Sound Structure-Preserving Transformation for Weakly-Left-Linear Deterministic Conditional Term Rewriting Systems

Ryota Nakayama Naoki Nishida Masahiko Sakai
Graduate School of Information Science
Nagoya University
Nagoya, Japan
{nakayama@trs.cm., nishida@, sakai@}is.nagoya-u.ac.jp

In this paper, we show that the SR transformation, a computationally equivalent transformation proposed by Şerbănuţă and Roşu, is a sound structure-preserving transformation for weakly-left-linear deterministic conditional term rewriting systems. More precisely, we show that every weakly-left-linear deterministic conditional term rewriting system can be converted to an equivalent weakly-left-linear and ultra-weakly-left-linear deterministic conditional term rewriting system and prove that the SR transformation is sound for weakly-left-linear and ultra-weakly-left-linear deterministic conditional term rewriting systems. Here, soundness for a conditional term rewriting system means that reduction of the transformed unconditional term rewriting system creates no undesired reduction sequence for the conditional system.

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* [17], and *reachability* [3]. 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 in general an overapproximation of the CTRS in terms of reduction. Such an approach enables us to use techniques for the analysis of TRSs, which have been well investigated in the literature. For example, if the transformed TRS is terminating, then the CTRS is operationally terminating [2]—to prove termination of the transformed TRS, we can use many termination proving techniques that have been well investigated for TRSs (cf. [14]).

There are two approaches to transformations of CTRSs into TRSs: *unravelings* [9, 10] proposed by Marchiori (see, e.g., [4, 11]), and a transformation [18] proposed by Viry (see, e.g., [15, 4]).

Unravelings are transformations from a CTRS into a TRS over an extension of the original signature for the CTRS, which are *complete* for (reduction of) the CTRS [9]. Here, completeness for a CTRS means that for every reduction sequence of the CTRS, there exists a corresponding reduction sequence of the unraveled TRS. In this respect, the unraveled TRS is an overapproximation of the CTRS w.r.t. reduction, and is useful for analyzing the properties of the CTRS, such as syntactic properties, modularity, and operational termination, since TRSs are in general much easier to handle than CTRSs.

The latest transformation based on Viry's approach is a *computationally equivalent* transformation proposed by Şerbănuţă and Roşu [15, 16] (the SR transformation, for short), which is one of *structure-preserving* transformations [7]. This transformation has been proposed for normal CTRSs in [15]—started with this class to simplify the discussion—and then been extended to *strongly or syntactically deterministic* CTRSs (SDCTRSs) that are *ultra-left-linear* (*semilinear* [16]). Here, for a syntactic property P, a CTRS is said to be *ultra-P* if its unraveled TRS via Ohlebusch's unraveling [13] has the property

H. Cirstea, S. Escobar (Eds.): Third International Workshop on Rewriting Techniques for Program Transformations and Evaluation (WPTE'16).
EPTCS 235, 2017, pp. 62–77, doi:10.4204/EPTCS.235.5

P. The SR transformation converts a confluent, operationally terminating, and ultra-left-linear SDCTRS into a TRS that is *computationally equivalent* to the CTRS. This means that such a converted TRS can be used to exactly simulate any reduction sequence of the original CTRS to a normal form.

As for unravelings, soundness of the SR transformation plays a very important role for, e.g., computational equivalence. Here, soundness for a CTRS means that reduction of the converted TRS creates no undesired reduction sequences for the CTRS. Neither any unraveling nor the SR transformation is sound for all CTRSs. Since soundness is one of the most important properties for transformations of CTRSs, sufficient conditions for soundness have been well investigated, especially for unravelings (see, e.g., [5, 11, 6]). For example, the *simultaneous unraveling* that has been proposed by Marchiori [9] (and then has been improved by Ohlebusch [13]) is sound for *weakly-left-linear* (WLL, for short), *confluent*, *non-erasing*, or *ground conditional* normal CTRSs [5], and for DCTRSs that are *confluent and right-stable*, *WLL*, or *ultra-right-linear* [6]. Normal CTRSs admit a rewrite rule to have conditions to test terms received via variables in the left-hand side, e.g., whether a term with such variables can reach a ground normal form specified by the rule. This means that we can add so-called *guard* conditions to rewrite rules. In addition to such a function, DCTRSs admit a rewrite rule to have so-called `let`-structures in functional languages. On the other hand, the WLL property allows CTRSs to have rules, e.g., $\mathsf{eq}(x,x) \to \mathsf{true}$, to test equivalence between terms via non-linear variables. For these reasons, the class of WLL DCTRSs is one of the most interesting and practical classes of CTRSs, as well as that of WLL normal CTRSs.

The main purpose of transformations along the Viry's approach is to use the soundly transformed TRS in order to simulate the reduction of the original CTRS. The experimental results in [15] indicate that the rewriting engine using the soundly transformed TRS is much more efficient than the one using the original left-linear normal CTRS. To get an efficient rewriting engine for CTRSs, soundness conditions for the SR transformation are worth investigating.

In the case of DCTRSs that are not normal CTRSs, the SR transformation is defined for *ultra-left-linear* SDCTRSs, and has been shown to be sound for such SDCTRSs [16]. On the other hand, unlike unravelings, soundness conditions for the SR transformation have been investigated *only* for normal CTRSs [15, 16, 12]. For example, it has been shown in [12] that the SR transformation is sound for WLL normal CTRSs, but the result has not been adapted to WLL SDCTRSs yet.

In this paper, we show that the SR transformation is a sound structure-preserving transformation for WLL DCTRSs that do not have to be SDCTRSs. To this end, we first show that every WLL DCTRSs can be converted to a WLL and ultra-WLL DCTRS such that the reductions of these DCTRSs are the same. Then, we show that the SR transformation is applicable to ultra-WLL DCTRSs without any change. Finally, we prove that the SR transformation is sound for WLL and ultra-WLL DCTRSs. These results imply that the composition of the conversion to ultra-WLL DCTRSs and the SR transformation is a sound structure-preserving transformation for WLL DCTRSs.

The contribution of this paper is summarized as follows. We adapt the result on soundness of the SR transformation for WLL normal CTRSs to WLL *deterministic* CTRSs. The result in this paper covers the result in [12] for WLL normal CTRSs showing a simpler proof that would be helpful for further development of the SR transformation and its soundness.

This paper is organized as follows. In Section 2, we briefly recall basic notions and notations of term rewriting. In Section 3, we recall the notion of soundness, the simultaneous unraveling, and the SR transformation for DCTRSs, and show that every WLL DCTRS can be converted to an equivalent WLL and ultra-WLL DCTRS. In Section 4, we show that the SR transformation is sound for WLL and ultra-WLL DCTRSs. In Section 5, we conclude this paper and describe future work on this research. Some missing proofs are available at `http://www.trs.cm.is.nagoya-u.ac.jp/~nishida/wpte16/`.

2 Preliminaries

In this section, we recall basic notions and notations of term rewriting [1, 14].

Throughout the paper, we use $\mathcal{V}$ as a countably infinite set of *variables*. Let $\mathcal{F}$ be a *signature*, a finite set of *function symbols* each of which has its own fixed arity, and $arity_{\mathcal{F}}(\mathsf{f})$ be the arity of function symbol f. We often write $\mathsf{f}/n \in \mathcal{F}$ instead of "$\mathsf{f} \in \mathcal{F}$ and $arity_{\mathcal{F}}(\mathsf{f}) = n$", "$\mathsf{f} \in \mathcal{F}$ such that $arity_{\mathcal{F}}(\mathsf{f}) = n$", and so on. The set of *terms* over $\mathcal{F}$ and V ($\subseteq \mathcal{V}$) is denoted by $T(\mathcal{F},V)$, and the set of variables appearing in any of the terms $t_1,\ldots,t_n$ is denoted by $\mathcal{V}ar(t_1,\ldots,t_n)$. The number of occurrences of a variable x in a term sequence $t_1,\ldots,t_n$ is denoted by $|t_1,\ldots,t_n|_x$. A term t is called *ground* if $\mathcal{V}ar(t) = \emptyset$. A term is called *linear* if any variable occurs in the term at most once, and called *linear w.r.t. a variable* if the variable appears at most once in t. For a term t and a position p of t, the *subterm of t at p* is denoted by $t|_p$. The function symbol at the *root* position ε of term t is denoted by $root(t)$. Given an n-hole *context* $C[\,]$ with parallel positions $p_1,\ldots,p_n$, the notation $C[t_1,\ldots,t_n]_{p_1,\ldots,p_n}$ represents the term obtained by replacing hole $\Box$ at position p_i with term t_i for all $1 \leq i \leq n$. We may omit the subscript "$p_1,\ldots,p_n$" from $C[\ldots]_{p_1,\ldots,p_n}$. For positions p and p' of a term, we write $p' \geq p$ if p is a prefix of p' (i.e., there exists a sequence q such that $pq = p'$). Moreover, we write $p' > p$ if p is a proper prefix of p'.

A *substitution* σ is a mapping from variables to terms such that the number of variables x with $\sigma(x) \neq x$ is finite, and is naturally extended over terms. The *domain* and *range* of σ are denoted by $\mathcal{D}om(\sigma)$ and $\mathcal{R}an(\sigma)$, respectively. We may denote σ by $\{x_1 \mapsto t_1,\ \ldots,\ x_n \mapsto t_n\}$ if $\mathcal{D}om(\sigma) = \{x_1,\ldots,x_n\}$ and $\sigma(x_i) = t_i$ for all $1 \leq i \leq n$. For $\mathcal{F}$ and V ($\subseteq \mathcal{V}$), the set of *substitutions* that range over $\mathcal{F}$ and V is denoted by $\mathcal{S}ub(\mathcal{F},V)$: $\mathcal{S}ub(\mathcal{F},V) = \{\sigma \mid \mathcal{R}an(\sigma) \subseteq T(\mathcal{F},V)\}$. For a substitution σ and a term t, the application $\sigma(t)$ of σ to t is abbreviated to $t\sigma$, and $t\sigma$ is called an *instance* of t. Given a set X of variables, $\sigma|_X$ denotes the *restricted* substitution of σ w.r.t. X: $\sigma|_X = \{x \mapsto x\sigma \mid x \in \mathcal{D}om(\sigma) \cap X\}$.

An (oriented) *conditional rewrite rule* over a signature $\mathcal{F}$ is a triple (l,r,c), denoted by $l \to r \Leftarrow c$, such that the *left-hand side* l is a non-variable term in $T(\mathcal{F},\mathcal{V})$, the *right-hand side* r is a term in $T(\mathcal{F},\mathcal{V})$, and the *conditional part* c is a sequence $s_1 \twoheadrightarrow t_1,\ldots,s_k \twoheadrightarrow t_k$ of term pairs ($k \geq 0$) where all of $s_1,t_1,\ldots,s_k,t_k$ are terms in $T(\mathcal{F},\mathcal{V})$. In particular, a conditional rewrite rule is called *unconditional* if the conditional part is the empty sequence (i.e., $k = 0$), and we may abbreviate it to $l \to r$. We sometimes attach a unique label ρ to the conditional rewrite rule $l \to r \Leftarrow c$ by denoting $\rho : l \to r \Leftarrow c$, and we use the label to refer to the rewrite rule.

An *(oriented) conditional term rewriting system* (CTRS) over a signature $\mathcal{F}$ is a set of conditional rewrite rules over $\mathcal{F}$. A CTRS is called an (unconditional) *term rewriting system* (TRS) if every rule $l \to r \Leftarrow c$ in the CTRS is unconditional and satisfies $\mathcal{V}ar(l) \supseteq \mathcal{V}ar(r)$. The *reduction relation* $\to_{\mathcal{R}}$ of a CTRS $\mathcal{R}$ is defined as $\to_{\mathcal{R}} = \bigcup_{n\geq 0} \to_{(n),\mathcal{R}}$, where $\to_{(0),\mathcal{R}} = \emptyset$, and $\to_{(i+1),\mathcal{R}} = \{(C[l\sigma]_p, C[r\sigma]_p) \mid \rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1,\ldots,s_k \twoheadrightarrow t_k \in \mathcal{R},\ s_1\sigma \to^*_{(i),\mathcal{R}} t_1\sigma,\ \ldots,\ s_k\sigma \to^*_{(i),\mathcal{R}} t_k\sigma\}$ for $i \geq 0$. To specify the applied rule ρ and the position p where ρ is applied, we may write $\to_{p,\rho}$ or $\to_{p,\mathcal{R}}$ instead of $\to_{\mathcal{R}}$. Moreover, we may write $\to_{>\varepsilon,\mathcal{R}}$ instead of $\to_{p,\mathcal{R}}$ if $p > \varepsilon$. The *underlying unconditional system* $\{l \to r \mid l \to r \Leftarrow c \in \mathcal{R}\}$ of $\mathcal{R}$ is denoted by $\mathcal{R}_u$. A term t is called a *normal form* (of $\mathcal{R}$) if t is irreducible w.r.t. $\mathcal{R}$. For a CTRS $\mathcal{R}$, a substitution σ is called *normalized* (w.r.t. $\mathcal{R}$) if $x\sigma$ is a normal form w.r.t. $\mathcal{R}$ for every variable $x \in \mathcal{D}om(\sigma)$. A term t is called *strongly irreducible* (w.r.t. $\mathcal{R}$) if $t\sigma$ is a normal form w.r.t. $\mathcal{R}$ for every normalized substitution σ. The sets of *defined symbols* and *constructors* of $\mathcal{R}$ are denoted by $\mathcal{D}_{\mathcal{R}}$ and $\mathcal{C}_{\mathcal{R}}$, respectively: $\mathcal{D}_{\mathcal{R}} = \{root(l) \mid l \to r \Leftarrow c \in \mathcal{R}\}$ and $\mathcal{C}_{\mathcal{R}} = \mathcal{F} \setminus \mathcal{D}_{\mathcal{R}}$. Terms in $T(\mathcal{C}_{\mathcal{R}},\mathcal{V})$ are called *constructor terms of* $\mathcal{R}$. $\mathcal{R}$ is called a *constructor system* if for every rule $l \to r \Leftarrow c$ in $\mathcal{R}$, all proper subterms of the l are constructor terms of $\mathcal{R}$. A CTRS is called *operationally terminating* if there is no infinite well-formed trees in a certain logical inference system [8].

A conditional rewrite rule $l \to r \Leftarrow c$ is called *left-linear* (LL) if l is linear, *right-linear* (RL) if r is linear, *non-erasing* (NE) if $\mathcal{V}ar(l) \subseteq \mathcal{V}ar(r)$, and *ground conditional* if c contains no variable. A conditional rewrite rule $\rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$ is called *weakly-left-linear* (WLL) [6] if $|l, t_1, \ldots, t_k|_x = 1$ for any variable $x \in \mathcal{V}ar(r, s_1, \ldots, s_k)$. For a syntactic property P of conditional rewrite rules, we say that a CTRS has the property P if all of its rules have the property P, e.g., a CTRS is called *LL* if all of its rules are LL. Note that not all LL CTRSs are WLL, e.g., $\mathsf{f}(x) \to x \Leftarrow \mathsf{g}(x) \twoheadrightarrow x$ is LL but not WLL.

A conditional rewrite rule $\rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$ is called *deterministic* if $\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l, t_1, \ldots, t_{i-1})$ for all $1 \leq i \leq k$, called *strongly deterministic* if every term t_i is strongly irreducible w.r.t. $\mathcal{R}$, and called *syntactically deterministic* if every t_i is a constructor term or a ground normal form of $\mathcal{R}_u$. We simply call a deterministic CTRS a *DCTRS*, and call a strongly or syntactically deterministic CTRS an *SDCTRS*. In addition, ρ is classified according to the distribution of variables in ρ as follows: *Type 1* if $\mathcal{V}ar(r, s_1, t_1, \ldots, s_k, t_k) \subseteq \mathcal{V}ar(l)$; *Type 2* if $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$; *Type 3* if $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l, s_1, t_1, \ldots, s_k, t_k)$; *Type 4* otherwise. A (D)CTRS is called an *i-(D)CTRS* if all of its rules are of Type *i*. A DCTRS $\mathcal{R}$ is called *normal* (or a *normal CTRS*) if, for every rule $l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k \in \mathcal{R}$, all of $t_1, \ldots, t_k$ are ground normal forms w.r.t. $\mathcal{R}_u$. In this paper, we only consider 3-DCTRSs.

We often denote a term sequence $t_i, t_{i+1}, \ldots, t_j$ by $\overrightarrow{t_{i..j}}$. Moreover, for the application of a mapping τ to $\overrightarrow{t_{i..j}}$, we denote the sequence $\tau(t_i), \ldots, \tau(t_j)$ by $\overrightarrow{\tau(t_{i..j})}$, e.g., for a substitution θ, we denote $t_i\theta, \ldots, t_j\theta$ by $\overrightarrow{\theta(t_{i..j})}$. For a finite set $X = \{o_1, o_2, \ldots, o_n\}$ of objects, a sequence $o_1, o_2, \ldots, o_n$ under some arbitrary but fixed order on the objects is denoted by $\overrightarrow{X}$, and given a mapping τ, the sequence $\tau(o_1), \tau(o_2), \ldots, \tau(o_n)$ is denoted by $\tau(\overrightarrow{X})$. Given an object o, we denote the sequence $\overbrace{o, \ldots, o}^{n}$ by o^n.

3 Transformations from DCTRSs into TRSs

In this section, we first recall *soundness* and *completeness* of transformations, the *simultaneous unraveling* [14], and the SR transformation [15] for DCTRSs. Then, we show that every WLL DCTRS can be converted to an equivalent WLL and ultra-WLL DCTRS. In the following, we use the terminology "conditional" for a rewrite rule that has at least one condition, and distinguish "conditional rules" and "unconditional rules".

3.1 Soundness and Completeness between Two Rewriting Systems

We first show a general notion of soundness and completeness between two (C)TRSs (see [4, 11]). We usually consider that one is obtained by transforming the other. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be (C)TRSs over signature $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively, ϕ be an *initialization* (total) mapping from $T(\mathcal{F}_1, \mathcal{V})$ to $T(\mathcal{F}_2, \mathcal{V})$, and ψ be a partial inverse of ϕ, a so-called *backtranslation* mapping from $T(\mathcal{F}_2, \mathcal{V})$ to $T(\mathcal{F}_1, \mathcal{V})$ such that $\psi(\phi(t_1)) = t_1$ for any term $t_1 \in T(\mathcal{F}_1, \mathcal{V})$. We say that

- $\mathcal{R}_2$ is *sound for* (*reduction of*) $\mathcal{R}_1$ *w.r.t.* (ϕ, ψ) if, for any term $t_1 \in T(\mathcal{F}_1, \mathcal{V})$ and for any term $t_2 \in T(\mathcal{F}_2, \mathcal{V})$, $\phi(t_1) \to^*_{\mathcal{R}_2} t_2$ implies $t_1 \to^*_{\mathcal{R}_1} \psi(t_2)$ whenever $\psi(t_2)$ is defined, and
- $\mathcal{R}_2$ is *complete for* (*reduction of*) $\mathcal{R}_1$ *w.r.t.* ϕ if for all terms t_1 and t'_1 in $T(\mathcal{F}_1, \mathcal{V})$, $t_1 \to^*_{\mathcal{R}_1} t'_1$ implies $\phi(t_1) \to^*_{\mathcal{R}_2} \phi(t'_1)$.

We now suppose that $\mathcal{R}_1$ is a CTRS and $\mathcal{R}_2$ is a TRS. $\mathcal{R}_2$ is called *computationally equivalent to* $\mathcal{R}_1$ if for every $\mathcal{R}_1$-operationally-terminating term t in $T(\mathcal{F}_1, \mathcal{V})$ with a unique normal form u (i.e., $t \to^*_{\mathcal{R}_1} u$), the term $\phi(t)$ is terminating w.r.t. $\mathcal{R}_2$ and all the normal forms of $\phi(t)$ w.r.t. $\mathcal{R}_2$ are translated by ψ to u. Note that if $\mathcal{R}_1$ is operationally terminating, $\mathcal{R}_2$ is confluent, terminating, and sound for $\mathcal{R}_1$ w.r.t.

(ϕ,ψ), and ψ is defined for all normal forms t such that $\phi(s) \to^*_{\mathcal{R}_2} t$ for some $s \in T(\mathcal{F}_1,\mathcal{V})$, then $\mathcal{R}_2$ is computationally equivalent to $\mathcal{R}_1$.

3.2 Simultaneous Unraveling

A transformation U of CTRSs into TRSs is called an *unraveling* if for every CTRS $\mathcal{R}$, we have that $\to_{\mathcal{R}} \subseteq \to^*_{U(\mathcal{R})}$ and $U(\mathcal{R}\cup\mathcal{R}') = U(\mathcal{R})\cup\mathcal{R}'$ whenever $\mathcal{R}'$ is a TRS [9, 11]. The *simultaneous unraveling* for DCTRSs has been defined in [10], and then has been refined by Ohlebusch [13] as follows.

Definition 3.1 (**$\mathbb{U}$ [14]**) *Let $\mathcal{R}$ be a DCTRS over a signature $\mathcal{F}$. For each conditional rule $\rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1,\dots,s_k \twoheadrightarrow t_k$ in $\mathcal{R}$, we introduce k new function symbols $U^\rho_1,\dots,U^\rho_k$, and transform ρ into a set of $k+1$ unconditional rules as follows:*

$$\mathbb{U}(\rho) = \{\quad l \to U^\rho_1(s_1,\overrightarrow{X_1}),\quad U^\rho_1(t_1,\overrightarrow{X_1}) \to U^\rho_2(s_2,\overrightarrow{X_2}),\quad \dots,\quad U^\rho_k(t_k,\overrightarrow{X_k}) \to r \quad\}$$

where $X_i = \mathcal{V}ar(l,\overrightarrow{t_{1..i-1}})$ for $1 \le i \le k$. We define $\mathbb{U}$ for an unconditional rule $l \to r \in \mathcal{R}$ as $\mathbb{U}(l \to r) = \{l \to r\}$. $\mathbb{U}$ is straightforwardly extended to DCTRSs: $\mathbb{U}(\mathcal{R}) = \bigcup_{\rho\in\mathcal{R}} \mathbb{U}(\rho)$. We abuse $\mathbb{U}$ to represent the extended signature of $\mathcal{F}$: $\mathbb{U}_{\mathcal{R}}(\mathcal{F}) = \mathcal{F} \cup \{U^\rho_i \mid \rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1,\dots,s_k \twoheadrightarrow t_k \in \mathcal{R},\ 1 \le i \le k\}$. We say that $\mathbb{U}$ (and also $\mathbb{U}(\mathcal{R})$) is sound *for $\mathcal{R}$ if $\mathbb{U}(\mathcal{R})$ is sound for $\mathcal{R}$ w.r.t. $(\mathtt{id}_{\mathcal{F}},\mathtt{id}_{\mathbb{U}_{\mathcal{R}}(\mathcal{F})})$, where $\mathtt{id}_{\mathcal{F}}$ is the identity mapping for $T(\mathcal{F},\mathcal{V})$, and $\mathtt{id}_{\mathbb{U}_{\mathcal{R}}(\mathcal{F})}$ is the partial identity mapping for $T(\mathbb{U}_{\mathcal{R}}(\mathcal{F}),\mathcal{V})$, i.e., $\mathtt{id}_{\mathcal{F}}(t) = \mathtt{id}_{\mathbb{U}_{\mathcal{R}}(\mathcal{F})}(t) = t$ for $t \in T(\mathcal{F},\mathcal{V})$ and $\mathtt{id}_{\mathbb{U}_{\mathcal{R}}(\mathcal{F})}(t)$ is undefined for $t \in T(\mathbb{U}_{\mathcal{R}}(\mathcal{F}),\mathcal{V}) \setminus T(\mathcal{F},\mathcal{V})$. We also say that $\mathbb{U}$ (and also $\mathbb{U}(\mathcal{R})$) is* complete *for $\mathcal{R}$ if $\mathbb{U}(\mathcal{R})$ is complete for $\mathcal{R}$ w.r.t. $\mathtt{id}_{\mathcal{F}}$.*

Note that $\mathbb{U}(\mathcal{R})$ is a TRS over $\mathbb{U}_{\mathcal{R}}(\mathcal{F})$, i.e., $\mathbb{U}$ transforms a DCTRS into a TRS. In examples below, we use $\mathsf{u}_1,\mathsf{u}_2,\dots$ for fresh U symbols introduced during the application of $\mathbb{U}$.

Example 3.2 Consider the following DCTRS from [14, Section 7.2.5]:

$$\mathcal{R}_1 = \left\{\begin{array}{r}
\mathsf{split}(x,\mathsf{nil}) \to \mathsf{pair}(\mathsf{nil},\mathsf{nil}),\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \\
\mathsf{split}(x,\mathsf{cons}(y,ys)) \to \mathsf{pair}(xs,\mathsf{cons}(y,zs)) \Leftarrow \mathsf{split}(x,ys) \twoheadrightarrow \mathsf{pair}(xs,zs),\ x \le y \twoheadrightarrow \mathsf{true}, \\
\mathsf{split}(x,\mathsf{cons}(y,ys)) \to \mathsf{pair}(\mathsf{cons}(y,xs),zs) \Leftarrow \mathsf{split}(x,ys) \twoheadrightarrow \mathsf{pair}(xs,zs),\ x \le y \twoheadrightarrow \mathsf{false}, \\
\mathsf{qsort}(\mathsf{nil}) \to \mathsf{nil},\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \\
\mathsf{qsort}(\mathsf{cons}(x,xs)) \to \mathsf{qsort}(ys) +\!\!+ \mathsf{cons}(x,\mathsf{qsort}(zs)) \Leftarrow \mathsf{split}(x,xs) \twoheadrightarrow \mathsf{pair}(ys,zs)
\end{array}\right\} \cup \mathcal{R}_2$$

where

$$\mathcal{R}_2 = \left\{\begin{array}{lll}
0 \le y \to \mathsf{true}, & \mathsf{s}(x) \le 0 \to \mathsf{false}, & \mathsf{s}(x) \le \mathsf{s}(y) \to x \le y, \\
\mathsf{nil} +\!\!+ ys \to ys, & \mathsf{cons}(x,xs) +\!\!+ ys \to \mathsf{cons}(x,xs +\!\!+ ys) &
\end{array}\right\}$$

Introducing U symbols u_1, u_2, u_3, u_4, and u_5 for conditional rules in $\mathcal{R}_1$, $\mathcal{R}_1$ is unraveled by $\mathbb{U}$ as follows:

$$\mathbb{U}(\mathcal{R}_1) = \left\{\begin{array}{rl}
\mathsf{split}(x,\mathsf{nil}) \to & \mathsf{pair}(\mathsf{nil},\mathsf{nil}), \\
\mathsf{split}(x,\mathsf{cons}(y,ys)) \to & \mathsf{u}_1(\mathsf{split}(x,ys),x,y,ys), \\
\mathsf{u}_1(\mathsf{pair}(xs,zs),x,y,ys) \to & \mathsf{u}_2(x \le y,x,y,ys,xs,zs), \\
\mathsf{u}_2(\mathsf{true},x,y,ys,xs,zs) \to & \mathsf{pair}(xs,\mathsf{cons}(y,zs)), \\
\mathsf{split}(x,\mathsf{cons}(y,ys)) \to & \mathsf{u}_3(\mathsf{split}(x,ys),x,y,ys), \\
\mathsf{u}_3(\mathsf{pair}(xs,zs),x,y,ys) \to & \mathsf{u}_4(x \le y,x,y,ys,xs,zs), \\
\mathsf{u}_4(\mathsf{false},x,y,ys,xs,zs) \to & \mathsf{pair}(\mathsf{cons}(y,xs),zs), \\
\mathsf{qsort}(\mathsf{nil}) \to & \mathsf{nil}, \\
\mathsf{qsort}(\mathsf{cons}(x,xs)) \to & \mathsf{u}_5(\mathsf{split}(x,xs),x,xs), \\
\mathsf{u}_5(\mathsf{pair}(ys,zs),x,xs) \to & \mathsf{qsort}(ys) +\!\!+ \mathsf{cons}(x,\mathsf{qsort}(zs))
\end{array}\right\} \cup \mathcal{R}_2$$

As shown in [9, 6], $\mathbb{U}$ is not sound for all DCTRSs, while $\mathbb{U}$ is sound for some classes of DCTRSs, e.g., "confluent and right-stable", "WLL", and "RL" (cf. [6]).

Theorem 3.3 ([6]) *$\mathbb{U}$ is sound for WLL DCTRSs.*

Let P be a property on rewrite rules, and U be an unraveling. A conditional rewrite rule ρ is said to be *ultra-P w.r.t. U* (U-P, for short) if all the rules in $U(\rho)$ have the property P. Note that U-P is a syntactic property on rewrite rules, and thus a DCTRS is called U-P if all rules in the DCTRS are U-P. For example, $\mathcal{R}$ is $\mathbb{U}$-LL if $\mathbb{U}(\mathcal{R})$ is LL. Some ultra-properties are reformulated without referring to unraveled systems (cf. [11]). In addition, by definition, the $\mathbb{U}$-WLL property is characterized without $\mathbb{U}$ as follows.

Theorem 3.4 *$\mathcal{R}$ is $\mathbb{U}$-WLL if and only if all unconditional rules in $\mathcal{R}$ are WLL and every conditional rule $l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$ ($k > 0$) in $\mathcal{R}$ satisfies that*

- *(a) the sequence $l, t_1, \ldots, t_{k-1}$ is linear, and*
- *(b) $|l, t_1, \ldots, t_k|_x \leq 1$ for any variable $x \in \mathcal{V}ar(r)$.*

Note that every $\mathbb{U}$-LL DCTRS is $\mathbb{U}$-WLL, while the converse of this implication does not hold in general. On the other hand, the class of $\mathbb{U}$-WLL DCTRSs is incomparable with the class of WLL DCTRSs, e.g., $\mathsf{f}(x) \to x \Leftarrow \mathsf{a} \twoheadrightarrow y,\ \mathsf{b} \twoheadrightarrow y,\ x \twoheadrightarrow \mathsf{c}$ is WLL but not $\mathbb{U}$-WLL, and $\mathsf{f}(x) \to x \Leftarrow \mathsf{a} \twoheadrightarrow y,\ y \twoheadrightarrow \mathsf{b},\ \mathsf{c} \twoheadrightarrow y$ is $\mathbb{U}$-WLL but not WLL. Though, every WLL DCTRS can be converted to a WLL and $\mathbb{U}$-WLL DCTRS such that the reductions of these DCTRSs are the same.

In the following, we show that every WLL DCTRS $\mathcal{R}$ can be converted to an equivalent WLL and $\mathbb{U}$-WLL DCTRS. We first convert a WLL conditional rule $\rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$ to a WLL and $\mathbb{U}$-WLL one as follows: for every variable x in ρ such that $|l, t_1, \ldots, t_k|_x > 1$, we linearize the occurrences of x by replacing each of them by a fresh variable, obtaining $\rho' : l' \to r \Leftarrow s_1 \twoheadrightarrow t'_1, \ldots, s_k \twoheadrightarrow t'_k$;[1] Let $x_1, \ldots, x_j$ be the introduced variables, and σ be the variable renaming that maps x_i to the original one, i.e., $\mathcal{D}om(\sigma) = \{x_1, \ldots, x_j\}$, $l'\sigma = l$, and $t'_i\sigma = t_i$ for $1 \leq i \leq k$; We add the condition $\mathsf{tuple}_j(x_1, \ldots, x_j) \twoheadrightarrow \mathsf{tuple}_j(x_1\sigma, \ldots, x_j\sigma)$ into ρ' as the last condition, where tuple_j is a fresh j-ary constructor. We denote this transformation by $\mathbb{T}$, i.e., $\mathbb{T}(\rho) = l' \to r \Leftarrow s_1 \twoheadrightarrow t'_1, \ldots, s_k \twoheadrightarrow t'_k, \mathsf{tuple}_j(x_1, \ldots, x_j) \twoheadrightarrow \mathsf{tuple}_j(x_1\sigma, \ldots, x_j\sigma)$. In addition, we abuse $\mathbb{T}$ for unconditional rules and $\mathcal{R}$: $\mathbb{T}(l \to r) = l \to r$ and $\mathbb{T}(\mathcal{R}) = \{\mathbb{T}(\rho) \mid \rho \in \mathcal{R}\}$. By definition, $\mathbb{T}(\rho)$ is WLL and $\mathbb{U}$-WLL, i.e., $\mathbb{T}$ transforms a WLL DCTRS into a WLL and $\mathbb{U}$-WLL DCTRS. It is clear that if $s \in T(\mathcal{F}, \mathcal{V})$ and $s \to^*_{\mathbb{T}(\mathcal{R})} t$, then $t \in T(\mathcal{F}, \mathcal{V})$.

Theorem 3.5 *Let $\mathcal{R}$ be a WLL DCTRS over a signature $\mathcal{F}$. Then, $\to^*_{\mathcal{R}} = \to^*_{\mathbb{T}(\mathcal{R})}$ over $T(\mathcal{F}, \mathcal{V})$*

Proof (Sketch). The following two claims can be proved by induction on the lexicographic product (m, n): (i) if $s \to^n_{(m),\mathcal{R}} t$ then $s \to^*_{\mathbb{T}(\mathcal{R})} t$, and (ii) if $s \to^n_{(m),\mathbb{T}(\mathcal{R})} t$ then $s \to^*_{\mathcal{R}} t$. □

3.3 The SR Transformation

Next, we introduce the SR transformation and its properties. Before transforming a CTRS $\mathcal{R}$, we first extend the signature of $\mathcal{R}$ as follows:

- we keep the constructors of $\mathcal{R}$, while replacing c/n by $\overline{\mathsf{c}}/n$,

[1] Such x does not appear in any of $r, s_1, \ldots, s_k$ because ρ is WLL.

- the arity n of defined symbol f is extended to $n+m$ where f has m conditional rules in $\mathcal{R}$, replacing f by $\overline{\mathsf{f}}$, the arity of which is $n+m$,
- a fresh constant $\bot$ and a fresh unary symbol $\langle\cdot\rangle$ are introduced, and
- for every conditional rule $\rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$ in $\mathcal{R}$, we introduce k fresh symbols $[\,]^{\rho}_{1}, [\,]^{\rho}_{2}, \ldots, [\,]^{\rho}_{k}$ with the arities $1, 1+|\mathcal{V}ar(t_1)|, 1+|\mathcal{V}ar(t_1,t_2)|, \ldots, 1+|\mathcal{V}ar(t_1,\ldots,t_{k-1})|$.

We assume that for every defined symbol f, the conditional rules for f are ranked by some arbitrary but fixed order. We denote the extended signature by $\overline{\mathcal{F}}$: $\overline{\mathcal{F}} = \{\overline{\mathsf{c}} \mid \mathsf{c} \in \mathcal{C}_{\mathcal{R}}\} \cup \{\overline{\mathsf{f}} \mid \mathsf{f} \in \mathcal{D}_{\mathcal{R}}\} \cup \{\bot, \langle\cdot\rangle\} \cup \{[\,]^{\rho}_{j} \mid \rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k \in \mathcal{R},\ 1 \le j \le k\}$. We introduce a mapping $\mathtt{ext}$ to extend the arguments of defined symbols in a term as follows: $\mathtt{ext}(x) = x$ for $x \in \mathcal{V}$; $\mathtt{ext}(\mathsf{c}(\overrightarrow{t_{1..n}})) = \mathsf{c}(\overrightarrow{\mathtt{ext}(t_{1..n})})$ for $\overline{\mathsf{c}}/n \in \mathcal{C}_{\mathcal{R}}$; $\mathtt{ext}(\mathsf{f}(\overrightarrow{t_{1..n}})) = \overline{\mathsf{f}}(\overrightarrow{\mathtt{ext}(t_{1..n})}, \overrightarrow{z_{1..m}})$ for $\mathsf{f}/n \in \mathcal{D}_{\mathcal{R}}$, where f has m conditional rules in $\mathcal{R}$, $arity_{\overline{\mathcal{F}}}(\overline{\mathsf{f}}) = n+m$, and $z_1, \ldots, z_m$ are fresh variables. The extended arguments of $\overline{\mathsf{f}}$ are used for evaluating the corresponding conditions, and the fresh constant $\bot$ is introduced to the extended arguments of defined symbols, which does not store any evaluation. To put $\bot$ into the extended arguments, we define a mapping $(\cdot)^{\bot}$ that puts $\bot$ to all the extended arguments of defined symbols, as follows: $(x)^{\bot} = x$ for $x \in \mathcal{V}$; $(\overline{\mathsf{c}}(\overrightarrow{t_{1..n}}))^{\bot} = \overline{\mathsf{c}}(\overrightarrow{(t_{1..n})^{\bot}})$ for $\mathsf{c}/n \in \mathcal{C}_{\mathcal{R}}$; $(\overline{\mathsf{f}}(\overrightarrow{t_{1..n}}, \overrightarrow{u_{1..m}}))^{\bot} = \overline{\mathsf{f}}(\overrightarrow{(t_{1..n})^{\bot}}, \bot, \ldots, \bot)$ for $\mathsf{f}/n \in \mathcal{D}_{\mathcal{R}}$; $(\langle t\rangle)^{\bot} = \langle (t)^{\bot}\rangle$; $(\bot)^{\bot} = \bot$; $([\ldots]^{\rho}_{j})^{\bot} = \bot$. Now we define a mapping $\bar{\cdot}$ from $T(\mathcal{F},\mathcal{V})$ to $T(\overline{\mathcal{F}},\mathcal{V})$ as $\bar{t} = (\mathtt{ext}(t))^{\bot}$. On the other hand, the partial inverse mapping $\widehat{\cdot}$ for $\bar{\cdot}$ is defined as follows: $\widehat{x} = x$ for $x \in \mathcal{V}$; $\widehat{\overline{\mathsf{c}}(\overrightarrow{t_{1..n}})} = \mathsf{c}(\widehat{t_1}, \ldots, \widehat{t_n})$ for $\mathsf{c}/n \in \mathcal{C}_{\mathcal{R}}$; $\widehat{\overline{\mathsf{f}}(\overrightarrow{t_{1..n}}, \ldots)} = \mathsf{f}(\widehat{t_1}, \ldots, \widehat{t_n})$ for $\mathsf{f}/n \in \mathcal{D}_{\mathcal{R}}$; $\widehat{\langle t\rangle} = \widehat{t}$. Note that in applying $(\cdot)^{\bot}$ or $\widehat{\cdot}$ to *reachable terms* defined later, the case of applying $(\cdot)^{\bot}$ to $\bot$ or $[\ldots]^{\rho}_{j}$ never happens.

The SR transformation [16] for SDCTRSs has been defined for only $\mathbb{U}$-LL SDCTRSs—more precisely, any other case has not been discussed in [16]. Originally, to generate a computationally equivalent TRS, a given CTRS $\mathcal{R}$ is assumed to be a $\mathbb{U}$-LL SDCTRS, while such an assumption is a sufficient condition for computational equivalence. To define the transformation itself, $\mathcal{R}$ does not have to be an SDCTRS, but the $\mathbb{U}$-LL property is used to ensure that for $\rho : l \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$, the sequence $l, t_1, \ldots, t_{k-1}$ is linear. To ensure it, the $\mathbb{U}$-WLL property is enough because of Theorem 3.4 (a). For this reason, the SR transformation is applicable not only to $\mathbb{U}$-LL SDCTRSs but also to $\mathbb{U}$-WLL DCTRSs without any change.

Definition 3.6 **($\mathbb{SR}$ [16])** *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS over a signature $\mathcal{F}$ and $\overline{\mathcal{F}}$ be the extended signature of $\mathcal{F}$ mentioned above. Then, the i-th conditional f-rule $\rho : \mathsf{f}(\overrightarrow{w_{1..n}}) \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$ is transformed into a set of $k+1$ unconditional rules as follows:*

$$\mathbb{SR}(\rho) = \left\{ \begin{array}{lcl} \overline{\mathsf{f}}(\overrightarrow{w'_{1..n}}, \overrightarrow{z_{1..i-1}}, \quad \bot, \quad \overrightarrow{z_{i+1..m}}) & \to & \overline{\mathsf{f}}(\overrightarrow{w'_{1..n}}, \overrightarrow{z_{1..i-1}}, [\langle \overline{s_1}\rangle, \overrightarrow{V_1}]^{\rho}_{1}, \overrightarrow{z_{i+1..m}}), \\ \overline{\mathsf{f}}(\overrightarrow{w'_{1..n}}, \overrightarrow{z_{1..i-1}}, [\langle \mathtt{ext}(t_1)\rangle, \overrightarrow{V_1}]^{\rho}_{1}, \overrightarrow{z_{i+1..m}}) & \to & \overline{\mathsf{f}}(\overrightarrow{w'_{1..n}}, \overrightarrow{z_{1..i-1}}, [\langle \overline{s_2}\rangle, \overrightarrow{V_2}]^{\rho}_{2}, \overrightarrow{z_{i+1..m}}), \\ & \vdots & \\ \overline{\mathsf{f}}(\overrightarrow{w'_{1..n}}, \overrightarrow{z_{1..i-1}}, [\langle \mathtt{ext}(t_k)\rangle, \overrightarrow{V_k}]^{\rho}_{k}, \overrightarrow{z_{i+1..m}}) & \to & \langle \overline{r}\rangle \end{array} \right\}$$

where $\overrightarrow{w'_{1..n}} = \overrightarrow{\mathtt{ext}(w_{1..n})}$, $V_j = \mathcal{V}ar(\overrightarrow{t_{1..j-1}})$ for all $1 \le j \le k$,[2] and $z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_m$ are fresh variables. An unconditional rule in $\mathcal{R}$ is converted as follows: $\mathbb{SR}(l \to r) = \{\ \mathtt{ext}(l) \to \langle \overline{r}\rangle\ \}$, that is, $\mathbb{SR}(\mathsf{f}(\overrightarrow{w_{1..n}}) \to r) = \{\ \overline{\mathsf{f}}(\overrightarrow{\mathtt{ext}(w_{1..n})}, \overrightarrow{z_{1..m}}) \to \langle \overline{r}\rangle\ \}$, where $z_1, \ldots, z_m$ are fresh variables. The set of

[2] For arbitrary DCTRSs, we may define V_j as $V_j = \mathcal{V}ar(\overrightarrow{t_{1..j-1}}) \setminus \mathcal{V}ar(\overrightarrow{w_{1..n}})$.

auxiliary rules is defined as follows:

$$\mathcal{R}_{aux} = \{\ \langle\langle x\rangle\rangle \to \langle x\rangle\ \} \cup \{\ \overline{\mathsf{c}}(\overrightarrow{x_{1..i-1}}, \langle x_i\rangle, \overrightarrow{x_{i+1..n}}) \to \langle \overline{\mathsf{c}}(\overrightarrow{x_{1..n}})\rangle \mid \mathsf{c}/n \in \mathcal{C}_{\mathcal{R}},\ 1 \le i \le n\ \}$$
$$\cup \{\ \overline{\mathsf{f}}(\overrightarrow{x_{1..i-1}}, \langle x_i\rangle, \overrightarrow{x_{i+1..n}}, \overrightarrow{z_{1..m}}) \to \langle \overline{\mathsf{f}}(\overrightarrow{x_{1..n}}, \bot, \ldots, \bot)\rangle \mid \mathsf{f}/n \in \mathcal{D}_{\mathcal{R}},\ 1 \le i \le n\ \}$$

where $x_1,\ldots,x_n,z_1,\ldots,z_m$ are distinct variables. The transformation $\mathbb{SR}$ is defined as follows: $\mathbb{SR}(\mathcal{R}) = \bigcup_{\rho\in\mathcal{R}} \mathbb{SR}(\rho) \cup \mathcal{R}_{aux}$. We say that $\mathbb{SR}$ (and also $\mathbb{SR}(\mathcal{R})$) is sound for $\mathcal{R}$ *if $\mathbb{SR}(\mathcal{R})$ is sound for $\mathcal{R}$ w.r.t. $(\langle\bar{\cdot}\rangle, \widehat{\cdot})$. We also say that $\mathbb{SR}$ (and also $\mathbb{SR}(\mathcal{R})$) is* complete for $\mathcal{R}$ *if $\mathbb{SR}(\mathcal{R})$ is complete for $\mathcal{R}$ w.r.t. $\langle\bar{\cdot}\rangle$.*

Note that $\mathbb{SR}(\mathcal{R})$ is a TRS over $\overline{\mathcal{F}}$, i.e., $\mathbb{SR}$ transforms a $\mathbb{U}$-WLL DCTRS into a TRS. In examples below, we use $[\,]_1, [\,]_2, \ldots$ for fresh tuple symbols introduced during the application of $\mathbb{SR}$, and we may abuse f instead of $\overline{\mathsf{f}}$ if all the rules for f in $\mathcal{R}$ are unconditional, and as in [15, 16], the original constructor c is abused instead of $\overline{\mathsf{c}}$. It has been shown in [15] that $\mathbb{SR}$ is complete for all $\mathbb{U}$-LL SDCTRSs. By definition, it is clear that $\mathbb{SR}$ is also complete for all $\mathbb{U}$-WLL DCTRSs.

Theorem 3.7 *$\mathbb{SR}$ is complete for $\mathbb{U}$-WLL DCTRSs.*

To evaluate conditions of the i-th conditional rule $\mathsf{f}(\overrightarrow{w_{1..n}}) \to r_i \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$, the i-th conditional rule is transformed into the $k+1$ unconditional rules: a term of the form $[\langle t\rangle, \overrightarrow{u_{1..n_j}}]^{\rho}_{j}$ represents an intermediate state t of the evaluation of the j-th condition $s_j \twoheadrightarrow t_j$ carrying $\overrightarrow{u_{1..n_j}}$ for $\mathcal{V}ar(\overrightarrow{t_{1..j-1}})$, the first unconditional rule starts to evaluate the condition (an instance of $\overline{s_1}$), and the remaining k rules examine whether the corresponding conditions hold. On the other hand, the first rule $\langle\langle x\rangle\rangle \to \langle x\rangle$ in $\mathcal{R}_{aux}$ removes the nesting of $\langle\cdot\rangle$, the second rule $\overline{\mathsf{c}}(\overrightarrow{x_{1..i-1}}, \langle x_i\rangle, \overrightarrow{x_{i+1..n}}) \to \langle\overline{\mathsf{c}}(\overrightarrow{x_{1..n}})\rangle$ is used for shifting $\langle\cdot\rangle$ upward, and the third rule $\overline{\mathsf{f}}(\overrightarrow{x_{1..i-1}}, \langle x_i\rangle, \overrightarrow{x_{i+1..n}}, \overrightarrow{z_{1..m}}) \to \langle\overline{\mathsf{f}}(\overrightarrow{x_{1..n}}, \bot, \ldots, \bot)\rangle$ is used for both shifting $\langle\cdot\rangle$ upward and resetting the evaluation of conditions at the extended arguments of $\overline{\mathsf{f}}$. The unary symbol $\langle\cdot\rangle$ and its rules in $\mathcal{R}_{aux}$ are introduced to preserve confluence of the original CTRS $\mathcal{R}$ on reachable terms (see [15] for the detail of the role of $\langle\cdot\rangle$ and its rules).

Example 3.8 Consider $\mathcal{R}_1$ in Example 3.2 again. Introducing tuple symbols $[\,]_1$, $[\,]_2$, $[\,]_3$, $[\,]_4$, and $[\,]_5$, $\mathcal{R}_1$ is transformed by $\mathbb{SR}$ as follows:

$$\mathbb{SR}(\mathcal{R}_1) = \left\{ \begin{array}{rcl}
\overline{\mathsf{split}}(x, \mathsf{nil}, z_1, z_2) & \to & \langle \mathsf{pair}(\mathsf{nil}, \mathsf{nil})\rangle \\
\overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), \bot, z_2) & \to & \overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), [\langle\overline{\mathsf{split}}(x, ys, \bot, \bot)\rangle]_1, z_2) \\
\overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), [\langle\mathsf{pair}(xs, zs)\rangle]_1, z_2) & \to & \overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), [\langle x \le y\rangle, xs, zs]_2, z_2) \\
\overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), [\langle\mathsf{true}\rangle, xs, zs]_2, z_2) & \to & \langle\mathsf{pair}(xs, \mathsf{cons}(y, zs))\rangle \\
\overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), z_1, \bot) & \to & \overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), z_1, [\langle\overline{\mathsf{split}}(x, ys, \bot, \bot)\rangle]_3) \\
\overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), z_1, [\langle\mathsf{pair}(xs, zs)\rangle]_3) & \to & \overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), z_1, [\langle x \le y\rangle, xs, zs]_4) \\
\overline{\mathsf{split}}(x, \mathsf{cons}(y, ys), z_1, [\langle\mathsf{false}\rangle, xs, zs]_4) & \to & \langle\mathsf{pair}(\mathsf{cons}(y, xs), zs)\rangle \\
\overline{\mathsf{qsort}}(\mathsf{nil}, z_1) & \to & \langle\mathsf{nil}\rangle \\
\overline{\mathsf{qsort}}(\mathsf{cons}(x, xs), \bot) & \to & \overline{\mathsf{qsort}}(\mathsf{cons}(x, xs), [\langle\overline{\mathsf{split}}(x, xs, \bot, \bot)\rangle]_5) \\
\overline{\mathsf{qsort}}(\mathsf{cons}(x, xs), [\langle\mathsf{pair}(ys, zs)\rangle]_5) & \to & \langle\overline{\mathsf{qsort}}(ys, \bot) +\!\!+\, \mathsf{cons}(x, \overline{\mathsf{qsort}}(zs, \bot))\rangle
\end{array} \right\} \cup \mathcal{R}_3$$

where

$$\mathcal{R}_3 = \left\{ \begin{array}{lll} 0 \le y \to \langle \mathsf{true} \rangle, & \mathsf{s}(x) \le 0 \to \langle \mathsf{false} \rangle, & \mathsf{s}(x) \le \mathsf{s}(y) \to \langle x \le y \rangle, \\ \mathsf{nil} +\!\!+ ys \to \langle ys \rangle, & \mathsf{cons}(x,xs) +\!\!+ ys \to \langle \mathsf{cons}(x, xs +\!\!+ ys) \rangle, & \\ \langle \langle x \rangle \rangle \to \langle x \rangle, & & \mathsf{s}(\langle x \rangle) \to \langle \mathsf{s}(x) \rangle, \\ \mathsf{cons}(\langle x \rangle, xs) \to \langle \mathsf{cons}(x,xs) \rangle, & & \mathsf{cons}(x, \langle xs \rangle) \to \langle \mathsf{cons}(x,xs) \rangle, \\ \mathsf{pair}(\langle x \rangle, y) \to \langle \mathsf{pair}(x,y) \rangle, & & \mathsf{pair}(x, \langle y \rangle) \to \langle \mathsf{pair}(x,y) \rangle, \\ \overline{\mathsf{split}}(\langle x \rangle, ys, z_1, z_2) \to \langle \overline{\mathsf{split}}(x, ys, \bot, \bot) \rangle, & & \overline{\mathsf{split}}(x, \langle ys \rangle, z_1, z_2) \to \langle \overline{\mathsf{split}}(x, ys, \bot, \bot) \rangle, \\ \overline{\mathsf{qsort}}(\langle xs \rangle, z_1) \to \langle \overline{\mathsf{qsort}}(xs, \bot) \rangle, & & \\ \langle x \rangle \le y \to \langle x \le y \rangle, & & x \le \langle y \rangle \to \langle x \le y \rangle, \\ \langle xs \rangle +\!\!+ ys \to \langle xs +\!\!+ ys \rangle, & & xs +\!\!+ \langle ys \rangle \to \langle xs +\!\!+ ys \rangle \end{array} \right\}$$

$\mathcal{R}_3$ is not a constructor system because of e.g., $\langle \langle x \rangle \rangle \to \langle x \rangle$, and thus, $\mathbb{SR}(\mathcal{R}_1)$ is not a constructor system, either, while $\mathcal{R}_1$ is so.

Rules in $\mathbb{U}(\mathcal{R})$ and $\mathbb{SR}(\mathcal{R}) \setminus \mathcal{R}_{aux}$ have some correspondence each other. An unconditional rule $l \to r \in \mathcal{R} \cap \mathbb{U}(\mathcal{R})$ is said to *correspond to* $\mathtt{ext}(l) \to \langle \overline{r} \rangle \in \mathbb{SR}(\mathcal{R})$, and vice versa; For the *i*-th conditional f-rule $\rho : \mathsf{f}(\overrightarrow{w_{1..n}}) \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k$, the j-th rule of $\mathbb{U}(\rho)$ in Definition 3.1 is said to *correspond to* the j-th rule of $\mathbb{SR}(\rho)$ in Definition 3.6, and vice versa.

One of the important properties of $\mathbb{SR}$ is that $\mathbb{U}(\mathcal{R})$ is WLL if and only if so is $\mathbb{SR}(\mathcal{R})$. By definition, this claim holds trivially.

Theorem 3.9 *$\mathcal{R}$ is $\mathbb{U}$-WLL if and only if $\mathbb{SR}(\mathcal{R})$ is WLL.*

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 $\langle \overline{s} \rangle \to^*_{\mathbb{SR}(\mathcal{R})} t$. It is clear that for any reachable term $t \in T(\overline{\mathcal{F}}, \mathcal{V})$, any term $t' \in T(\overline{\mathcal{F}}, \mathcal{V})$ with $t \to^*_{\mathbb{SR}(\mathcal{R})} t'$ is reachable. In the following, for the extended signature $\overline{\mathcal{F}}$, we only consider subterms of reachable terms because it suffices to consider them in discussing soundness. For brevity, subterms of reachable terms are also called *reachable*. In reachable terms, the introduced symbols $\bot$ and $[\;]^{\rho}_{i}$ appear at appropriate positions of a term, i.e., at the root position of the term or an i-th argument of a subterm $\overline{\mathsf{f}}(\ldots)$ where $i > n$ and f is an n-ary defined symbol.

4 Soundness for WLL and Ultra-WLL DCTRSs

In this section, we prove that $\mathbb{SR}$ is sound for WLL and $\mathbb{U}$-WLL DCTRSs. In the following, we use $\mathcal{R}$ as a $\mathbb{U}$-WLL DCTRS over a signature $\mathcal{F}$.

It would be possible to follow the proof shown in [12] for soundness of $\mathbb{SR}$ for WLL normal CTRSs. However, the proof is very long, and it is easy to guess that an analogous proof for WLL and $\mathbb{U}$-WLL DCTRSs—more complicated systems than normal CTRSs—becomes much longer. In this paper, we try to shorten the proof, providing a clearer one.

Our insight for a proof is that a term in $T(\overline{\mathcal{F}}, \mathcal{V})$ represents some corresponding terms in $T(\mathbb{U}(\mathcal{F}), \mathcal{V})$, and a derivation of $\mathbb{SR}(\mathcal{R})$ starting with $\langle \overline{s} \rangle$ ($s \in T(\mathcal{F}, \mathcal{V})$) represents the corresponding computation tree of $\mathbb{U}(\mathcal{R})$, whose root is s. We illustrate this observation by the following WLL and $\mathbb{U}$-WLL normal DCTRS:

$$\mathcal{R}_4 = \left\{ \begin{array}{lll} \mathsf{f}(x) \to \mathsf{c} \Leftarrow x \twoheadrightarrow \mathsf{c}, & \mathsf{a} \to \mathsf{c}, & \mathsf{b} \to \mathsf{c}, \\ \mathsf{f}(x) \to \mathsf{d} \Leftarrow x \twoheadrightarrow \mathsf{d}, & \mathsf{a} \to \mathsf{d}, & \mathsf{b} \to \mathsf{d}, \\ \mathsf{g}(x) \to \mathsf{h}(x,x), & \mathsf{h}(\mathsf{c},\mathsf{d}) \to \mathsf{c}, & \mathsf{h}(x, \mathsf{f}(x)) \to \mathsf{d} \end{array} \right\}$$

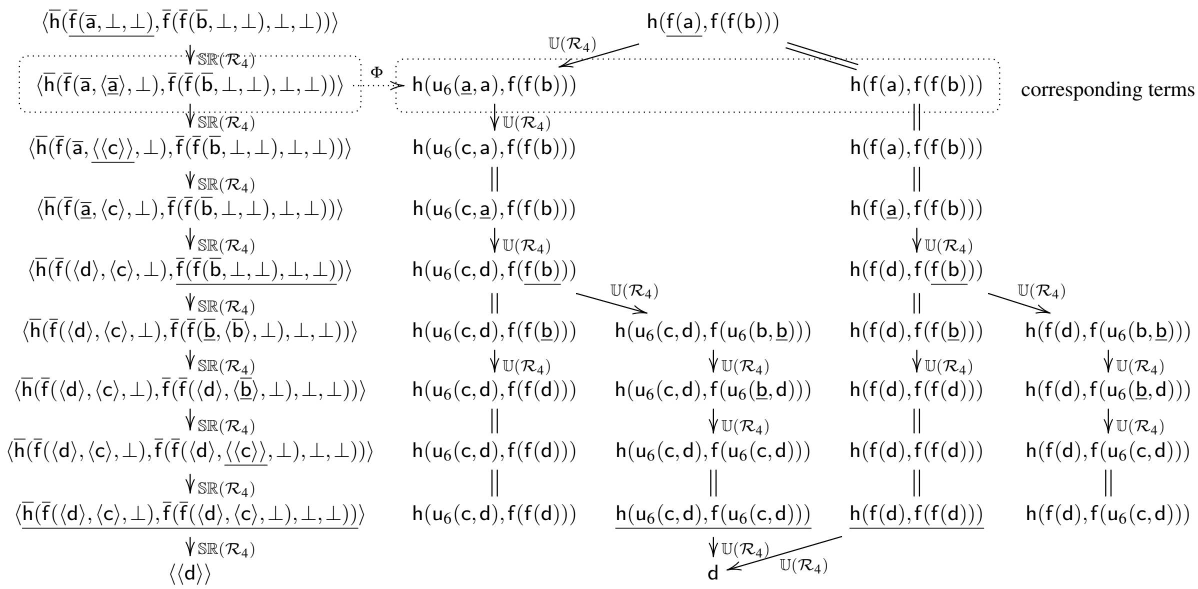

Figure 1: a derivation of $\mathbb{SR}(\mathcal{R}_4)$ and its corresponding computation tree (DAG) of $\mathbb{U}(\mathcal{R}_4)$.

To simplify the discussion, we use a normal CTRS, and omit $[\ ]^{\rho}_{j}$ introduced during the application of $\mathbb{SR}$. $\mathcal{R}_4$ is transformed by $\mathbb{U}$ and $\mathbb{SR}$, respectively, as follows:

$$\mathbb{U}(\mathcal{R}_4) = \left\{ \begin{array}{llll} \mathsf{f}(x) \to \mathsf{u}_6(x,x), & \mathsf{u}_6(\mathsf{c},x) \to \mathsf{c}, & \mathsf{a} \to \mathsf{c}, & \mathsf{b} \to \mathsf{c}, \\ \mathsf{f}(x) \to \mathsf{u}_7(x,x), & \mathsf{u}_7(\mathsf{d},x) \to \mathsf{d}, & \mathsf{a} \to \mathsf{d}, & \mathsf{b} \to \mathsf{d}, \\ \mathsf{g}(x) \to \mathsf{h}(x,x), & \mathsf{h}(\mathsf{c},\mathsf{d}) \to \mathsf{c}, & \mathsf{h}(x,\mathsf{f}(x)) \to \mathsf{d} & \end{array} \right\}$$

$$\mathbb{SR}(\mathcal{R}_4) = \left\{ \begin{array}{llll} \overline{\mathsf{f}}(x,\perp,z_2) \to \overline{\mathsf{f}}(x,\langle x\rangle,z_2), & \overline{\mathsf{f}}(x,\langle \mathsf{c}\rangle,z_2) \to \langle \mathsf{c}\rangle, & \overline{\mathsf{a}} \to \langle \mathsf{c}\rangle, & \overline{\mathsf{b}} \to \langle \mathsf{c}\rangle, \\ \overline{\mathsf{f}}(x,z_1,\perp) \to \overline{\mathsf{f}}(x,z_1,\langle x\rangle), & \overline{\mathsf{f}}(x,z_1,\langle \mathsf{d}\rangle) \to \langle \mathsf{d}\rangle, & \overline{\mathsf{a}} \to \langle \mathsf{d}\rangle, & \overline{\mathsf{b}} \to \langle \mathsf{d}\rangle, \\ \overline{\mathsf{g}}(x) \to \langle \overline{\mathsf{h}}(x,x)\rangle, & \overline{\mathsf{h}}(\mathsf{c},\mathsf{d}) \to \langle \mathsf{c}\rangle, & \overline{\mathsf{h}}(x,\overline{\mathsf{f}}(x,z_1,z_2)) \to \langle \mathsf{d}\rangle, & \dots \end{array} \right\}$$

Each reachable term in $T(\overline{\mathcal{F}},\mathcal{V})$ represents a finite set of terms in $T(\mathbb{U}(\mathcal{F}),\mathcal{V})$: $\overline{\mathsf{f}}(\overline{\mathsf{a}},\langle\mathsf{c}\rangle,\perp)$ represents two terms $\mathsf{f}(\mathsf{a})$ and $\mathsf{u}_6(\mathsf{c},\mathsf{a})$; $\langle\overline{\mathsf{h}}(\overline{\mathsf{f}}(\overline{\mathsf{a}},\langle\mathsf{c}\rangle,\perp),\overline{\mathsf{f}}(\overline{\mathsf{a}},\langle\mathsf{c}\rangle,\perp))\rangle$ represents four terms $\mathsf{h}(\mathsf{f}(\mathsf{a}),\mathsf{f}(\mathsf{a}))$, $\mathsf{h}(\mathsf{f}(\mathsf{a}),\mathsf{u}_6(\mathsf{c},\mathsf{a}))$, $\mathsf{h}(\mathsf{f}(\mathsf{a}),\mathsf{f}(\mathsf{a}))$, and $\mathsf{h}(\mathsf{u}_6(\mathsf{c},\mathsf{a}),\mathsf{u}_6(\mathsf{c},\mathsf{a}))$. These correspondence will be formalized as a mapping Φ from $T(\overline{\mathcal{F}},\mathcal{V})$ to $2^{T(\mathbb{U}(\mathcal{F}),\mathcal{V})}$ later. Figure 1 illustrates a derivation of $\mathbb{SR}(\mathcal{R}_4)$ and its corresponding computation tree (more precisely, a DAG) of $\mathbb{U}(\mathcal{R}_4)$ where reduced terms are underlined, and in each row, the leftmost term is the one appearing in the derivation of $\mathbb{SR}(\mathcal{R}_4)$ and the remaining are terms in $T(\mathbb{U}(\mathcal{F}),\mathcal{V})$ that are represented by the leftmost one.

We capture the observation above by a mapping defined below.

Definition 4.1 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS. Then, a mapping Φ from reachable terms (and lists of terms) in $T(\overline{\mathcal{F}},\mathcal{V})$ to $2^{T(\mathbb{U}(\mathcal{F}),\mathcal{V})}$ is recursively defined with an auxiliary mapping Ψ as follows:*

- $\Phi(x) = \{x\}$ *for* $x \in \mathcal{V}$,
- $\Phi(\overline{\mathsf{c}}(\overrightarrow{t_{1..n}})) = \{\mathsf{c}(\overrightarrow{t'_{1..n}}) \mid \overrightarrow{t'_{1..n}} \in \Phi(\overrightarrow{t_{1..n}})\}$ *for* $\mathsf{c}/n \in \mathcal{C}_{\mathcal{R}}$,
- $\Phi(\overline{\mathsf{f}}(\overrightarrow{t_{1..n}},\overrightarrow{u_{1..m}})) = \Psi(\overline{\mathsf{f}}(\overrightarrow{t_{1..n}},\perp,\dots,\perp)) \cup \bigcup_{i\in\{1,\dots,m\},u_i\neq\perp} \Psi(\overline{\mathsf{f}}(\overrightarrow{t_{1..n}},\perp^{i-1},u_i,\perp^{m-i})$ *for* $\mathsf{f}/n \in \mathcal{D}_{\mathcal{R}}$,

- $\Phi(\langle t \rangle) = \Phi(t)$,
- $\Phi(t) = \emptyset$ *where t is not of the form above,*
- $\Phi(\varepsilon) = \{\varepsilon\}$,
- $\Phi(\overrightarrow{t_{1..n}}) = \{\overrightarrow{t'_{1..n}} \mid t'_i \in \Phi(t_i),\ 1 \leq i \leq n\}$ *for $n > 1$,*
- $\Psi(\overline{\mathsf{f}}(\overrightarrow{t_{1..n}}, \bot, \ldots, \bot)) = \{\mathsf{f}(\overrightarrow{t'_{1..n}}) \mid \overrightarrow{t'_{1..n}} \in \Phi(\overrightarrow{t_{1..n}})\}$ *for $\mathsf{f}/n \in \mathcal{D}_\mathcal{R}$,*
- $\Psi(\overline{\mathsf{f}}(\overrightarrow{s'_{1..n}}, \bot^{i-1}, [\langle t' \rangle, \overrightarrow{t'_{1..n_j}}]^\rho_j, \bot^{m-i}) = \{U^\rho_j(u', \overrightarrow{u'_{1..|X_j|}}) \mid u' \in \Phi(t'),\ \overrightarrow{u'_{1..|X_j|}} \in \Phi(\sigma(\overrightarrow{X_j}))\}$ *for $\mathsf{f}/n \in \mathcal{D}_\mathcal{R}$, where $\rho : \mathsf{f}(\overrightarrow{w_{1..n}}) \to r \Leftarrow s_1 \twoheadrightarrow t_1, \ldots, s_k \twoheadrightarrow t_k \in \mathcal{R}$ is the i-th conditional rule of f, $U^\rho_j(t_j, \overrightarrow{X_j}) \to r' \in \mathbb{U}(\rho)$, $X_j = \mathcal{V}ar(\mathsf{f}(\overrightarrow{w_{1..n}}), \overrightarrow{t_{1..j-1}})$, $V_j = \mathcal{V}ar(\overrightarrow{t_{1..j-1}})$, and σ is a substitution such that $\sigma(\overrightarrow{V_j}) = \overrightarrow{t'_{1..n_j}}$ and $\sigma\left(\overrightarrow{\mathtt{ext}(w_{1..n})}\right) = \overrightarrow{s'_{1..n}}$, and*
- $\Psi(t) = \emptyset$ *where t is not of the form above.*

The mapping Φ is straightforwardly extended to substitutions that have only reachable terms in the range: $\Phi(\sigma) = \{\sigma' \mid \mathcal{D}om(\sigma') \subseteq \mathcal{D}om(\sigma), \forall x \in \mathcal{D}om(\sigma).\ x\sigma' \in \Phi(x\sigma)\}$.

In applying Φ to reachable terms, Φ is never applied to terms rooted by either $\bot$ or $[\ ]^\rho_j$. Though, to simplify proofs below, we define Φ for $[\ ]^\rho_j$: $\Phi([\overrightarrow{t_{1..n}}]^\rho_j) = \overrightarrow{\Phi(t_{1..n})}$. $\Phi([\overrightarrow{t_{1..n}}]^\rho_j)$ is a set of term sequences, and in the following, we are interested in $|\Phi([\overrightarrow{t_{1..n}}]^\rho_j)|$ rather than elements in $\Phi([\overrightarrow{t_{1..n}}]^\rho_j)$.

We say that a term $s \in T(\overline{\mathcal{F}}, \mathcal{V})$ *contains an evaluation of conditions* if s has a subterm of the form $\overline{\mathsf{f}}(\overrightarrow{t_{1..n}}, \ldots, [\ldots]^\rho_j, \ldots)$ for some $\mathsf{f}/n \in \mathcal{F}$. We say that a term $\overline{\mathsf{f}}(\overrightarrow{t_{1..n}}, \overrightarrow{u_{1..m}}) \in T(\overline{\mathcal{F}}, \mathcal{V})$ *cannot continue any evaluation of conditions at root position* if for any conditional f-rule $l \to r \Leftarrow c \in \mathcal{R}$, $\mathtt{ext}(l)$ does not match $\overline{\mathsf{f}}(\overrightarrow{t_{1..n}}, \overrightarrow{u_{1..m}})$. We also say that a term $s \in T(\overline{\mathcal{F}}, \mathcal{V})$ *cannot continue any evaluation of conditions* if any subterm of s, which is rooted by a symbol $\overline{\mathsf{f}}$ with $\mathsf{f} \in \mathcal{D}_\mathcal{R}$, cannot continue any evaluation of conditions at root position. For a term s, $|\Phi(s)| = 1$ means that s is mapped by Φ to a unique one in $T(\mathbb{U}(\mathcal{F}), \mathcal{V})$, i.e., s does not contain any evaluation or cannot continue any evaluation of conditions. For a substitution σ and a term t, $|\Phi(\sigma|_{\mathcal{V}ar(t)})| = 1$ means that for each variable x in t, the term substituted for x is mapped by Φ to a unique one in $T(\mathbb{U}(\mathcal{F}), \mathcal{V})$, i.e., $|\Phi(x\sigma)| = 1$. The mapping Φ has the following properties.

Lemma 4.2 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS, s be a term in $T(\mathcal{F}, \mathcal{V})$, t be a reachable term in $T(\overline{\mathcal{F}}, \mathcal{V})$, and σ be a substitution in $\mathcal{S}ub(\overline{\mathcal{F}}, \mathcal{V})$. Then, all of the following hold:*

(a) $\Phi(\overline{s}) = \{s\}$.

(b) $\widehat{t} \in \Phi(t)$ *(i.e., $|\Phi(t)| \geq 1$).*

(c) $\{t'\sigma' \mid t' \in \Phi(t),\ \sigma' \in \Phi(\sigma)\} \subseteq \Phi(t\sigma)$.

(d) If $\Phi(\sigma|_{\mathcal{V}ar(t)}) = \{\sigma'\}$ for some substitution σ' (i.e., $|\Phi(\sigma|_{\mathcal{V}ar(t)})| = 1$), then $\Phi(\widehat{\widehat{t}\,\sigma}) = \{\widehat{t}\sigma'\}$.

(e) If $\Phi(\sigma|_{\mathcal{V}ar(s)}) = \{\sigma'\}$ for some substitution σ' (i.e., $|\Phi(\sigma|_{\mathcal{V}ar(s)})| = 1$), then $\Phi(\overline{s}\sigma) = \{s\sigma'\}$.

Proof (Sketch). Claims (a) and (b) are trivial by definition. Claim (c) can be proved by structural induction on t. Claim (d) can be proved analogously to (c) using $\Phi(\sigma|_{\mathcal{V}ar(t)}) = \{\sigma'\}$. Claim (e) is trivial by (d). □

As illustrated in Figure 1, our idea is simple and intuitive. Unfortunately, however, the proof for soundness needs some technical lemmas, while the entire proof is simpler than that in [12].

Using the mapping Φ and soundness of $\mathbb{U}$ for WLL DCTRSs, we show that for a term $s_0 \in T(\mathcal{F},\mathcal{V})$ and a term $t \in T(\overline{\mathcal{F}},\mathcal{V})$, if $\langle\overline{s_0}\rangle \to^*_{\mathbb{SR}(\mathcal{R})} t$, then $s_0 \to^*_{\mathbb{U}(\mathcal{R})} \widehat{t} \in \Phi(t)$ (Lemma 4.8 and Theorem 4.9). Since $\Phi(\langle\overline{s_0}\rangle) = \{s_0\}$, to show this claim generally, it suffices to prove the subclaim that for all reachable terms s and t in $T(\overline{\mathcal{F}},\mathcal{V})$, if $s \to_{l\to r\in\mathbb{SR}(\mathcal{R})} t$, then for each term $t' \in \Phi(t)$, there exists a term $s' \in \Phi(s)$ such that $s' \to^*_{\mathbb{U}(\mathcal{R})} t'$. If t' does not contain a converted term obtained from the reduced subterm in t, then t' is also in $\Phi(s)$. Otherwise, for the single rewrite step $s \to_{l\to r\in\mathbb{SR}(\mathcal{R})} t$, one of the following three cases holds:

- The case where $l \to r$ is an auxiliary rule in $\mathcal{R}_{aux}$. In this case, $\Phi(s) \supseteq \Phi(t)$, and thus, the subclaim holds. For example, for any rewrite step by $\mathcal{R}_{aux}$ in Figure 1, each term in $\Phi(t)$ appears in $\Phi(s)$, i.e., for each node t' for $\Phi(t)$, there exists a node that is for $\Phi(s)$ and is connected with t' by the $=$-edge.
- The case where $l \to r$ is in $\mathbb{SR}(\mathcal{R}) \setminus \mathcal{R}_{aux}$ and r is linear. It is easy to find $s' \in \Phi(s)$ such that s' is reduced by the rule in $\mathbb{U}(\mathcal{R})$ corresponding to $l \to r$: $s' \to_{\mathbb{U}(\mathcal{R})} t'$. In summary, for each $t' \in \Phi(t)$, there exists a term $s' \in \Phi(s)$ such that $s' \;(= \cup \to_{\mathbb{U}(\mathcal{R})})\; t'$. For example, the DAG for $\mathbb{U}(\mathcal{R}_4)$ in Figure 1 has only $=$- or $\to_{\mathbb{U}(\mathcal{R}_4)}$-edges because there are only rewrite steps with RL rules in $\mathbb{SR}(\mathcal{R}_4)$.
- The remaining case where $l \to r$ is in $\mathbb{SR}(\mathcal{R}) \setminus \mathcal{R}_{aux}$ and r is *not* linear. The difficulty of proving the subclaim comes from this case. We will discuss the detail of the difficulty later.

In proving soundness of $\mathbb{SR}$, neither a variable with non-linear occurrences nor a non-constructor pattern in the left-hand sides in $\mathcal{R}$ is problematic. For example, $\langle\overline{\mathsf{h}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot),\overline{\mathsf{f}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot),\bot,\bot))\rangle$ in Figure 1 is reduced by $\mathbb{SR}(\mathcal{R}_4)$ to $\langle\langle\mathsf{d}\rangle\rangle$, but neither $\mathsf{h}(\mathsf{u}_6(\mathsf{c},\mathsf{d}),\mathsf{f}(\mathsf{f}(\mathsf{d})))$ nor $\mathsf{h}(\mathsf{f}(\mathsf{d}),\mathsf{f}(\mathsf{u}_6(\mathsf{c},\mathsf{d})))$ in $\Phi(\langle\overline{\mathsf{h}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot),\overline{\mathsf{f}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot),\bot,\bot))\rangle)$ can be reduced by $\mathbb{U}(\mathcal{R}_4)$ to d. Though, this is not a problem because for each converted term, we need the existence of an ancestor but not a descendant. Viewed in this light, non-left-linearity of rules is not a problem, but non-right-linearity of rules causes difficulty of proving soundness. On the other hand, $\langle\overline{\mathsf{h}}(\mathsf{d},\overline{\mathsf{f}}(\mathsf{d},\bot,\bot))\rangle$ is reduced by $\mathbb{SR}(\mathcal{R}_4)$ to $\overline{\mathsf{h}}(\mathsf{d},\overline{\mathsf{f}}(\mathsf{d},\langle\mathsf{d}\rangle,\bot))$, and then to $\langle\langle\mathsf{d}\rangle\rangle$. We cannot reduce $\mathsf{h}(\mathsf{d},\mathsf{u}_6(\mathsf{d},\mathsf{d}))$ in $\Phi(\langle\overline{\mathsf{h}}(\mathsf{d},\overline{\mathsf{f}}(\mathsf{d},\langle\mathsf{d}\rangle,\bot))\rangle)$ by the corresponding rule $\mathsf{h}(x,\mathsf{f}(x)) \to \mathsf{d}$ in $\mathbb{U}(\mathcal{R}_4)$ to d, but another term $\mathsf{h}(\mathsf{d},\mathsf{f}(\mathsf{d}))$ in $\Phi(\langle\overline{\mathsf{h}}(\mathsf{d},\overline{\mathsf{f}}(\mathsf{d},\langle\mathsf{d}\rangle,\bot))\rangle)$ can be reduced to d, simulating the step of $\mathbb{SR}(\mathcal{R}_4)$.

Let us get back to the case where we apply non-right-linear rules in $\mathbb{SR}(\mathcal{R}) \setminus \mathcal{R}_{aux}$. Figure 2 illustrates a derivation of $\mathbb{SR}(\mathcal{R}_4)$ and its corresponding computation tree of $\mathbb{U}(\mathcal{R}_4)$, where non-right-linear rules are applied. In applying non-right-linear rules in $\mathbb{SR}(\mathcal{R}) \setminus \mathcal{R}_{aux}$ to s with $s \to_{l\to r\in\mathbb{SR}(\mathcal{R})\setminus\mathcal{R}_{aux}} t$, it is not only difficult but also sometimes impossible to show that for each $t' \in \Phi(t)$, there exists a term $s' \in \Phi(s)$ such that $s' \to^*_{\mathbb{U}(\mathcal{R})} t'$. For example, $\langle\overline{\mathsf{g}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot))\rangle$ in Figure 2 is reduced by $\mathbb{SR}(\mathcal{R}_4)$ to $\langle\langle\overline{\mathsf{h}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot),\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot))\rangle\rangle$, and represents two terms in $T(\mathbb{U}(\mathcal{F}),\mathcal{V})$: $\Phi(\langle\overline{\mathsf{g}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot))\rangle) = \{\ \mathsf{g}(\mathsf{f}(\mathsf{d})),\ \mathsf{g}(\mathsf{u}_6(\mathsf{c},\mathsf{d}))\ \}$. Though, no term in $\Phi(\langle\overline{\mathsf{g}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot))\rangle)$ is reduced by a single step of $\to_{\mathbb{SR}(\mathcal{R}_4)}$ to either $\mathsf{h}(\mathsf{f}(\mathsf{d}),\mathsf{u}_6(\mathsf{c},\mathsf{d}))$ or $\mathsf{h}(\mathsf{u}_6(\mathsf{c},\mathsf{d}),\mathsf{f}(\mathsf{d}))$ in $\Phi(\langle\langle\overline{\mathsf{h}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot),\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot))\rangle\rangle)$. However, $\mathsf{g}(\mathsf{f}(\mathsf{a})) \in \Phi(\langle\overline{\mathsf{g}}(\overline{\mathsf{f}}(\mathsf{a},\langle\mathsf{c}\rangle,\bot))\rangle)$ can be reduced to both $\mathsf{h}(\mathsf{f}(\mathsf{d}),\mathsf{u}_6(\mathsf{c},\mathsf{d}))$ and $\mathsf{h}(\mathsf{u}_6(\mathsf{c},\mathsf{d}),\mathsf{f}(\mathsf{d}))$ including rewrite steps of $\mathsf{g}(x) \to \mathsf{h}(x,x) \in \mathbb{U}(\mathcal{R}_4)$ corresponding to $\overline{\mathsf{g}}(x) \to \langle\overline{\mathsf{h}}(x,x)\rangle \in \mathbb{SR}(\mathcal{R}_4)$. The existence of such reduction sequences represented by $\to^+_{\mathbb{U}(\mathcal{R}_4)}$-edges of the DAG in Figure 2 is ensured by the reduction $\langle\overline{\mathsf{g}}(\overline{\mathsf{f}}(\overline{\mathsf{a}},\langle\mathsf{c}\rangle,\bot))\rangle \to^*_{\mathbb{SR}(\mathcal{R}_4)} \langle\langle\overline{\mathsf{h}}(\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot),\overline{\mathsf{f}}(\langle\mathsf{d}\rangle,\langle\mathsf{c}\rangle,\bot))\rangle\rangle$. In summary, for $s_0 \to^*_{\mathbb{SR}(\mathcal{R})} s \to_{l\to r\in\mathbb{SR}(\mathcal{R})\setminus\mathcal{R}_{aux}} t$, we will show the existence of a derivation $s_0 \to^*_{\mathbb{SR}(\mathcal{R})} s' \to_{l\to r\in\mathbb{SR}(\mathcal{R})\setminus\mathcal{R}_{aux}} t' \to^*_{\mathbb{SR}(\mathcal{R})} t$ such that for each term $t'' \in \Phi(t')$, $s'' \to_{l'\to r'\in\mathbb{U}(\mathcal{R})} t''$ for some term $s'' \in \Phi(s')$, where $l' \to r'$ corresponds to $l \to r$ (Lemmas 4.6 and 4.7).

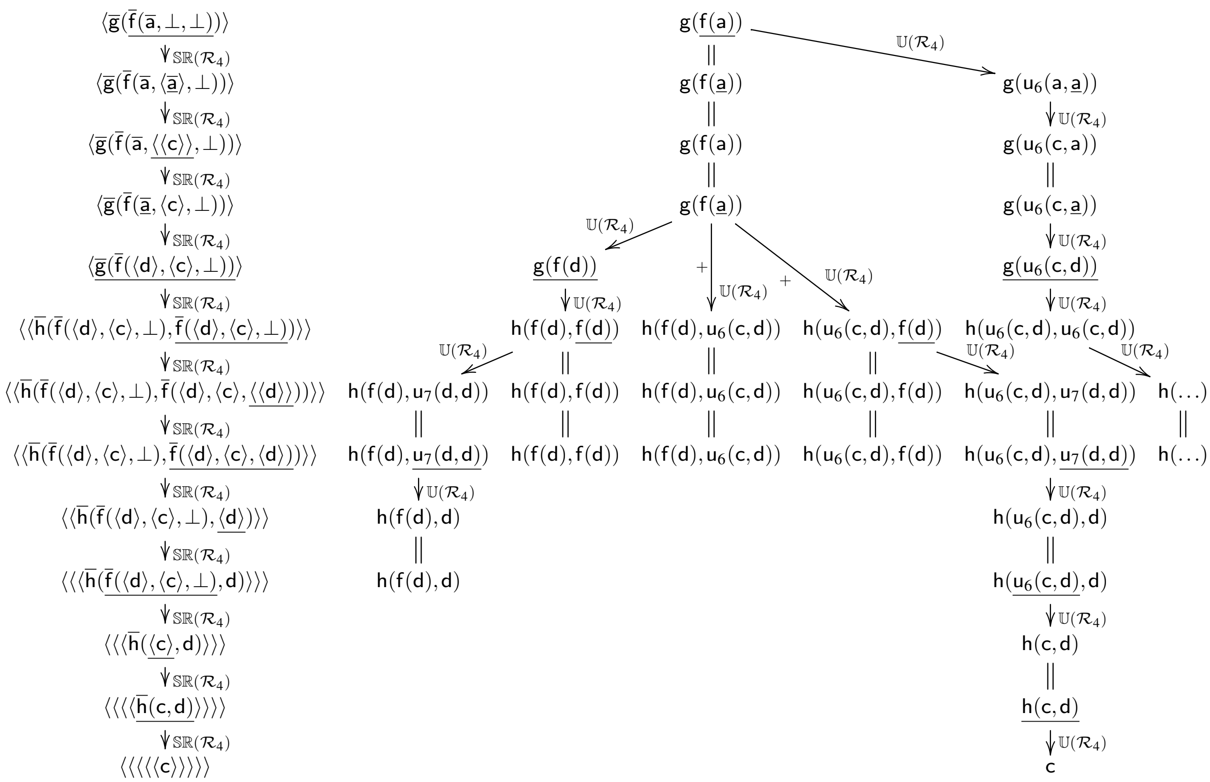

Figure 2: another derivation of $\mathbb{SR}(\mathcal{R}_4)$ and its corresponding computation tree (DAG) of $\mathbb{U}(\mathcal{R}_4)$.

Before showing the key lemmas (Lemmas 4.6 and 4.7), we show some auxiliary lemmas along the intuition above. The following lemma says that if a term t is reduced and matches a linear pattern obtained from $\mathcal{R}$ by applying $\mathtt{ext}$, then the initial term t also matches the pattern.

Lemma 4.3 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS, t be a reachable term in $T(\overline{\mathcal{F}},\mathcal{V})$, w be a linear term in $T(\mathcal{F},\mathcal{V})$, $w' = \mathtt{ext}(w)$ (i.e., $w' \in T(\overline{\mathcal{F}} \setminus \{\langle\rangle, \bot, [\,]^{\rho}_{j}\}, \mathcal{V})$),[3] and θ be a substitution in $\mathcal{S}ub(\overline{\mathcal{F}},\mathcal{V})$. If $t \to^*_{\mathbb{SR}(\mathcal{R})} w'\theta$, then there exists a substitution σ such that $t = w'\sigma$ and $x\sigma \to^*_{\mathbb{SR}(\mathcal{R})} x\theta$ for all variables $x \in \mathcal{V}ar(w)$.*

Proof (Sketch). It suffices to show that if $t \to_{p,l\to r\in\mathbb{SR}(\mathcal{R})} w'\theta$, then there exists a substitution σ such that $t = w'\sigma$ and $x\sigma \to^*_{\mathbb{SR}(\mathcal{R})} x\theta$ for all variables $x \in \mathcal{V}ar(w)$. This claim can be proved by structural induction on w. □

For the sake of readability, we introduce a binary relation $\Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})}$ over $T(\overline{\mathcal{F}},\mathcal{V})$: $s \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} t$ if and only if for each $t' \in \Phi(t)$, there exists a term $s' \in \Phi(s)$ such that $s' \to^*_{\mathbb{U}(\mathcal{R})} t'$. It is clear that $\Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})}$ is reflexive and transitive. The relation $\Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})}$ is closed under contexts.

Lemma 4.4 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS, $C[\,]$ be a context, and s and t be terms in $T(\overline{\mathcal{F}},\mathcal{V})$ such that $s \to^*_{\mathbb{SR}(\mathcal{R})} t$ and $s \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} t$. Then, $C[s] \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} C[t]$.*

[3] Patterns $w'_1,\ldots,w'_n,\mathtt{ext}(t_1),\ldots,\mathtt{ext}(t_k)$ in Definition 3.6 are in $T(\overline{\mathcal{F}}/\{\langle\rangle,\bot,[\,]^{\rho}_{j}\},\mathcal{V})$.

Proof (Sketch). Using the definition of Φ and Lemma 4.3, this lemma can be proved by structural induction on $C[\,]$. □

The following lemma is a variant of Lemma 4.4 that is useful to prove the main key lemma shown later.

Lemma 4.5 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS, t be a term in $T(\overline{\mathcal{F}},\mathcal{V})$, and σ and θ be substitutions such that $x\sigma \to^*_{\mathbb{SR}(\mathcal{R})} x\theta$ and $x\sigma \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} x\theta$ for all variables $x \in \mathcal{V}ar(t)$. Then, $t\sigma \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} t\theta$.*

Proof (Sketch). It is easy to extend Lemma 4.4 to contexts with multiple holes. Thus, this lemma is a direct consequence of the extended lemma since a linear term can be considered a context with multiple holes. □

When $l \to r \in \mathbb{SR}(\mathcal{R})$ has a variable x such that $|r|_x > 1$ and $|\Phi(x\theta)| > 1$, we have at least two terms obtained by converting $x\theta$, and thus, $\Phi(r\sigma)$ contains a term that has no ancestor in $\Phi(l\theta)$ w.r.t. $\to_{\mathbb{U}(\mathcal{R})}$. This problem does not happen if $\Phi(\theta|_{\mathcal{V}ar(r)})$ is a singleton set.

Lemma 4.6 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS, $l \to r \in \mathbb{SR}(\mathcal{R})$, and σ be a substitution such that for any variable $x \in \mathcal{V}ar(r)$, if $|r|_x > 1$ and $x\sigma \neq \bot$ then $|\Phi(x\sigma)| = 1$. Then, $l\sigma \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} r\sigma$.*

Proof (Sketch). Referring to the definition of Φ and Lemma 4.2, this lemma can be proved by a case distinction depending on what $l \to r$ is. □

For a derivation $s \to^*_{\mathbb{SR}(\mathcal{R})} t\theta$, the following lemma ensures the existence of an ancestor for a variable x in t such that $|t|_x = 1$ and $|\Phi(x\theta)| > 1$.

Lemma 4.7 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS, s and t be terms in $T(\overline{\mathcal{F}},\mathcal{V})$, θ be a substitution in $\mathcal{S}ub(\overline{\mathcal{F}},\mathcal{V})$, and $X \subseteq \{x \in \mathcal{V}ar(t) \mid |t|_x = 1\}$. If $|\Phi(s)| = 1$ and $s \to^d_{\mathbb{SR}(\mathcal{R})} t\theta$ ($d \geq 0$), then there exist a substitution $\delta \in \mathcal{S}ub(\overline{\mathcal{F}},\mathcal{V})$ and natural numbers d' and d_x for $x \in X$ such that*

- *(a)* $d' + \sum_{x\in X} d_x \leq d$,
- *(b)* $s \to^{d'}_{\mathbb{SR}(\mathcal{R})} t\delta$,
- *(c)* $|\Phi(x\delta)| = 1$ *and* $x\delta \to^{d_x}_{\mathbb{SR}(\mathcal{R})} x\theta$ *for all variables* $x \in X$ *such that* $x\theta \neq \bot$, *and*
- *(d)* $x\delta = x\theta$ *for all variables* $x \in \mathcal{V}ar(t) \setminus \{y \in X \mid y\theta \neq \bot\}$.[4]

Proof (Sketch). Using Theorem 3.9, Lemma 4.2 (a), (c), (d), and (e), and Lemma 4.3, this lemma can be proved by induction on the lexicographic product of d and the size of s. □

We show the main key lemma on the relationship between $\to^*_{\mathbb{SR}(\mathcal{R})}$ and $\Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})}$.

Lemma 4.8 *Let $\mathcal{R}$ be a $\mathbb{U}$-WLL DCTRS, and s and t be terms in $T(\overline{\mathcal{F}},\mathcal{V})$. If $|\Phi(s)| = 1$ and $s \to^d_{\mathbb{SR}(\mathcal{R})} t$ ($d \geq 0$), then $s \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} t$.*

Proof (Sketch). Using Theorem 3.9, Lemmas 4.4, 4.6 and 4.7, this lemma can be proved by induction on d. □

Finally, we show the key result of this paper.

[4] Note that if $x\theta = \bot$, then $x\delta = \bot$.

Theorem 4.9 *$\mathbb{SR}$ is sound for WLL and $\mathbb{U}$-WLL DCTRSs.*

Proof. Let $\mathcal{R}$ be a WLL and $\mathbb{U}$-WLL DCTRS over a signature $\mathcal{F}$, $s \in T(\mathcal{F},\mathcal{V})$, and $t \in T(\overline{\mathcal{F}},\mathcal{V})$. Suppose that $\langle \overline{s} \rangle \to^*_{\mathbb{SR}(\mathcal{R})} t$. It follows from Lemma 4.8 that $\langle \overline{s} \rangle \Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})} t$. It follows from Lemma 4.2 (a), (b) that $\Phi(\langle \overline{s} \rangle) = \Phi(\overline{s}) = \{s\}$ and $\widehat{t} \in \Phi(t)$, and hence, by the definition of $\Rightarrow_{\Phi,\mathbb{U}(\mathcal{R})}$, $s \to^*_{\mathbb{U}(\mathcal{R})} \widehat{t}$. Since $\mathbb{U}$ is sound for $\mathcal{R}$ by Theorem 3.3, it holds that $s \to^*_{\mathcal{R}} \widehat{t}$. Therefore, $\mathbb{SR}$ is sound for $\mathcal{R}$. □

Let us consider the conversion $\mathbb{T}$ in Section 3.1 again. As a consequence of Theorems 3.5 and 4.9, we show that the composed transformation $\mathbb{SR} \circ \mathbb{T}$ of a WLL DCTRS into a WLL TRS is sound for WLL DCTRSs.

Theorem 4.10 *The composed transformation $\mathbb{SR} \circ \mathbb{T}$ is sound and complete for WLL DCTRSs.*

5 Conclusion

In this paper, we have shown that every WLL DCTRS can be converted to an equivalent WLL and $\mathbb{U}$-WLL DCTRS and the SR transformation is applicable to $\mathbb{U}$-WLL DCTRSs without any change. Then, we have proved that the SR transformation is sound for WLL and $\mathbb{U}$-WLL DCTRSs. As a consequence of these results, we have shown that the composition of the conversion and the SR transformation is a sound structure-preserving transformation for WLL DCTRSs. For computational equivalence to WLL SDCTRSs, we have to show that if $\mathcal{R}$ is confluent, then so is $\mathbb{SR}(\mathcal{R})$. To prove this claim as in [16] is one of our future works. To expand the applicability of the SR transformation, we will extend the SR transformation to other classes.

Acknowledgements We thank the anonymous reviewers very much for their useful comments to improve this paper.

References

[1] Franz Baader & Tobias Nipkow (1998): *Term Rewriting and All That*. Cambridge University Press, doi:10.1145/505863.505888.

[2] Francisco Durán, Salvador Lucas, José Meseguer, Claude Marché & Xavier Urbain (2004): *Proving termination of membership equational programs*. In Nevin Heintze & Peter Sestoft, editors: *Proceedings of the 2004 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-based Program Manipulation*, ACM, pp. 147–158, doi:10.1145/1014007.1014022.

[3] Guillaume Feuillade & Thomas Genet (2003): *Reachability in Conditional Term Rewriting Systems*. *Electronic Notes in Theoretical Computer Science* 86(1), pp. 133–146, doi:10.1016/S1571-0661(04)80658-3.

[4] Karl Gmeiner & Bernhard Gramlich (2009): *Transformations of Conditional Rewrite Systems Revisited*. In Andrea Corradini & Ugo Montanari, editors: *Proceedings of the 19th International Workshop on Recent Trends in Algebraic Development Techniques*, *Lecture Notes in Computer Science* 5486, Springer, pp. 166–186, doi:10.1007/978-3-642-03429-9_12.

[5] Karl Gmeiner, Bernhard Gramlich & Felix Schernhammer (2010): *On (Un)Soundness of Unravelings*. In Christopher Lynch, editor: *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, *Leibniz International Proceedings in Informatics* 6, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 119–134, doi:10.4230/LIPIcs.RTA.2010.119.

[6] Karl Gmeiner, Bernhard Gramlich & Felix Schernhammer (2012): *On Soundness Conditions for Unraveling Deterministic Conditional Rewrite Systems*. In Ashish Tiwari, editor: *Proceedings of the 23rd International Conference on Rewriting Techniques and Applications*, *Leibniz International Proceedings in Informatics* 15, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 193–208, doi:10.4230/LIPIcs.RTA.2012.193.

[7] Karl Gmeiner & Naoki Nishida (2014): *Notes on Structure-Preserving Transformations of Conditional Term Rewrite Systems*. In Manfred Schmidt-Schauß, Masahiko Sakai, David Sabel & Yuki Chiba, editors: *Proceedings of the first International Workshop on Rewriting Techniques for Program Transformations and Evaluation*, *OpenAccess Series in Informatics* 40, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 3–14, doi:10.4230/OASIcs.WPTE.2014.3.

[8] Salvador Lucas, Claude Marché & José Meseguer (2005): *Operational termination of conditional term rewriting systems*. *Information Processing Letters* 95(4), pp. 446–453, doi:10.1016/j.ipl.2005.05.002.

[9] Massimo Marchiori (1996): *Unravelings and Ultra-properties*. In Michael Hanus & Mario Rodríguez-Artalejo, editors: *Proceedings of the 5th International Conference on Algebraic and Logic Programming*, *Lecture Notes in Computer Science* 1139, Springer, pp. 107–121, doi:10.1007/3-540-61735-3_7.

[10] Massimo Marchiori (1997): *On Deterministic Conditional Rewriting*. Computation Structures Group, Memo 405, MIT Laboratory for Computer Science.

[11] Naoki Nishida, Masahiko Sakai & Toshiki Sakabe (2012): *Soundness of Unravelings for Conditional Term Rewriting Systems via Ultra-Properties Related to Linearity*. *Logical Methods in Computer Science* 8(3), pp. 1–49, doi:10.2168/LMCS-8(3:4)2012.

[12] Naoki Nishida, Makishi Yanagisawa & Karl Gmeiner (2014): *On Proving Soundness of the Computationally Equivalent Transformation for Normal Conditional Term Rewriting Systems by Using Unravelings*. In Manfred Schmidt-Schauß, Masahiko Sakai, David Sabel & Yuki Chiba, editors: *Proceedings of the First International Workshop on Rewriting Techniques for Program Transformations and Evaluation*, *OpenAccess Series in Informatics* 40, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 39–50, doi:10.4230/OASIcs.WPTE.2014.39.

[13] Enno Ohlebusch (2001): *Termination of Logic Programs: Transformational Methods Revisited*. *Applicable Algebra in Engineering, Communication and Computing* 12(1/2), pp. 73–116, doi:10.1007/s002000100064.

[14] Enno Ohlebusch (2002): *Advanced Topics in Term Rewriting*. Springer, doi:10.1007/978-1-4757-3661-8.

[15] Traian-Florin Şerbănuţă & Grigore Roşu (2006): *Computationally Equivalent Elimination of Conditions*. In Frank Pfenning, editor: *Proceedings of the 17th International Conference on Rewriting Techniques and Applications*, *Lecture Notes in Computer Science* 4098, Springer, pp. 19–34, doi:10.1007/11805618_3.

[16] Traian-Florin Şerbănuţă & Grigore Roşu (2006): *Computationally Equivalent Elimination of Conditions*. Technical Report UIUCDCS-R-2006-2693, Department of Computer Science, University of Illinois at Urbana-Champaign.

[17] Taro Suzuki, Aart Middeldorp & Tetsuo Ida (1995): *Level-Confluence of Conditional Rewrite Systems with Extra Variables in Right-Hand Sides*. In Jieh Hsiang, editor: *Proceedings of the 6th International Conference on Rewriting Techniques and Applications*, *Lecture Notes in Computer Science* 914, Springer, pp. 179–193, doi:10.1007/3-540-59200-8_56.

[18] Patrick Viry (1999): *Elimination of Conditions*. *Journal of Symbolic Computation* 28(3), pp. 381–401, doi:10.1006/jsco.1999.0288.