecting U symbols. For example, there are two results of $\mathbb{T} \circ \mathbb{U}_\mu$ for the conditional rule ρ_3 . The results are unique if the fourth condition $\mathbf{h}(w, z) \rightarrow z$ is replaced with $\mathbf{h}(w, z) \rightarrow v$:

$$\rho'_3 : f(x, x') \rightarrow z \Leftarrow \mathbf{g}(x) \rightarrow y \wedge \mathbf{g}(x') \rightarrow z \wedge \mathbf{g}(y) \rightarrow w \wedge \mathbf{h}(w, z) \rightarrow v.$$

The same is said of $\mathbb{T} \circ \mathbb{U}_{O\mu}$. In this way, the result of \mathbb{T} is not always unique. We can see that the transformation \mathbb{T} does not always succeed in removing all U symbols even if we search all possible results exhaustively. 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. In this paper, in order to determine $\mathbb{T}(\mathbb{U}_\mu(R))$ uniquely, we select the \mathbf{u}_j^ρ at every step of S_i , whose index j is the greatest in all removable U symbols of ρ .

⁹ That is, $\text{Dom}(\delta|_{\text{Var}(t_1, \dots, t_m)}) \subseteq (\bigcup_{k \in \mu_i(\mathbf{u}_j^\rho)} \text{Var}(t_k)) \setminus (\bigcup_{k \notin \mu_i(\mathbf{u}_j^\rho)} \text{Var}(t_k))$

Here we give an example showing that \mathbb{U} helps \mathbb{T} to decrease the number of \mathbb{U} symbols more than \mathbb{U}_O . Consider the rule

$$\begin{aligned} \rho_4 : f(x, x') \rightarrow h(y, w) \Leftarrow g(x) \rightarrow x \wedge g(g(x)) \rightarrow y \\ \wedge g(x') \rightarrow z \wedge h(y, z) \rightarrow w \wedge g(y) \rightarrow b. \end{aligned}$$

There are two results of $\mathbb{T}(\mathbb{U}_O(\rho_4))$ where tree \mathbb{U} symbols are remaining. On the other hand, the result of $\mathbb{T}(\mathbb{U}_\mu(\rho_4))$ is unique where two \mathbb{U} symbol is remaining. This means that the improvement of \mathbb{U}_O in Section 3 is sometimes effective for reducing the number of \mathbb{U} symbols and for obtaining the unique result of \mathcal{T} .

4.4 Simulation-Completeness of Transformation

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

Lemma 4.4 *Let ρ be a conditional rewrite rule in a deterministic CTRS R over a signature \mathcal{F} , (R_i, μ_i) and (R_{i+1}, μ_{i+1}) ($i \geq 0$) be the i -th and $i+1$ -th intermediate CS-TRSs of the transformation process for $\mathbb{T}(\mathbb{U}_\mu(\rho))$, respectively, u_j^ρ be a removable symbol that is removed in the i -th step of the process, $\mathbb{U}_\mu(R \setminus \{\rho\}) = (R', \mu')$, and s and t be terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. 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 proving the *only-if* part is not difficult, we only prove the *if* part “ $s \xrightarrow[\in \mathcal{T}(\mathcal{F}, \mathcal{V})]{*} (R_{i+1} \cup R', \mu_{i+1} \cup \mu') t$ implies $s \xrightarrow[\in \mathcal{T}(\mathcal{F}, \mathcal{V})]{*} (R_i \cup R', \mu_i \cup \mu') t$ ”, by induction on the lexicographic products of the length of the reduction sequences $\xrightarrow[\in \mathcal{T}(\mathcal{F}, \mathcal{V})]{*} (R_{i+1} \cup R', \mu_{i+1} \cup \mu')$ and term structure of s . To simplify the presentation, 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 \rightarrow_i and \rightarrow_{i+1} , respectively.

From the discussion in the proof of Theorem 3.7, we can assume without loss of generality that $u_j^\rho(t_1\delta, \dots, t_m\delta)$ in Definition 4.1 is $u_j^\rho(t_1\delta, \vec{X}_j)$. Variables in t_1 are divided into those in the following three kinds: (i) ones instantiated by δ and remaining at an active position in $\mu_i(u)$, (ii) ones not instantiated by δ and remaining at an active position in $\mu_i(u)$, and (iii) ones erased by the rule of u_j^ρ . From this observation, we can assume without loss of generality the following:

- $R_i \setminus R_{i+1} = \left\{ l \rightarrow u_j^\rho(\underline{f(u, u, u', y)}, z), \right. \\ \left. u_j^\rho(\underline{f(x, x, x', y)}, z) \rightarrow u(\underline{s'}, x, y, z), \quad u(\underline{t'}, x, y, z) \rightarrow r' \right\}$ and
- $R_{i+1} \setminus R_i = \{ l \rightarrow u(\underline{s'\delta}, x\delta, y, z), \quad u(\underline{t'}, x, y, z) \rightarrow r' \}.$

where $\delta = \{x \mapsto u, x' \mapsto u'\}$, $\mu_i(u_j^\rho) = \mu_i(u) = \{1\}$, and $\mu_{i+1}(u) = \{1, 2\}$. It follows from RMC that $x \notin \text{Var}(s')$. We only show the most critical case. Suppose that

$$s \xrightarrow[\rightarrow_{i+1}]{*} l\sigma_1 \rightarrow_{\rightarrow_{i+1}} u(\underline{s'\delta}, x\delta, y, z)\sigma_1 \xrightarrow[\rightarrow_{i+1}]{*} u(\underline{t'}, x, y, z)\sigma_2 \rightarrow_{\rightarrow_{i+1}} r'\sigma_2 \xrightarrow[\rightarrow_{i+1}]{*} t.$$

It follows from the membership condition that σ_1 and σ_2 do not introduce any term having a \mathbb{U} symbol, that is, $\text{Ran}(\sigma_1) \cup \text{Ran}(\sigma_2) \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$. 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

$$s \xrightarrow{*}_i l\sigma_1, \quad s'\delta\sigma_1 \xrightarrow{*}_i t'\sigma_2, \quad x\delta\sigma_1 \xrightarrow{*}_i x\sigma_2, \quad \text{and} \quad r'\sigma_2 \xrightarrow{*}_i t.$$

Considering the corresponding reduction by \rightarrow_i , we have to show that the reduction $u\sigma_1 \equiv x\delta\sigma_1 \xrightarrow{*}_{i+1} x\sigma_2$ that appears in the second argument of u can be simulated in the first argument of u_j^ρ . It follows from $x \notin \text{Var}(s')$ that $s'\delta\sigma_1 \equiv s'\sigma_1$. Let $\theta = \{x \mapsto x\sigma_2, x' \mapsto u'\sigma_1, y \mapsto y\sigma_2, z \mapsto z\sigma_2\}$. Then, we have

$$\begin{aligned} s \xrightarrow{*}_i l\sigma_1 &\rightarrow_i u_j^\rho(\underline{f(u, u, u', y)}, z)\sigma_1 \xrightarrow{*}_i u_j^\rho(\underline{f(x\sigma_2, x\sigma_2, u'\sigma_1, y\sigma_1)}, z\sigma_1) \\ &\equiv u_j^\rho(\underline{f(x, x, x', y)}, z)\theta \rightarrow_i u(\underline{s'}, x, y, z)\theta \equiv u(\underline{s'\sigma_1}, x\sigma_2, y\sigma_2, z\sigma_2) \\ &\xrightarrow{*}_i u(\underline{t'\sigma_2}, x\sigma_2, y\sigma_2, z\sigma_2) \rightarrow_i r'\sigma_2 \xrightarrow{*}_i t. \end{aligned}$$

Similarly to the above case, we can prove the remaining case that r in Definition 4.1 is not rooted with a U symbol. \square

Theorem 4.5 *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 4.4 and Theorems 4.5 and 3.7, the composition $\mathbb{T} \circ \mathbb{U}_\mu$ of the unraveling \mathbb{U} and the transformation \mathbb{T} can be considered as a simulation-complete unraveling from CTRSs to CS-TRSs.

Corollary 4.6 *Theorem 3.3 also holds for $\mathbb{T} \circ \mathbb{U}_\mu$.*

If all U symbols are removed, the reductions by R and $\mathbb{T}(\mathbb{U}_\mu(R))$ are equivalent.

Corollary 4.7 *Let R be a deterministic CTRS over a signature \mathcal{F} , and $(R', \mu') = \mathbb{T}(\mathbb{U}_\mu(R))$. Suppose that R' has no U symbol. Then, $\rightarrow_{(R', \mu')} = \rightarrow_{R'} \subseteq \rightarrow_R \subseteq \xrightarrow{+}_{R'} = \xrightarrow{+}_{(R', \mu')}$, and hence $\xrightarrow{*}_R = \xrightarrow{*}_{(R', \mu')} = \xrightarrow{*}_{R'}$.*

It follows from this corollary that the sets of all normal forms of R and of R' are equivalent, and hence R and R' are computationally equivalent.

Consequently, it can be said that in this case a CTRS R can be represented by a computationally equivalent TRS R' , that is, $\xrightarrow{*}_R = \xrightarrow{*}_{R'}$.

5 On Confluence of CTRSs and the Unraveled CTRSs

In this section, we discuss relationships between confluence of CTRSs and the unraveled CTRSs. For the case of \mathbb{U}_N , it was shown that confluence of left-linear normal CTRSs is preserved by \mathbb{U}_N [7]. Given an abstract reduction system (A, \rightarrow) and a set $T \subseteq A$, we say that (A, \rightarrow) is *confluent on T* if for all $s, t_1, t_2 \in T$, $s \xrightarrow{*} t_1$ and $s \xrightarrow{*} t_2$ implies a $u \in T$ such that $t_1 \xrightarrow{*} u$ and $t_2 \xrightarrow{*} u$.

In general, confluence of CTRSs is not preserved by unravelings with respect to the extended signatures. One of the reasons is that irreducibility of normal forms of CTRSs is not always preserved by the unravelings. Consider a normal form t of a confluent CTRS over a signature. The normal form t may be matched with the left-hand side of a conditional rule $\rho : l \rightarrow r \Leftarrow s_1 \rightarrow t_1$ whose conditional part is not

satisfied by the matching substitution σ where $t \equiv l\sigma$. In this case, however, t becomes reducible on the unraveled CTRS and then t is reduced to $u_1^\rho(s_1, \vec{X}_1)\sigma$. Since the condition $s_1 \rightarrow t_1$ is not satisfied by σ , the term $u_1^\rho(s_1, \vec{X}_1)\sigma$ is not reachable to $r\sigma$ (or even any terms over the original signature), and hence all terms that are reachable from t on the unraveled CTRS contain a U symbol. When the number of normal forms in such terms is more than one, the unraveled CTRS is not confluent. More precisely, in order to cause such a situation, another rule whose left-hand-side is matched with l and whose conditional part is not satisfied by the matching substitution is required. Thus, we can see that terms containing U symbols prevent the unravelings from preserving confluence of CTRSs. From this observation, as far as terms without U symbols are concerned, confluence of CTRSs is preserved by the unravelings if simulation-completeness is preserved. As described in the previous section, the unraveling \mathbb{U}_μ and the transformation \mathbb{T} preserve simulation-completeness, and \mathbb{T} sometimes remove all U symbols. In such cases, confluence of the systems obtained by $\mathbb{T} \circ \mathbb{U}_\mu$ coincides with that of the original CTRSs.

Corollary 5.1 *Let R be a deterministic CTRS R over a signature \mathcal{F} . Then, all of the following are equivalent:*

- R is confluent on $\mathcal{T}(\mathcal{F}, \mathcal{V})$,
- $\mathbb{U}_\mu(R)$ is confluent on $\mathcal{T}(\mathcal{F}, \mathcal{V})$, and
- $\mathbb{T}(\mathbb{U}_\mu(R))$ is confluent on $\mathcal{T}(\mathcal{F}, \mathcal{V})$.

Let $(R', \mu') = \mathbb{T}(\mathbb{U}_\mu(R))$ and suppose that R' has no U symbol. Then, R is confluent on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ 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 are still remaining. However, in order to decide confluence of R on $\mathcal{T}(\mathcal{F}, \mathcal{V})$, we can use ordinary techniques for deciding confluence of $\mathbb{T}(\mathbb{U}_\mu(R))$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ if \mathbb{T} removes all U symbols.

Confluence of the unraveled CTRSs $\mathbb{U}_\mu(R)$ on $\mathcal{T}(\mathcal{F}_{\mathbb{U}(R)}, \mathcal{V})$ cannot guarantee confluence of the original CTRSs R on $\mathcal{T}(\mathcal{F}, \mathcal{V})$. Consider the CTRS

$$R_5 = \{ f(x, y) \rightarrow x \Leftarrow x \rightarrow d, \quad g(x) \rightarrow f(x, a), \quad g(x) \rightarrow f(x, b) \}.$$

This CTRS is not confluent on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ because $g(a)$ is reducible to normal forms $f(a, a)$ and $f(a, b)$. The CTRS R_5 is unraveled to the TRS

$$\mathbb{U}(R_5) = \begin{cases} f(x, y) \rightarrow u_8(x, x), & u_8(d, x) \rightarrow x, \\ g(x) \rightarrow f(x, a), & g(x) \rightarrow f(x, b). \end{cases}$$

The TRS $\mathbb{U}(R_5)$ is terminating and $f(a, a)$ and $f(a, b)$ are joinable to $u_8(a, a)$ on $\mathbb{U}(R_5)$. Hence $\mathbb{U}(R_5)$ is confluent on $\mathcal{T}(\mathcal{F}_{\mathbb{U}(R_5)}, \mathcal{V})$.

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 Revisiting the Condition for Removing U Symbols

It is probably impossible to relax the condition RMC in Subsection 4.2 in maintaining simulation-completeness of the unraveled CTRSs. On 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:

$$R_6 = \begin{cases} \text{twofib}(0) \rightarrow \text{tp}_2(0, \text{s}(0)), \\ \text{twofib}(\text{s}(n)) \rightarrow \text{tp}_2(\#2(m), \text{add}(\#2(m), \#1(m))) \Leftarrow \text{twofib}(n) \rightarrow m, \\ \vdots \end{cases}$$

where $\text{tp}_2(t_1, t_2)$ denotes the pair of two terms t_1 and t_2 . The second rule is unraveled into the system (R_7, μ_{R_7}) where

$$R_7 = \begin{cases} \text{twofib}(\text{s}(n)) \rightarrow \text{u}_9(\text{twofib}(n)), \\ \text{u}_9(m) \rightarrow \text{tp}_2(\#2(m), \text{add}(\#2(m), \#1(m))), \end{cases}$$

and $\mu_{R_7}(\text{u}_9) = \{1\}$. Under the *innermost* reduction strategy, efficiency is still alive in (R_7, μ_{R_7}) . Now, the system (R_7, μ_{R_7}) can be transformed by \mathbb{T} as follows:

$$\text{twofib}(\text{s}(n)) \rightarrow \text{tp}_2(\#2(\text{twofib}(n)), \text{add}(\#2(\text{twofib}(n)), \#1(\text{twofib}(n)))).$$

\mathbb{T} succeeds in removing all U symbols from (R_7, μ_{R_6}) . 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 bundles 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 process of \mathbb{T} , by instantiating variable m in $\text{tp}_2(\#2(m), \text{add}(\#2(m), \#1(m)))$, whose occurrence is non-linear, with `twofib(n)`. In order to prevent such instantiation in these cases, we must tighten the condition RMC as follows:

$$(\text{RMC}') \text{ } r \text{ is linear with respect to } \text{Dom}(\delta).$$

It is clear that RMC' implies RMC . The above 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. In order to keep the advantage of CTRSs relative to efficiency of reductions, \mathbb{T} should be based not only on RMC but also on RMC' even when all U symbols can be removed by \mathbb{T} based on RMC .

7 Concluding Remarks and Related Works

We firstly show an application of our method. Consider the following rule that is one of the rules automatically generated by the inversion compiler in [11] from the TRS computing multiplication:

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

where div and sub compute division and subtraction of natural numbers, respectively, and $\mathsf{tp}_i(t_1, \dots, t_i)$ denotes the tuple of i terms t_1, \dots, t_i . If we can identify $\mathsf{tp}_1(t)$ ¹⁰ with t , the above rule can be transformed into the following rule in the sense of computing division:

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

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

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

Using \mathbb{T} , we succeeded in removing all U symbols from $\mathbb{U}(\rho'_{\text{div}})$, and the above rule coincides with the rewrite rule $\mathsf{s}(x) \div \mathsf{s}(y) \rightarrow \mathsf{s}((x - y) \div \mathsf{s}(y))$ which is a typical definition of division for natural numbers. This justifies the correctness of the program generated by the compiler relative to the handmade program. This example shows that in some cases, the transformation \mathbb{T} may make it possible to obtain a smart unconditional rule that is simulation-complete for the original conditional rule.

Next, we discuss on non-“operational-termination” of CTRSs. Unfortunately, non-“operational-termination” of CTRSs is not always preserved by \mathbb{T} while it is preserved by \mathbb{U}_μ (we will show an example later in extra remarks). If so, we can prove “operational termination” of CTRSs by proving termination of the corresponding TRSs obtained by \mathbb{T} . The reason is that redexes causative of non-“operational-termination” are sometimes erased in the process of \mathbb{T} . In order to avoid this situation, we strengthen RMC again as follows:

(RMC'') *RMC is satisfied and any instantiated variable is not erased, that is, all xs in $\text{Dom}(\delta)$ appear in r .*

It is clear that RMC'' implies RMC . Note that RMC'' and RMC' are incomparable. In order to guarantee “operational termination” of R by termination of $\mathbb{T}(\mathbb{U}_\mu(R))$, we use \mathbb{T} based on RMC'' . When $\mathbb{T}(\mathbb{U}_\mu(R))$ has no U symbols, termination of $\mathbb{T}(\mathbb{U}_\mu(R))$ coincides with “operational termination” of R . Therefore, several methods of proving termination of TRSs are applicable for proving “operational termination” of R .

¹⁰ This identification is natural but $\mathsf{tp}_1(t)$ and t raise different behavior of reductions in some situations.

In summary, the transformation \mathbb{T} is harmless for simulation-completeness (confluence), termination (non-termination) and efficient reductions of CTRSs if it is based on both RMC' and RMC'' . Therefore, properties of TRSs obtained by $\mathbb{T} \circ \mathbb{U}_\mu$ based on both RMC' and RMC'' are useful in discussing those of the original CTRSs.

We offer some extra remarks.

- The results for \mathbb{U}_O in [9] also hold for \mathbb{U} and $\mathbb{T} \circ \mathbb{U}$. More precisely, two syntactic conditions of CTRSs that are sufficient to preserve simulation-completeness without the context-sensitive and membership condition were shown in [9], and they are also sufficient conditions for \mathbb{U} and $\mathbb{T} \circ \mathbb{U}$ to preserve simulation-completeness without context-sensitive and membership conditions. On the other hand, neither of the two conditions are sufficient nor necessary for removing all \mathbb{U} symbols successfully.
- It is clear that each CS-TRS that appears in the process of \mathbb{T} can be considered as the unraveled rules for some conditional rule that may be different from the original conditional rule. For example, the CS-TRS (R_4, μ_{R_4}) corresponds to the following conditional rule:

$$f(x, y) \rightarrow z \Leftarrow g(y) \rightarrow z \wedge h(g(x), x) \rightarrow z.$$

It can be said that the CTRSs obtained by such reverse transformation for \mathbb{U} are computationally equivalent with the corresponding original CTRSs. Therefore, the transformation \mathbb{T} is useful in optimizing conditional rewrite rules.

- For a conditional rule such that the recursive reduction of the conditional part is not (operationally-)terminating, the corresponding reduction in the resulting CS-TRS transformed by $\mathbb{T} \circ \mathbb{U}_\mu$ sometimes becomes terminating while the whole system is still not terminating. For example, consider the following CTRS:

$$R_8 = \{ f(x, y) \rightarrow z \Leftarrow g(x) \rightarrow z, \quad a \rightarrow g(a) \}.$$

This CTRS R_8 is transformed by $\mathbb{T} \circ \mathbb{U}_\mu$ into

$$R'_8 = \{ f(x, y) \rightarrow g(x), \quad a \rightarrow g(a) \}.$$

When $f(x, y) \rightarrow z \Leftarrow g(x) \rightarrow z$ is applied to $f(a, a)$, the recursive reduction of the instantiated condition $g(a) \rightarrow z\{z \mapsto t\}$ does not effectively terminate (hence not operationally terminate) because the number of candidates for t is infinite. On the other hand, in the case of applying $f(x, y) \rightarrow g(x)$, the conditional part disappears and we have no reduction corresponding to the recursive reduction of the condition $g(x) \rightarrow z$. Note that R_8 and R'_8 are not terminating.

- In Section 6, we tightened RMC to RMC' to keep the feature of **let** structures that are often used in functional programming to obtain efficient computation. In order to test the effect of RMC' , we must implement a rewriting engine for unravelings with context-sensitive and membership conditions.

As another approach to remove conditional parts of conditional rules, P. Viry proposed a transformation from join CTRSs into TRSs, that is also applicable to normal CTRSs [17]. Unlike unravelings, his transformation does not introduce \mathbb{U} symbols but extends the arity of defined symbols. Similarly to unravelings, his transformation is not simulation-complete for all CTRSs in general. The example

in [7] is also a counterexample against simulation-completeness of his transformation. Note that Viry's transformation is not applicable to deterministic 3-CTRSs. S. Antoy, B. Brassel, and M. Hanus applied Viry's transformation to conditional narrowing of constructor-based CTRSs that are restricted normal CTRSs [1]. G. Rosu proposed a transformation of join CTRSs for implementing an efficient conditional rewriting engine [14], that is based on Viry's one. 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. As a revised variant of Rosu's transformation, T.-F. Serbanuta and Rosu enhanced Rosu's transformation to become simulation-complete for deterministic 3-CTRSs that are transformed into either left-linear or ground-confluent TRSs [15]. In contrast with unravelings, one of the important advantages of these transformations is that they preserve irreducibility of normal forms of CTRSs. From this advantage, these transformations (especially Serbanuta and Rosu's one) are much suitable for normalizing computation than unravelings. On the other hand, rules obtained by these transformations are a bit more complicated than those by unravelings. From this observation, unravelings are more suitable for analyzing CTRS properties or for optimizing conditional rules. Moreover, the conditions imposed on CTRSs to maintain computational equivalence appears to be tight for practical and complicated examples (those conditions are similar to conditions [9] for unravelings to preserve simulation-completeness without the strategy). We believe that the context-sensitive and membership conditions for unravelings makes the above transformations preserve computational equivalence. On the contrary, the homomorphisms used in these transformations will makes Ohlebusch's unraveling and its variants preserve computational equivalence if X_i in Definition 3.4 is defined as $\text{Var}(l) \cup (\text{Var}(\cup T_1 \cup \dots \cup T_{i-1}) \cap \text{Var}(T_i \cup S_{i+1} \cup T_{i+1} \cup \dots \cup S_k \cup T_k \cup \{r\}))$.

In summary, the transformation proposed in [15] is more suitable for normalizing computation but not for analyzing CTRS properties or optimizing conditional rules than unravelings. The transformation \mathbb{T} in this paper is usable for optimizing conditional rules because each set of unconditional rules obtained by it from a conditional rule has a corresponding simplified conditional rule. From this observation, using \mathbb{T} as a preprocessor of the transformation in [15] is much more effective for those two transformations. Furthermore, Ohlebusch's unraveling with the context-sensitive and membership conditions and Serbanuta and Rosu's transformation should be combined as a method because they are very close now.

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. We would like to thank the anonymous referees for several comments that are useful to make the paper more precise. We are deeply grateful to Salvador Lucas for his valuable comments helping us to improve the final version of the paper.

References

- [1] Antoy, S., B. Brassel and M. Hanus, *Conditional narrowing without conditions*, in: *Proceedings of the 5th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming* (2003), pp. 20–31.
- [2] Baader, F. and T. Nipkow, “Term Rewriting and All That,” Cambridge University Press, United Kingdom, 1998.
- [3] Bergstra, J. A. and J. W. Klop, *Conditional rewrite rules: Confluence and termination.*, Journal of Computer and System Sciences **32** (1986), pp. 323–362.
- [4] Durán, F., S. Lucas, J. Meseguer, C. Marché and X. Urbain, *Proving termination of membership equational programs*, in: *Proceedings of the 2004 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-based Program Manipulation* (2004), pp. 147–158.
- [5] Lucas, S., *Context-sensitive computations in functional and functional logic programs*, Journal of Functional and Logic Programming **1998** (1998).
- [6] Lucas, S., C. Marché and J. Meseguer, *Operational termination of conditional term rewriting systems*, Information Processing Letters **95** (2005), pp. 446–453.
- [7] Marchiori, M., *Unravelings and ultra-properties*, in: *Proceedings of the 5th International Conference on Algebraic and Logic Programming*, Lecture Notes in Computer Science **1139** (1996), pp. 107–121.
- [8] Nishida, N., M. Sakai and T. Sakabe, *Narrowing-based simulation of term rewriting systems with extra variables and its termination proof*, Electronic Notes in Theoretical Computer Science **86** (2003), issue 3, pp. 1–18.
- [9] Nishida, N., M. Sakai and T. Sakabe, *On simulation-completeness of unraveling for conditional term rewriting systems*, IEICE Technical Report SS2004-18, the Institute of Electronics, Information and Communication Engineers (2004), vol. 104, No. 243, pp. 25–30.
- [10] Nishida, N., M. Sakai and T. Sakabe, *Generation of inverse computation programs of constructor term rewriting systems*, The IEICE Transactions on Information and Systems **J88-D-I** (2005), pp. 1171–1183 (in Japanese).
- [11] Nishida, N., M. Sakai and T. Sakabe, *Partial inversion of constructor term rewriting systems*, in: *Proceedings of the 16th International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science **3467** (2005), pp. 264–278.
- [12] Ohlebusch, E., *Termination of logic programs: Transformational methods revisited*, Applicable Algebra in Engineering, Communication and Computing **12** (2001), pp. 73–116.
- [13] Ohlebusch, E., “Advanced Topics in Term Rewriting,” Springer, 2002.
- [14] Rosu, G., *From conditional to unconditional rewriting*, in: *Revised selected papers of the 17th International Workshop on Recent Trends in Algebraic Development Techniques*, Lecture Notes in Computer Science **3423** (2005), pp. 218–233.
- [15] Serbanuta, T.-F. and G. Rosu, *Computationally equivalent elimination of conditions*, in: *Proceedings of the 17th International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science **4098** (2006), pp. 19–34.
- [16] Toyama, Y., *Confluent term rewriting systems with membership conditions*, in: *Proceedings of the 1st International Workshop on Conditional Term Rewriting Systems*, Lecture Notes in Computer Science **308** (1987), pp. 228–241.
- [17] Viry, P., *Elimination of conditions*, Journal of Symbolic Computation **28** (1999), pp. 381–401.