

On Proving Confluence of Conditional Term Rewriting Systems via the Computationally Equivalent Transformation*

Naoki Nishida¹, Makishi Yanagisawa¹ and Karl Gmeiner²

¹ Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 4648603, Japan

nishida@is.nagoya-u.ac.jp makishi@trs.cm.is.nagoya-u.ac.jp

² Institute of Computer Science, UAS Technikum Wien
gmeiner@technikum-wien.at

Abstract

This paper improves the existing criterion for proving confluence of a normal conditional term rewriting system (CTRS) via the Şerbănuţă-Roşu transformation, a computationally equivalent transformation of CTRSs into unconditional term rewriting systems (TRS), showing that a weakly left-linear normal CTRS is confluent if the transformed TRS is confluent. Then, we discuss usefulness of the optimization of the Şerbănuţă-Roşu transformation, which has informally been proposed in the literature.

1 Introduction

Conditional term rewriting is known to be much more complicated than unconditional term rewriting in the sense of analyzing properties, e.g., operational termination [8], confluence [16], reachability [4], and so on. A popular approach to the analysis of conditional term rewriting systems (CTRS) is to transform a CTRS into an unconditional term rewriting system (TRS) that is an overapproximation of the CTRS in terms of reduction. This approach enables us to use techniques for the analysis of TRSs, which are well investigated in the literature. For example, if the transformed TRS is terminating then the CTRS is operationally terminating [3].

There are two approaches to such transformations: *unravelings* [9, 10] proposed by Marchiori (see, e.g., [5, 11]), and a transformation [17] proposed by Viry (see, e.g., [14, 5]). The latest transformation based on Viry's approach is a *computationally equivalent* transformation proposed by Şerbănuţă and Roşu [14, 15], called *the SR transformation*. This converts a left-linear confluent normal CTRS into a TRS which is computationally equivalent to the CTRS. This means that the converted TRS can be used to exactly simulate reduction sequences of the CTRS to normal forms — there is no reduction to from possible initial terms to normal forms, which does not hold on the original CTRS. Another interesting use of the SR transformation is to prove confluence of a left-linear normal CTRS: if the converted TRS is confluent *on reachable terms*, then the CTRS is confluent [14]. However, as far as we know, there are no formal method to prove confluence on reachable terms.

In this paper, we revisit the SR transformation from the viewpoint of proving confluence of CTRSs, especially normal CTRSs. First, we improve the existing criterion [14] for proving confluence of normal CTRSs via the SR transformation, showing that a *weakly left-linear* normal CTRS is confluent if the transformed TRS is confluent on reachable terms. Then, by an example, we show uselessness of the improved criterion for the case that we attempt to use confluence on

*The research in this paper is partly supported by the Austrian Science Fund (FWF) international project I963 and the Japan Society for the Promotion of Science (JSPS).

arbitrary terms instead of confluence on reachable terms. Finally, we discuss usefulness of the optimization of the SR transformation, which has informally been proposed in the literature [14].

2 The SR transformation for Normal CTRSs

In this section, we briefly recall the SR transformation [14] for normal CTRSs. For the sake of readability, as in [14], we restrict our interest to normal 1-CTRSs, any rule of which has *at most one condition*.

This paper assumes familiarity of readers with the basic notions and notations of term rewriting [2, 13]. We only give the definition of *normal CTRSs*. Throughout the paper, we use \mathcal{V} as a countably infinite set of *variables*. 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 either an empty sequence or a pair $s \rightarrow t$ of terms s and t over \mathcal{F} . We may abbreviate it to $l \rightarrow r$ if the conditional part is the empty sequence. A *conditional term rewriting system* (CTRS) is a finite set \mathcal{R} of conditional rewrite rules, and it is called a *normal 1-CTRS* if, for every rule $l \rightarrow r \Leftarrow s \rightarrow t \in \mathcal{R}$, $\text{Var}(l) \supseteq \text{Var}(r) \cup \text{Var}(s)$ and the term t is a ground normal form w.r.t. the *underlying unconditional system* $\mathcal{R}_u = \{l \rightarrow r \mid l \rightarrow r \Leftarrow c \in \mathcal{R}\}$. The sets of *defined symbols* and *constructors* of \mathcal{R} are denoted by $\mathcal{D}_{\mathcal{R}}$ and $\mathcal{C}_{\mathcal{R}}$, respectively.

In the following, the word “conditional rule” is used for representing rules having exactly one condition. We often denote a sequence o_i, o_{i+1}, \dots, o_j of objects by $\mathbf{o}_{i..j}$. Moreover, for the application of a mapping op to $\mathbf{o}_{i..j}$, we denote $op(o_i), \dots, op(o_j)$ by $op(\mathbf{o}_{i..j})$, e.g., for a sequence $\mathbf{t}_{i..j}$ of terms and a substitution θ , we denote $t_i\theta, \dots, t_j\theta$ by $\theta(\mathbf{t}_{i..j})$.

Before transforming a CTRS \mathcal{R} , we extend the signature of \mathcal{R} as follows: we leave constructors of \mathcal{R} without any change; the arity of an n -ary defined symbol f is expanded to $n + m$ where f has m conditional rules in \mathcal{R} , and we replace f by \bar{f} with the arity $n + m$; a fresh constant \perp and a fresh unary symbol $\langle \cdot \rangle$ are introduced. We denote the extended signature by $\bar{\mathcal{F}}$: $\bar{\mathcal{F}} = \{c \mid c \in \mathcal{C}_{\mathcal{R}}\} \cup \{\bar{f} \mid f \in \mathcal{D}_{\mathcal{R}}\} \cup \{\perp, \langle \cdot \rangle\}$. We introduce the mapping $\text{ext}(\cdot)$ to extend the arguments of defined symbols by applying to terms inductively as follows: $\text{ext}(x) = x$ for $x \in \mathcal{V}$; $\text{ext}(c(\mathbf{t}_{1..n})) = c(\text{ext}(\mathbf{t}_{1..n}))$ for $c/n \in \mathcal{C}_{\mathcal{R}}$; $\text{ext}(f(\mathbf{t}_{1..n})) = \bar{f}(\text{ext}(\mathbf{t}_{1..n}), \mathbf{z}_{1..m})$ for $f/n \in \mathcal{D}_{\mathcal{R}}$ where $\text{arity}_{\bar{\mathcal{F}}}(\bar{f}) = n + m$ and z_1, \dots, z_m are fresh variables. The expanded arguments of \bar{f} are used for evaluating the corresponding conditions, and the fresh constant \perp is introduced to the expanded arguments of defined symbols, which does not store any evaluation. We define a mapping $\bar{\cdot}$ from $T(\mathcal{F}, \mathcal{V})$ to $T(\bar{\mathcal{F}}, \mathcal{V})$, which extends the arguments of defined symbols and puts \perp to all the expanded arguments by applying to terms inductively as follows: $\bar{x} = x$ for $x \in \mathcal{V}$; $\bar{c}(\mathbf{t}_{1..n}) = c(\bar{\mathbf{t}}_{1..n})$ for $c \in \mathcal{C}_{\mathcal{R}}$; $\bar{f}(\mathbf{t}_{1..n}) = \bar{f}(\bar{\mathbf{t}}_{1..n}, \perp, \dots, \perp)$ for $f \in \mathcal{D}_{\mathcal{R}}$.

The SR transformation [14] is formally defined as follows.

Definition 1 (SR). *Let $f/n \in \mathcal{D}_{\mathcal{R}}$ that has m conditional rules in \mathcal{R} . Then, $\text{SR}(f(\mathbf{w}_{1..n}) \rightarrow r) = \{ \bar{f}(\text{ext}(\mathbf{w}_{1..n}), \mathbf{z}_{1..m}) \rightarrow \langle \bar{r} \rangle \}$ and, for the i -th conditional rule of f ,*

$$\text{SR}(f(\mathbf{w}_{1..n}) \rightarrow r_i \Leftarrow s_i \rightarrow t_i) = \left\{ \begin{array}{l} \bar{f}(\mathbf{w}'_{1..n}, \mathbf{z}_{1..i-1}, \perp, \mathbf{z}_{i+1..m}) \rightarrow \bar{f}(\mathbf{w}'_{1..n}, \mathbf{z}_{1..i-1}, \langle s_i \rangle, \mathbf{z}_{i+1..m}), \\ \bar{f}(\mathbf{w}'_{1..n}, \mathbf{z}_{1..i-1}, \langle t_i \rangle, \mathbf{z}_{i+1..m}) \rightarrow \langle \bar{r}_i \rangle \end{array} \right\}$$

where z_1, \dots, z_m are fresh variables, and $w'_j = \text{ext}(w_j)$ for all $1 \leq j \leq n$. The set of auxiliary rules is defined as follows:

$$\mathcal{R}_{aux} = \{ \langle \langle x \rangle \rangle \rightarrow \langle x \rangle \} \cup \{ c(\mathbf{x}_{1..i-1}, \langle x_i \rangle, \mathbf{x}_{i+1..n}) \rightarrow \langle c(\mathbf{x}_{1..n}) \rangle \mid c/n \in \mathcal{C}_{\mathcal{R}}, 1 \leq i \leq n \} \\ \cup \{ \bar{f}(\mathbf{x}_{1..i-1}, \langle x_i \rangle, \mathbf{x}_{i+1..n}, \mathbf{z}_{1..m}) \rightarrow \langle \bar{f}(\mathbf{x}_{1..n}, \perp, \dots, \perp) \rangle \mid f/n \in \mathcal{D}_{\mathcal{R}}, 1 \leq i \leq n \}$$

where z_1, \dots, z_m are fresh variables. The transformation SR is defined as follows: $\text{SR}(\mathcal{R}) = \bigcup_{\rho \in \mathcal{R}} \text{SR}(\rho) \cup \mathcal{R}_{aux}$. Note that $\text{SR}(\mathcal{R})$ is a TRS over $\bar{\mathcal{F}}$. The backtranslation mapping $\hat{\cdot}$ for $\bar{\cdot}$ is

defined as follows: $\widehat{x} = x$ for $x \in \mathcal{V}$; $\widehat{c(\mathbf{t}'_{1..n})} = c(\widehat{\mathbf{t}'_{1..n}})$ for $c \in \mathcal{C}_{\mathcal{R}}$; $\widehat{f(\mathbf{t}'_{1..n}, \mathbf{u}_{1..m})} = f(\widehat{\mathbf{t}'_{1..n}})$ for $f \in \mathcal{D}_{\mathcal{R}}$; $\widehat{\langle t' \rangle} = \langle \widehat{t'} \rangle$. A term t' in $T(\overline{\mathcal{F}}, \mathcal{V})$ is called *reachable* if there exists a term $s \in T(\mathcal{F}, \mathcal{V})$ such that $\langle \overline{s} \rangle \rightarrow_{\mathbb{S}\mathbb{R}(\mathcal{R})}^* t'$. We say that $\mathbb{S}\mathbb{R}$ (and also $\mathbb{S}\mathbb{R}(\mathcal{R})$) is *sound* for (reduction of) \mathcal{R} if, for all terms s in $T(\mathcal{F}, \mathcal{V})$ and terms $t' \in T(\overline{\mathcal{F}}, \mathcal{V})$, $\langle \overline{s} \rangle \rightarrow_{\mathbb{S}\mathbb{R}(\mathcal{R})}^* t'$ implies $s \rightarrow_{\mathcal{R}}^* \widehat{t'}$.

Note that $\widehat{\cdot}$ is not defined for \perp , but $\widehat{\cdot}$ is a total function for reachable terms and their *structural* subterms [14].

$\mathbb{S}\mathbb{R}$ is *complete* for all normal CTRSs [14], i.e., for all terms s and t in $T(\mathcal{F}, \mathcal{V})$, $s \rightarrow_{\mathcal{R}}^* t$ implies $\langle \overline{s} \rangle \rightarrow_{\mathbb{S}\mathbb{R}(\mathcal{R})}^* \langle \overline{t} \rangle$. On the other hand, $\mathbb{S}\mathbb{R}$ is not sound for all normal CTRSs [14]. The first rule in \mathcal{R}_{aux} removes the nest of $\langle \cdot \rangle$, the second rule is used for shifting $\langle \cdot \rangle$ upward, and the third rules are used for both shifting $\langle \cdot \rangle$ upward and resetting the evaluation of conditions at the expanded arguments of \overline{f} (see [14] for the detail of the role of $\langle \cdot \rangle$ and its rules).

Example 2. Consider the following normal CTRS, a variant of the one in [14]:

$$\mathcal{R}_1 = \{ e(0) \rightarrow \text{true}, \quad e(s(x)) \rightarrow \text{true} \Leftarrow e(x) \Rightarrow \text{false}, \quad e(s(x)) \rightarrow \text{false} \Leftarrow e(x) \Rightarrow \text{true} \}$$

\mathcal{R}_1 is transformed by $\mathbb{S}\mathbb{R}$ as follows:

$$\mathbb{S}\mathbb{R}(\mathcal{R}_1) = \left\{ \begin{array}{l} \overline{e}(0, z_1, z_2) \rightarrow \langle \text{true} \rangle, \\ \overline{e}(s(x), \perp, z_2) \rightarrow \overline{e}(s(x), \langle \overline{e}(x, \perp, \perp) \rangle, z_2), \quad \overline{e}(s(x), \langle \text{false} \rangle, z_2) \rightarrow \langle \text{true} \rangle, \\ \overline{e}(s(x), z_1, \perp) \rightarrow \overline{e}(s(x), z_1, \langle \overline{e}(x, \perp, \perp) \rangle), \quad \overline{e}(s(x), z_1, \langle \text{true} \rangle) \rightarrow \langle \text{false} \rangle, \\ \langle \langle x \rangle \rangle \rightarrow \langle x \rangle, \quad s(\langle x \rangle) \rightarrow \langle s(x) \rangle, \quad \overline{e}(\langle x \rangle, z_1, z_2) \rightarrow \langle \overline{e}(x, \perp, \perp) \rangle \end{array} \right\}$$

$\mathbb{S}\mathbb{R}(\mathcal{R}_1)$ is confluent on reachable terms, but $\mathbb{S}\mathbb{R}(\mathcal{R}_1)$ is not confluent.

3 Proving Confluence of CTRSs via the Transformation

It has been shown that if $\mathbb{S}\mathbb{R}(\mathcal{R})$ is left-linear and confluent on reachable terms, then \mathcal{R} is confluent [14]. Note that by definition, \mathcal{R} is left-linear iff so is $\mathbb{S}\mathbb{R}(\mathcal{R})$. As described in [14], in this claim, left-linearity is assumed in order to ensure soundness. Here, we give a proof so as to relax the left-linearity to soundness.

Theorem 3. *If $\mathbb{S}\mathbb{R}(\mathcal{R})$ is sound for \mathcal{R} and confluent on reachable terms, then \mathcal{R} is confluent.*

Proof. Let s , t_1 , and t_2 be terms in $T(\mathcal{F}, \mathcal{V})$ such that $t_1 \leftarrow_{\mathcal{R}}^* s \rightarrow_{\mathcal{R}}^* t_2$. It follows from completeness of $\mathbb{S}\mathbb{R}$ that $\langle \overline{t_1} \rangle \leftarrow_{\mathbb{S}\mathbb{R}(\mathcal{R})}^* \langle \overline{s} \rangle \rightarrow_{\mathbb{S}\mathbb{R}(\mathcal{R})}^* \langle \overline{t_2} \rangle$. Since $\mathbb{S}\mathbb{R}(\mathcal{R})$ is confluent on reachable terms, there exists a term $u' \in T(\overline{\mathcal{F}}, \mathcal{V})$ such that $\langle \overline{t_1} \rangle \rightarrow_{\mathbb{S}\mathbb{R}(\mathcal{R})}^* u' \leftarrow_{\mathbb{S}\mathbb{R}(\mathcal{R})}^* \langle \overline{t_2} \rangle$. It follows from soundness of $\mathbb{S}\mathbb{R}(\mathcal{R})$ that $t_1 \rightarrow_{\mathcal{R}}^* \widehat{u'} \leftarrow_{\mathcal{R}}^* t_2$. Therefore, \mathcal{R} is confluent. \square

Theorem 3 means that soundness conditions of $\mathbb{S}\mathbb{R}$ are very useful in proving confluence of normal CTRSs. A normal CTRS \mathcal{R} is called *weakly left-linear* [6] if every conditional rewrite rule having at least one condition is left-linear, and for every unconditional rule, any non-linear variable in the left-hand side does not occur in the right-hand side. Note that a left-linear CTRS is weakly left-linear. Note also that \mathcal{R} is weakly left-linear iff so is $\mathbb{S}\mathbb{R}(\mathcal{R})$. It has been shown that $\mathbb{S}\mathbb{R}$ is sound for weakly left-linear normal CTRSs [12]. Thus, Theorem 3 provides us a new sufficient condition for confluence of normal CTRSs.

Theorem 4. *A weakly left-linear normal CTRS \mathcal{R} is confluent if $\mathbb{S}\mathbb{R}(\mathcal{R})$ is confluent on reachable terms.*

It would be difficult to directly prove that $\mathbb{S}\mathbb{R}(\mathcal{R})$ is confluent on reachable terms. A trivial sufficient condition for the property is confluence on arbitrary terms.

Lemma 5. *$\mathbb{S}\mathbb{R}(\mathcal{R})$ is confluent on reachable terms if $\mathbb{S}\mathbb{R}(\mathcal{R})$ is confluent.*

Due to Lemma 5, to prove confluence of \mathcal{R} , instead of proving confluence on reachable terms, we try to prove confluence of $\mathbb{S}\mathbb{R}(\mathcal{R})$. Unfortunately, this approach looks impractical.

Example 6. As described in Example 2, $\mathbb{S}\mathbb{R}(\mathcal{R}_1)$ is not confluent. Since \mathcal{R}_1 is a basic example of normal CTRSs, the combination of Theorem 4 and Lemma 5 looks unpractical.

It is often that critical pairs between rules in $\mathbb{S}\mathbb{R}(\mathcal{R}) \setminus \mathcal{R}_{aux}$ and \mathcal{R}_{aux} are not joinable. For this reason, confluence of $\mathbb{S}\mathbb{R}(\mathcal{R})$ is too severe to prove termination of \mathcal{R} , and thus, $\mathbb{S}\mathbb{R}$ is not so useful to prove confluence of normal CTRSs.

Let us consider a direct proof for confluence on reachable terms again. When $\mathbb{S}\mathbb{R}(\mathcal{R})$ is terminating, it would be sufficient to prove that any critical pair is joinable if an instance of the critical peak is reachable. For example, to prove confluence of $\mathbb{S}\mathbb{R}(\mathcal{R}_1)$ on reachable terms, it would be sufficient to prove that any instance of $\bar{e}(s(x), \langle \text{false} \rangle, \langle \text{true} \rangle)$ is not reachable. However, it is in general undecidable whether a term is reachable.

Instead of such unreachability, an optimization has already been discussed in [14]. We recall the optimization via the following example.

Example 7. Consider \mathcal{R}_1 in Example 2 again. The overlapping rules $e(s(x)) \rightarrow \text{true} \Leftarrow e(x) \rightarrow \text{false}$ and $e(s(x)) \rightarrow \text{false} \Leftarrow e(x) \rightarrow \text{true}$ have the same initial terms $e(x)$ of conditions to be evaluated. For this reason, we do not need two extra arguments of \bar{e} , and then $\mathbb{S}\mathbb{R}(\mathcal{R}_1)$ is optimized as follows, where we denote the optimization of $\mathbb{S}\mathbb{R}$ by $\mathbb{S}\mathbb{R}_{opt}$:

$$\mathbb{S}\mathbb{R}_{opt}(\mathcal{R}_1) = \left\{ \begin{array}{ll} \bar{e}(0, z) \rightarrow \langle \text{true} \rangle, & \bar{e}(s(x), \perp) \rightarrow \bar{e}(s(x), \langle \bar{e}(x, \perp) \rangle), \\ \bar{e}(s(x), \langle \text{false} \rangle) \rightarrow \langle \text{true} \rangle, & \bar{e}(s(x), \langle \text{true} \rangle) \rightarrow \langle \text{false} \rangle, \quad \dots \end{array} \right\}$$

$\mathbb{S}\mathbb{R}_{opt}(\mathcal{R}_1)$ is still not confluent since there are two critical pairs which are not joinable: $(\bar{e}(\langle s(x) \rangle), \langle \text{false} \rangle), \langle \text{true} \rangle)$ and $(\bar{e}(\langle s(x) \rangle), \langle \text{true} \rangle), \langle \text{false} \rangle)$. As described in [14], the introduced unary symbol $\langle \cdot \rangle$ and its related rules are not necessary for constructor CTRSs in the sense of soundness. Then, by removing them from $\mathbb{S}\mathbb{R}(\mathcal{R}_1)$, we obtain the following orthogonal TRS:

$$\{ \bar{e}(0, z) \rightarrow \text{true}, \bar{e}(s(x), \perp) \rightarrow \bar{e}(s(x), \bar{e}(x, \perp)), \bar{e}(s(x), \text{false}) \rightarrow \text{true}, \bar{e}(s(x), \text{true}) \rightarrow \text{false} \}$$

Note that the resulting TRS above is equivalent to that obtained by [1]. Therefore, from Theorem 4 and Lemma 5, \mathcal{R}_1 is confluent.

The optimization is useful in proving confluence of \mathcal{R}_2 , but it is not always successful.

Example 8. Consider the following constructor normal CTRS, a variant of \mathcal{R}_1 :

$$\mathcal{R}_2 = \left\{ \begin{array}{lll} e(0) \rightarrow \text{true}, & e(s(x)) \rightarrow \text{true} \Leftarrow o(x) \rightarrow \text{true}, & e(s(x)) \rightarrow \text{false} \Leftarrow e(x) \rightarrow \text{true}, \\ o(0) \rightarrow \text{false}, & o(s(x)) \rightarrow \text{true} \Leftarrow e(x) \rightarrow \text{true}, & o(s(x)) \rightarrow \text{false} \Leftarrow o(x) \rightarrow \text{true} \end{array} \right\}$$

For \mathcal{R}_2 , $\mathbb{S}\mathbb{R}_{opt}$ does not differ from $\mathbb{S}\mathbb{R}$, i.e., $\mathbb{S}\mathbb{R}(\mathcal{R}_2) = \mathbb{S}\mathbb{R}_{opt}(\mathcal{R}_2)$ even when $\langle \cdot \rangle$ is not introduced. $\mathbb{S}\mathbb{R}(\mathcal{R}_2)$ with/without $\langle \cdot \rangle$ is not confluent, and thus, useful to prove confluence of \mathcal{R}_2 .

Finally, we consider the case of non-constructor-based CTRSs.

Example 9. Consider the following normal CTRS over the signature $\{0, s, o, \text{true}, \text{false}\}$ [14]:

$$\mathcal{R}_3 = \left\{ \begin{array}{lll} o(x, o(y, l)) \rightarrow o(y, o(x, l)) \Leftarrow x < y \rightarrow \text{true}, \\ 0 < 0 \rightarrow \text{false}, & 0 < s(0) \rightarrow \text{true}, & 0 < s(s(x)) \rightarrow 0 < s(x), \\ s(0) < 0 \rightarrow \text{false}, & s(s(x)) < 0 \rightarrow s(x) < 0, & s(x) < s(y) \rightarrow x < y \end{array} \right\}$$

\mathcal{R}_3 is operationally terminating and the critical pair $(o(x, o(z, o(y, l))), o(y, o(x, o(z, l)))) \Leftarrow y < z \rightarrow \text{true}, x < y \rightarrow \text{true}$ is joinable. Thus, \mathcal{R}_3 is confluent. The CTRS \mathcal{R}_3 is transformed by $\mathbb{S}\mathbb{R}$ into the following TRS: $\mathbb{S}\mathbb{R}(\mathcal{R}_3) = \mathbb{S}\mathbb{R}_{opt}(\mathcal{R}_3) = \{ \bar{o}(x, \bar{o}(y, l, c), \perp) \rightarrow \bar{o}(x, \bar{o}(y, l, c), \langle x \bar{<} y \rangle), \bar{o}(x, \bar{o}(y, l, c), \langle \text{true} \rangle) \rightarrow \langle \bar{o}(y, \bar{o}(x, l, \perp)) \rangle, \dots \}$. $\mathbb{S}\mathbb{R}(\mathcal{R}_3)$ is not confluent because there is some critical pairs which are not joinable, e.g., $(\langle \bar{o}(x, \bar{o}(y, l, c), \perp) \rangle, \langle \bar{o}(y, \bar{o}(x, l, \perp)) \rangle)$. Notice that $\bar{o}(\langle x \rangle, l, z_1) \rightarrow \langle \bar{o}(x, l, \perp) \rangle \in \mathbb{S}\mathbb{R}(\mathcal{R}_3)$.

4 Conclusion

In this paper, we showed that a weakly left-linear normal CTRS is confluent if the transformed TRS is confluent (on reachable terms). Then, by an example, we showed uselessness of the improved criterion for the case that we attempt to use confluence on arbitrary terms instead of confluence on reachable terms. Finally, we discussed usefulness of the optimization of the SR transformation. We will make an experiment to evaluate usefulness of the optimization in terms of proving confluence of CTRSs. The discussion is not a sufficient evidence for usefulness of the optimization, and thus, we will formalize the optimization, adapt the claims which hold on \mathbb{SR} to the optimization, and compare the optimization with that of unravelings [7]. We will also develop a more practical method to prove confluence of a CTRS by using critical pairs of the transformed TRS.

References

- [1] S. Antoy, B. Brassel, and M. Hanus. Conditional narrowing without conditions. In *Proc. PPDP 2003*, pp. 20–31, ACM, 2003.
- [2] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
- [3] F. Durán, S. Lucas, J. Meseguer, C. Marché, and X. Urbain. Proving termination of membership equational programs. In *Proc. PEPM 2004*, pp. 147–158, ACM, 2004.
- [4] G. Feuillade and T. Genet. Reachability in conditional term rewriting systems. *Electr. Notes Theor. Comput. Sci.*, 86(1):133–146, 2003.
- [5] K. Gmeiner and B. Gramlich. Transformations of conditional rewrite systems revisited. In *Proc. WADT 2008*, vol. 5486 of *LNCS*, pp. 166–186. Springer, 2009.
- [6] K. Gmeiner, B. Gramlich, and F. Schernhammer. On (un)soundness of unravelings. In *Proc. RTA 2010*, vol. 6 of *LIPICs*, pp. 119–134, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010.
- [7] K. Gmeiner, N. Nishida, and B. Gramlich. Proving confluence of conditional term rewriting systems via unravelings. In *Proc. IWC 2013*, pp. 35–39, 2013.
- [8] S. Lucas, C. Marché, and J. Meseguer. Operational termination of conditional term rewriting systems. *Inf. Process. Lett.*, 95(4):446–453, 2005.
- [9] M. Marchiori. Unravelings and ultra-properties. In *Proc. ALP 1996*, vol. 1139 of *LNCA*, pp. 107–121, Springer, 1996.
- [10] M. Marchiori. On deterministic conditional rewriting. Computation Structures Group, Memo 405, MIT Laboratory for Computer Science, 1997.
- [11] N. Nishida, M. Sakai, and T. Sakabe. Soundness of unravelings for conditional term rewriting systems via ultra-properties related to linearity. *Logical Methods in Computer Science*, 8(3):1–49, 2012.
- [12] N. Nishida, M. Yanagisawa, and K. Gmeiner. On proving soundness of the computationally equivalent transformation for normal conditional term rewriting systems by using unravelings. In *Proc. WPTE 2014*, vol. 40 of *OASICs*, issue 1, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014 (to appear).
- [13] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer, 2002.
- [14] T.-F. Şerbănuţă and G. Roşu. Computationally equivalent elimination of conditions. In *Proc. RTA 2006*, vol. 4098 of *LNCS*, pp. 19–34, Springer, 2006.
- [15] T.-F. Şerbănuţă and G. Roşu. Computationally equivalent elimination of conditions. Technical Report UIUCDCS-R-2006-2693, Department of Computer Science, University of Illinois at Urbana-Champaign, 2006.
- [16] T. Suzuki, A. Middeldorp, and T. Ida. Level-confluence of conditional rewrite systems with extra variables in right-hand sides. In *Proc. RTA 1995*, vol. 914 of *LNCS*, pp. 179–193, Springer, 1995.
- [17] P. Viry. Elimination of conditions. *J. Symb. Comput.*, 28(3):381–401, 1999.