December 27, 2005

Partial Inversion of Constructor Term Rewriting Systems ⋆

Naoki Nishida, Masahiko Sakai, and Toshiki Sakabe

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

Abstract. Partial-inversion compilers generate programs which compute some unknown inputs of given programs from a given output and the rest of inputs whose values are already given. In this paper, we propose a partial-inversion compiler of constructor term rewriting systems. The compiler automatically generates a conditional term rewriting system, and then unravels it to an unconditional system. To improve the efficiency of inverse computation, we show that innermost strategy is usable to obtain all solutions if the generated system is right-linear.

1 Introduction

Roughly speaking, an *inverse* of a program $\mathtt{P}$ with one argument is a program $\mathtt{P}'$ such that $\mathtt{P}(\mathtt{a}) = \mathtt{b}$ and $\mathtt{P}'(\mathtt{b}) = \mathtt{a}$ coincide for any data $\mathtt{a}$ and $\mathtt{b}$. In case of a program $\mathtt{P}$ with two arguments, its *full inverse* is a program $\mathtt{P}'$ such that $\mathtt{P}(\mathtt{a},\mathtt{b}) = \mathtt{c}$ and $\mathtt{P}'(\mathtt{c}) = (\mathtt{a},\mathtt{b})$ coincide for any data $\mathtt{a}$, $\mathtt{b}$ and $\mathtt{c}$. On the other hand, a *partial inverse* of $\mathtt{P}$ with respect to the first argument is a program $\mathtt{P}''$ such that $\mathtt{P}(\mathtt{a},\mathtt{b}) = \mathtt{c}$ and $\mathtt{P}''(\mathtt{c},\mathtt{a}) = \mathtt{b}$ coincide.

We can find inverse programs in practical cases. Data compression and extraction commands (for example, `gzip` and `gunzip`) are examples of a program with one argument and its inverse. For a cryptographic encoder $\mathtt{E}(x,k)$ with a symmetric key k, the decoder $\mathtt{D}(y,k)$ can be seen as a partial inverse of $\mathtt{E}(x,k)$ with respect to the second argument. Other typical examples of partial-inverses are subtraction for addition, and division for multiplication.

The *inversion* sometimes helps us to generate a program from a specification given by equations. For example, the function gcd that computes the greatest common divisor is specified by equations $\mathsf{gcd}(x{+}y,y) = \mathsf{gcd}(x,y)$, $\mathsf{gcd}(x,\mathsf{0}) = x$ and $\mathsf{gcd}(x,y) = \mathsf{gcd}(y,x)$. From these equations, we can construct the following program:

$$\begin{cases} \mathsf{gcd}(z,y) \to \mathsf{gcd}(z-y,y), \\ \mathsf{gcd}(x,\mathsf{0}) \to x, \\ \mathsf{gcd}(x,y) \to \mathsf{gcd}(y,x). \end{cases} \qquad \begin{array}{c} x - \mathsf{0} \to x, \\ \mathsf{s}(x) - \mathsf{s}(y) \to x - y, \end{array}$$

⋆ This work is partly supported by MEXT. KAKENHI #15500007 and #16300005.

This is a revised and full version of the paper in *Proceedings of RTA 2005*, LNCS **3467**, Springer-Verlag, pp. 264–278, 2005.

Subtraction used in the above program is the partial inverse of addition in the specification. In this case, we fortunately know the definition of the subtraction. However, we do not always know the definition of the needed inverse. For example, it is not easy to code division as a recursive function by hand, while the function of division is well-known.

Can we automatically construct an inverse from a given program? This naive question motivates the study of *inversion compilers*. A (partial-)inversion compiler is an algorithm which automatically generates a (partial) inverse for a given program. In this paper, we present a partial-inversion compiler in the framework of term rewriting systems (TRSs), prove its correctness, and discuss the efficiency of the computation done by inverses generated by the compiler.

The partial-inversion compiler which we propose generates a partial-inverse EV-TRS (a TRS with *extra variables* that appear only in right-hand sides of rewrite rules) from a given constructor TRS. The compiler consists of two stages. In the first stage, the compiler generates a conditional TRS (a CTRS) as a partial-inverse program. In the second stage, the compiler transforms it to an equivalent EV-TRS, being based on the method of *unraveling* [14, 17, 18]. We prove the correctness of the compiler, that is, the generated EV-TRS is really a partial-inverse of the given TRS.

In inverse computation, it is not easy to handle values which are erased in the forward computation. In order to represent a guess of such values, we exploit extra variables. Although the reduction of an EV-TRS is essentially infinitely-branching and non-terminating, the reduction can be simulated by *narrowing* sequences starting from ground terms [16]. The termination problem of such narrowing is closer to that of the TRS reduction than to that of the ordinary narrowing. Therefore, the existence of extra variables in the generated systems is not disadvantageous for the compiler in this paper. The compiler sometimes generates a TRS (which has no extra variable), and it is terminating if it is lucky.

In logic programs like Prolog, inverse computation is realized by narrowing-based computation. However, the execution of inverse computation in Prolog does not terminate in some practical cases, when solving erased values in the forward computation. By contrast, the corresponding generated EV-TRS generated by our compiler terminates and gives all solutions by the depth-first search.

Unravelings bring a problem into the compiler. The reductions of CTRSs cannot be always simulated by the unraveled CTRSs completely. Unravelings [14, 18] which are developed in order to analyze properties of CTRSs, are not suitable for the simulation of the CTRS reduction. In this paper, we show that the combination of *membership conditional* [24] and *context-sensitive* [12] reductions assures the completeness. We also refer to syntactic conditions of CTRSs [17] for the completeness.

Another important issue for inverse computation is the efficiency. It is often necessary to find all normal forms of a given term. However, the exhaustive search is inefficient. In this paper, we show that the generated EV-TRSs always satisfy *ILRJ* property which is a part of conditions in [20] for the completeness of the innermost strategy on TRSs.

This paper is organized as follows. Section 2 prepares notations. In Section 3, we define a partial-inversion compiler of constructor TRSs, and prove its correctness. In Section 4, we discuss the computation of the generated EV-TRSs, and in Section 5 the improvement of the efficiency of partial-inverse computation. In Section 6, we refer to some related works and give concluding remarks. The missing proofs of theorems will be found in the appendix of this paper.

2 Preliminaries

This paper follows general notations of term rewriting [2, 19].

Let $\mathcal{V}$ be a countably infinite set of *variables*. The set of all *terms* over a *signature* $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F},\mathcal{V})$. Especially, we abbreviate the set $\mathcal{T}(\mathcal{F},\emptyset)$ of all *ground terms* to $\mathcal{T}(\mathcal{F})$. The set of variables occurring in one of terms $t_1,\ldots,t_n$ is represented by $\mathcal{V}ar(t_1,\ldots,t_n)$. *Identity* of terms is denoted by $\equiv$. The *root* symbol of a term t is represented by $\mathtt{root}(t)$. For a *context* $C[,\ldots,]$ with n holes $\Box$ at positions $p_1,\ldots,p_n$ and for terms $t_1,\ldots,t_n$, the notation $C[t_1,\ldots,t_n]_{p_1,\ldots,p_n}$ (or simply $C[t_1,\ldots,t_n]$) denotes the term obtained by replacing $\Box$ at p_i with t_i for $i=1,\ldots,n$. We denote the (proper) *subterm relation* by $\trianglelefteq$ ($\triangleleft$). The term $\sigma(t)$ obtained by applying a substitution σ to a term t is abbreviated to $t\sigma$. The *composition* $\sigma\sigma'$ of substitutions σ and σ' is defined as $t\sigma\sigma' \equiv \sigma'(\sigma(t))$.

An *(oriented) conditional rewrite rule* $l \to r \Leftarrow c$ over a signature $\mathcal{F}$ consists of the left-hand side l ($\in \mathcal{T}(\mathcal{F},\mathcal{V}) \setminus \mathcal{V}$) (lhs), the right-hand side r ($\in \mathcal{T}(\mathcal{F},\mathcal{V})$) (rhs) and the conditional part c which is a sequence $s_1 \to t_1,\ldots,s_k \to t_k$ of *oriented conditions* with $\{s_1,t_1,\ldots,s_k,t_k\} \subseteq \mathcal{T}(\mathcal{F},\mathcal{V})$. We write $l \to r$ instead of $l \to r \Leftarrow c$ if it is unconditional (that is, if $k=0$). We denote by $\rho : l \to r \Leftarrow c$ the rule $l \to r \Leftarrow c$ with a unique label ρ. The set $\mathcal{EV}ar(\rho)$ of all *extra variables* of $\rho : l \to r \Leftarrow c$ is defined as $\mathcal{EV}ar(\rho) = \mathcal{V}ar(r,c) \setminus \mathcal{V}ar(l)$.

An *(oriented) conditional rewriting system* (CTRS) over a signature $\mathcal{F}$ is a finite set of oriented conditional rewrite rules over $\mathcal{F}$. A CTRS is a *term rewriting system with extra variables* (EV-TRS) if its every rule is unconditional, and it is a *term rewriting system* (TRS) if it is an EV-TRS without extra variables. We use $R^{\Leftarrow}$ and $S^{\Leftarrow}$ for CTRSs (possibly EV-TRSs), and R and S for EV-TRSs. For a CTRS $R^{\Leftarrow}$, the *rewrite relation* of $R^{\Leftarrow}$ is denoted by $\to_{R^{\Leftarrow}}$. To specify the position p for $s \to_{R^{\Leftarrow}} t$, we write $s \to^{p}_{R^{\Leftarrow}} t$. The set of all *normal forms* of $R^{\Leftarrow}$ is denoted by $NF_{R^{\Leftarrow}}(\mathcal{F},\mathcal{V})$. A conditional rewrite rule $\rho : l \to r \Leftarrow c$ is classified according to the distribution of variables among l, r and c, as follows: ρ is in *type 3* if $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l,c)$, and in *type 4* if no restriction is imposed. An *i-CTRS* contains only conditional rewrite rules of type i. A CTRS $R^{\Leftarrow}$ is said to be *deterministic* if every rule $\rho : l \to r \Leftarrow s_1 \to t_1,\ldots,s_k \to t_k \in R^{\Leftarrow}$ is deterministic, that is, $\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l,t_1,\ldots,t_{i-1})$ for $1 \le i \le k$.

The set of *defined symbols* for a CTRS $R^{\Leftarrow}$ over a signature $\mathcal{F}$ is $\mathcal{D}_{R^{\Leftarrow}} = \{\, \mathtt{root}(l) \mid l \to r \Leftarrow c \in R^{\Leftarrow} \,\}$. The signature is partitioned as $\mathcal{F} = \mathcal{D}_{R^{\Leftarrow}} \uplus \mathcal{C}_{R^{\Leftarrow}}$ where $\uplus$ is the disjoint union of sets. Function symbols in $\mathcal{C}_{R^{\Leftarrow}}$ are called *constructors* of $R^{\Leftarrow}$. A term t in $\mathcal{T}(\mathcal{C}_{R^{\Leftarrow}},\mathcal{V})$ is called a *constructor term* of $R^{\Leftarrow}$. A *constructor system* is a CTRS $R^{\Leftarrow}$ such that every rule $f(t_1,\ldots,t_n) \to r \Leftarrow c \in R^{\Leftarrow}$ satisfies $\{t_1,\ldots,t_n\} \subseteq \mathcal{T}(\mathcal{C}_{R^{\Leftarrow}},\mathcal{V})$. A CTRS $R^{\Leftarrow}$ is *convergent* if it is confluent and strongly normalizing with respect to $\to_{R^{\Leftarrow}}$. We use a, b, c as constructors, f, g, h as defined symbols, and x, y, z as variables.

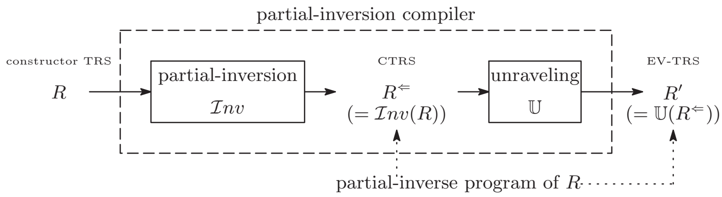

Fig. 1. The structure of the partial-inversion compiler proposed in this paper.

3 Partial-Inversion Compiler

In this section, we present a partial-inversion compiler of constructor TRSs which consists of two stages (Fig. 1). The first stage is an actual partial-inversion which generates a partial-inverse CTRS from a given constructor TRS. The second is the unraveling which transforms the CTRS to an equivalent EV-TRS [19].

We prepare special constructors $\{\mathsf{tp}_0, \mathsf{tp}_1, \ldots\}$ to denote the tuple $(t_1, \ldots, t_i)$ of terms $t_1, \ldots, t_i$ as $\mathsf{tp}_i(t_1, \ldots, t_i)$. The tuple $\mathsf{tp}_1(t)$ of a term t may be abbreviated to the term t. The reason why introducing such symbols is that inverses of n-ary functions return tuples of some terms.

An *index* for an n-ary defined symbol f is a natural number i such that $1 \leq i \leq n$, which intuitively stands for an argument position of f. We use sets of indexes for f to represent which arguments of f are given. For a set I of indexes for f, we denote by $|I|$ the cardinality of I. The j-th index of I in the increasing order is denoted by I_j. That is, if $I = \{i_1, \ldots, i_m\}$ and $i_j < i_{j+1}$ for all $1 \leq j < m$, then I_j represents i_j. The notation $\bar{I}$ denotes the complement of I, that is, $\bar{I} = \{1, \ldots, n\} \setminus I$. For a set $\mathcal{D}$ of defined symbols, the set of all pairs of a defined symbol and a set of indexes for the symbol is denoted by $\mathbb{I}_{\mathcal{D}}$: $\mathbb{I}_{\mathcal{D}} = \{\, (g, I) \mid g \in \mathcal{D}, I \subseteq \{1, \ldots, |g|\} \,\}$ where $|g|$ is the arity of g.

3.1 Definition of Partial-Inverses

Here we give a concrete definition of *partial-inverses*.

Definition 1. *Let $R^{\Leftarrow}$ be a CTRS over a signature $\mathcal{F}$, and $S^{\Leftarrow}$ be a CTRS over a signature $\mathcal{F}'$ satisfying $\mathcal{C}_{R^{\Leftarrow}} \subseteq \mathcal{C}_{S^{\Leftarrow}}$. Let f and g be defined symbols of $R^{\Leftarrow}$ and $S^{\Leftarrow}$, respectively, and I be a set of indexes for f. Then, g is a* partial inverse *of f with respect to I if the following holds:*

> *for all ground constructor terms $t, t_1, \ldots, t_n$ of $R^{\Leftarrow}$, $f(t_1, \ldots, t_n) \xrightarrow{*}_{R^{\Leftarrow}} t$ if and only if $g(t, t_{I_1}, \ldots, t_{I_{|I|}}) \xrightarrow{*}_{S^{\Leftarrow}} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1}, \ldots, t_{\bar{I}_{|\bar{I}|}})$,*

where $\mathsf{tp}_{|\bar{I}|} \in \mathcal{C}_{S^{\Leftarrow}}$. In particular, g is a full inverse *of f if $I = \emptyset$. The CTRS $S^{\Leftarrow}$ is called a* partial-inverse system *of $R^{\Leftarrow}$ with respect to a set $D_{\mathcal{I}} \subseteq \mathbb{I}_{\mathcal{D}_{R^{\Leftarrow}}}$ (simply called a partial-inverse system of $R^{\Leftarrow}$ when $D_{\mathcal{I}} = \mathbb{I}_{\mathcal{D}_{R^{\Leftarrow}}}$) if for every $(f, I) \in D_{\mathcal{I}}$, there exists $g \in \mathcal{D}_{S^{\Leftarrow}}$ such that g is a partial-inverse of f with respect to I. $R^{\Leftarrow}$ is sometimes called the* forward(-computation) system *for $S^{\Leftarrow}$.*

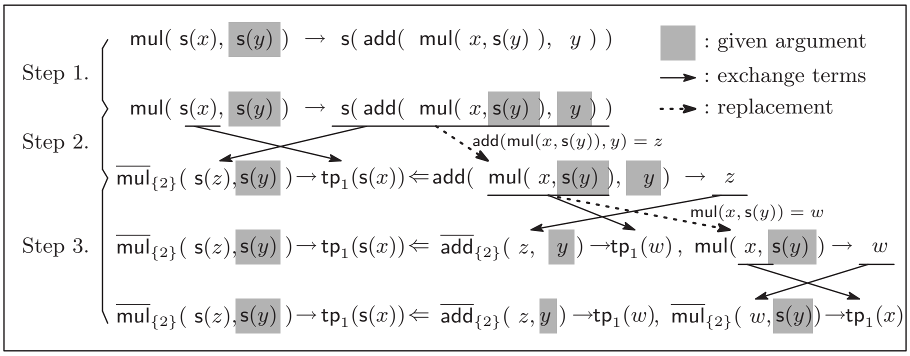

Fig. 2. Sketch of partial-inversion of the third rule of R_1.

Example 2. Consider the following convergent constructor TRSs over the signature $\{0, \mathsf{s}, \mathsf{add}, \mathsf{mul}, \mathsf{minus}\}$:

$$R_1 = \begin{cases} \mathsf{add}(0, y) \to y, \\ \mathsf{add}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{add}(x, y)), \\ \mathsf{mul}(0, y) \to 0, \quad \mathsf{mul}(x, 0) \to 0, \\ \mathsf{mul}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{s}(\mathsf{add}(\mathsf{mul}(x, \mathsf{s}(y)), y)) \end{cases} \qquad R_2 = \begin{cases} \mathsf{minus}(x, 0) \to x, \\ \mathsf{minus}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{minus}(x, y). \end{cases}$$

R_2 is a partial-inverse TRS of R_1 with respect to $\{(\mathsf{add}, \{1\})\}$.

3.2 Idea of Partial Inversion

This subsection intuitively explains how to generate a partial-inverse CTRS from a given constructor TRS, by using R_1 in Example 2 and the pair $(\mathsf{mul}, \{2\})$.

Roughly speaking, we generate a conditional rewrite rule for a rewrite rule in a given system R and a set of indexes for the root symbol of the lhs. The idea of the generation is based on the essential property of inverses: $f(v_1, \ldots, v_n) = v$ if and only if $f^{-1}(v, v_1, \ldots, v_i) = (v_{i+1}, \ldots, v_n)$. For a given pair (f, I) of a defined symbol f and a set I of indexes for f, we use $\overline{f}_I$ as a symbol of the partial-inverse of f with respect to I. We add a special rewrite rule for each (f, I) (as we will show the detail later). The partial-inverse CTRS of R_1 with respect to $\{(\mathsf{mul}, \{2\})\}$ is generated as follows:

$$R_3^{\Leftarrow} = \begin{cases} \overline{\mathsf{add}}_{\{2\}}(y, y) \to \mathsf{tp}_1(0), \\ \overline{\mathsf{add}}_{\{2\}}(\mathsf{s}(z), y) \to \mathsf{tp}_1(\mathsf{s}(x)) \Leftarrow \overline{\mathsf{add}}_{\{2\}}(z, y) \to \mathsf{tp}_1(x), \\ \overline{\mathsf{mul}}_{\{2\}}(0, y) \to \mathsf{tp}_1(0), \qquad \overline{\mathsf{mul}}_{\{2\}}(0, 0) \to \mathsf{tp}_1(x), \\ \overline{\mathsf{mul}}_{\{2\}}(\mathsf{s}(z), \mathsf{s}(y)) \to \mathsf{tp}_1(\mathsf{s}(x)) \\ \qquad \Leftarrow \overline{\mathsf{add}}_{\{2\}}(z, y) \to \mathsf{tp}_1(w), \overline{\mathsf{mul}}_{\{2\}}(w, \mathsf{s}(y)) \to \mathsf{tp}_1(x), \\ \overline{\mathsf{add}}_{\{2\}}(\mathsf{add}(x, y), y) \to \mathsf{tp}_1(x), \qquad \overline{\mathsf{mul}}_{\{2\}}(\mathsf{mul}(x, y), y) \to \mathsf{tp}_1(x). \end{cases}$$

We give an intuitive explanation by using the third rule of mul. To generate the fifth rule in R_3, we apply the following three steps to it (see Fig. 2):

(Step 1). *This step analyzes the rule and classifies variables in the rule into the given and the unknown, depending on whether value is assigned in future execution.* The result of this step is illustrated in the second line of Fig. 2. The pair $(\mathsf{mul}, \{2\})$ means that the second argument $\mathsf{s}(y)$ of mul in the lhs is given. Hence, the value of y is assigned in the execution. On the contrary, the value of x is unknown. Therefore, the second arguments of add and mul in the rhs are given, and their first arguments are unknown[1].

(Step 2). *This step creates a rule for inverses by exchanging parts of both the left- and right-hand sides of the rule, except the given arguments in the lhs.* When exchanging, we replace the term $\mathsf{add}(\mathsf{mul}(x, \mathsf{s}(y)), y)$ in the rhs with a fresh variable z (not used in the rule), and add the condition $\mathsf{add}(\mathsf{mul}(x, \mathsf{s}(y)), y) \to z$ caused by the replacement to the conditional part. This transformation makes the original rhs a constructor term. We also replace the root symbol mul in the lhs with the symbol $\overline{\mathsf{mul}}_{\{2\}}$.

(Step 3). *This step applies Step 2 to the condition part until it becomes deterministic.* By applying Step 2 twice to the condition part, we obtain the conditional rewrite rule of $\overline{\mathsf{mul}}_{\{2\}}$ illustrated in the bottom line of Fig. 2.

By applying the above three steps to all rules in R_1 and all needed pairs $((\mathsf{add}, \{2\})$ and $(\mathsf{mul}, \{2\}))$, we obtain the first five rules in $R_3^{\Leftarrow}$.

In addition to the above three steps, we generate for every pair $(f, I) \in \mathbb{I}_{\mathcal{D}}$ the special rewrite rule $\overline{f}_I(f(x_1, \ldots, x_n), x_{I_1}, \ldots, x_{I_{|\bar{I}|}}) \to \mathsf{tp}_{|\bar{I}|}(x_{\bar{I}_1}, \ldots, x_{\bar{I}_{|\bar{I}|}})$, called the *inverse-property rule* of f with respect to I. These rules are necessary for inverse computation of functions that invoke partial functions. Consider the TRS $R_4 = R_1 \cup \{\ \mathsf{half}(0) \to 0,\ \mathsf{half}(\mathsf{s}^2(x)) \to \mathsf{s}(\mathsf{half}(x)),\ \mathsf{g}(x) \to \mathsf{mul}(x, \mathsf{half}(\mathsf{s}(x)))\ \}$. Since we have a derivation $\mathsf{g}(0) \xrightarrow{*}_{R_4} 0$, the inverse computation $\overline{\mathsf{g}}_{\emptyset}(0) \xrightarrow{*} \mathsf{tp}_1(0)$ should hold. If $\overline{\mathsf{half}}_{\emptyset}(\mathsf{half}(x)) \to \mathsf{tp}_1(x)$ were missing, $\overline{\mathsf{g}}_{\emptyset}(0)$ could not be reachable to $\mathsf{tp}_1(0)$ because the only applicable rule to $\overline{\mathsf{g}}_{\emptyset}(0)$ is $\overline{\mathsf{g}}_{\emptyset}(y) \to \mathsf{tp}_1(x) \Leftarrow \overline{\mathsf{mul}}_{\emptyset}(y) \to \mathsf{tp}_2(x, z), \overline{\mathsf{half}}(z) \to \mathsf{tp}_1(\mathsf{s}(x))$ and we must find z satisfying $\overline{\mathsf{half}}_{\emptyset}(z) \to \mathsf{tp}_1(\mathsf{s}(0))$.

3.3 Generation of Partial-Inverse CTRSs

In this subsection, we formalize the idea described in Subsection 3.2. To simplify the presentation, we focus on generating partial-inverse CTRSs with respect to $\mathbb{I}_{\mathcal{D}}$. We can easily get the CTRS with respect to a subset of $\mathbb{I}_{\mathcal{D}}$ by collecting usable rules after the generation.

We first provide definitions (Definition 3–5) necessary to show the first stage of our compiler. We write $C[\![t_1, \ldots, t_n]\!]$ instead of a term $C[t_1, \ldots, t_n]$ if C is a constructor context (that is, $C[, \ldots,] \in \mathcal{T}(\mathcal{C} \cup \{\Box\}, \mathcal{V})$) and $\mathtt{root}(t_i)$ is a defined symbol for every $i \in \{1, \ldots, n\}$. It is clear that every term s can be represented in the form of $C'[\![s_1, \ldots, s_k]\!]$.

Definition 3. *Let $\rho : f(w_1, \ldots, w_n) \to C[\![r_1, \ldots, r_m]\!]$ be a rewrite rule over a signature $\mathcal{D} \uplus \mathcal{C}$ and I be a set of indexes for f. The set $\mathcal{UV}ar(\rho, I)$ of variables in ρ, whose values are* unknown*, is defined as follows:*

[1] The value of $\mathsf{mul}(x, \mathsf{s}(y))$ is not given because this term has the unknown variable x.

$$\mathcal{UV}ar\,(\,f(w_1,\ldots,w_n) \to C[\![r_1,\ldots,r_m]\!], I\,) \\ = \mathcal{V}ar(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}}) \setminus \mathcal{V}ar(w_{I_1},\ldots,w_{I_{|I|}}, C[,\ldots,]).$$

The above definition corresponds to the preparation of Step 1 in Subsection 3.2. An example of the above definition will be found after Definition 5.

For a term t and the set X of unknown variables, we define a label attachment to each defined symbol in t. The label of the defined symbol g for a subterm $g(t_1,\ldots,t_n)$ of t is a set I of indexes for g such that $i \in I$ if all variables in t_i are known (no variable in t is in X).

Definition 4. *Let $\mathcal{F}$ $(= \mathcal{C} \uplus \mathcal{D})$ be a signature, $f(t_1,\ldots,t_n)$ be a term with $f \in \mathcal{D}$, and X be a set of variables. We define the set of indexes which specifies positions of arguments t_is not containing any variable in X, as $\mathcal{I}\,(f(t_1,\ldots,t_n), X) = \{\, i \mid \mathcal{V}ar(t_i) \cap X = \emptyset \,\}$.*

For a term t, the labeled term $\mathcal{L}ab(t,X)$ in which defined symbols are labeled with a set of indexes, is recursively defined as follows:

- *$\mathcal{L}ab(x,X) = x$ where x is a variable,*
- *$\mathcal{L}ab(c(t_1,\ldots,t_n),X) = c\,(\mathcal{L}ab(t_1,X),\ldots,\mathcal{L}ab(t_n,X))$ where $c \in \mathcal{C}$, and*
- *$\mathcal{L}ab(f(t_1,\ldots,t_n),X) = f_I\,(\mathcal{L}ab(t_1,X),\ldots,\mathcal{L}ab(t_n,X))$ where $f \in \mathcal{D}$ and $I = \mathcal{I}(f(t_1,\ldots,t_n),X)$.*

The transformation from a rewrite rule $f(w_1,\ldots,w_n) \to r$ to the labeled rule $f_I(w_1,\ldots,,w_n) \to \mathcal{L}ab(r,\mathcal{UV}ar(\rho,I))$ corresponds to Step 1 in Subsection 3.2.

We here define the procedure that produces a term and a sequence of conditions, which are parts of constructing a conditional rule from a given rule.

Definition 5. *Let $\mathcal{F}$ $(= \mathcal{C} \uplus \mathcal{D})$ be a signature. The procedure $\mathscr{T}$, which outputs a pair of a term and a condition part from an input labeled term, is inductively defined as follows:*

(a) $\mathscr{T}(x) = \langle\, x \,;\, \epsilon \,\rangle$ *where x is a variable,*

(b) $\mathscr{T}(c(t_1,\ldots,t_n)) = \langle c(u_1,\ldots,u_n); \mathcal{C}ond_1,\ldots,\mathcal{C}ond_n\rangle$ *where $c \in \mathcal{C}$ and* $\mathscr{T}(t_i) = \langle\, u_i \,;\, \mathcal{C}ond_i \,\rangle$ *for $1 \le i \le n$,*

(c) $\mathscr{T}(f_I(t_1,\ldots,t_n)) =$
$\langle\, y \,;\, \overline{f}_I(y,u_{I_1},\ldots,u_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(s_{\bar{I}_1},\ldots,s_{\bar{I}_{|\bar{I}|}}), \mathcal{C}ond'_{\bar{I}_1},\ldots,\mathcal{C}ond'_{\bar{I}_{|\bar{I}|}} \,\rangle$
where $f \in \mathcal{D}$, $|I| < n$, y is a 'fresh'[2] variable, $\mathscr{T}(t_i) = \langle\, u_i \,;\, \mathcal{C}ond_i \,\rangle$ *for $1 \le i \le n$,*
  - $u_{\bar{I}_j} = C_{\bar{I}_j}[\![u_{\bar{I}_j,1},\ldots,u_{\bar{I}_j,m_{\bar{I}_j}}]\!]$, $s_{\bar{I}_j} = C_{\bar{I}_j}[z_{\bar{I}_j,1},\ldots,z_{\bar{I}_j,m_{\bar{I}_j}}]$,
  - $\mathcal{C}ond'_{\bar{I}_j} = \mathcal{C}ond_{\bar{I}_j}, u_{\bar{I}_j,1} \to z_{\bar{I}_j,1},\ldots,u_{\bar{I}_j,m_{\bar{I}_j}} \to z_{\bar{I}_j,m_{\bar{I}_j}}$, *and*
  - $z_{\bar{I}_j,k}$ *is a 'fresh'[2] variable, and,*

(d) $\mathscr{T}(f_{\{1,\ldots,n\}}(t_1,\ldots,t_n)) = \langle\, f(u_1,\ldots,u_n) \,;\, \epsilon \,\rangle$ *where $f \in \mathcal{D}$ and* $\mathscr{T}(t_i) = \langle\, u_i \,;\, \mathcal{C}ond_i \,\rangle$ *for $1 \le i \le n$,*

where we write ϵ to represent the empty sequence of conditions.

[2] This means that
- in the case (c), $y \notin \bigcup_{i=1}^{n} \mathcal{V}ar(t_i,u_i,\mathcal{C}ond_i)$, $z_{\bar{I}_j,k} \notin \{y\} \cup \bigcup_{i=1}^{n} \mathcal{V}ar(t_i,u_i,\mathcal{C}ond_i)$, $z_{\bar{I}_j,k'} \not\equiv z_{\bar{I}_j,k''}$ $(k' \neq k'')$, and
- in the both (b) and (c), variables introduced in each $\mathscr{T}(t_i) = \langle\, u_i \,;\, \mathcal{C}ond_i \,\rangle$ are disjoint, that is, $(\mathcal{V}ar(u_i,\mathcal{C}ond_i) \setminus \mathcal{V}ar(t_i)) \cap \mathcal{V}ar(t_j,u_j,\mathcal{C}ond_j) = \emptyset$ for $i \neq j$.

It is clear that the above procedure $\mathscr{T}$ always terminates and returns a pair of a term and a conditional part. Note that for $\mathscr{T}(t) = \langle\, u \,;\, \mathcal{C}ond \,\rangle$, u can be represented as $C[\![u_1,\ldots,u_n]\!]$ $(\in \mathcal{T}(\mathcal{F},\mathcal{V}))$.

Example 6. $\mathcal{UV}ar$, $\mathcal{I}$, $\mathcal{L}ab$ and $\mathscr{T}$ perform for the fifth rule of R_1 as follows:

$$\begin{array}{c}
\mathcal{UV}ar(\ \mathsf{mul}(\mathsf{s}(x),\mathsf{s}(y)) \to \mathsf{s}(\mathsf{add}(\mathsf{mul}(x,\mathsf{s}(y)),y)),\ \{2\}\ ) = \{x\},\\
\mathcal{I}(\ \mathsf{add}(\mathsf{mul}(x,\mathsf{s}(y)),y),\ \{x\}\ ) = \{2\}, \quad \mathcal{I}(\ \mathsf{mul}(x,\mathsf{s}(y)),\ \{x\}\ ) = \{2\},\\
\mathcal{L}ab(\ \mathsf{s}(\mathsf{add}(\mathsf{mul}(x,\mathsf{s}(y)),y)),\ \{x\}\ ) = \mathsf{s}(\mathsf{add}_{\{2\}}(\mathsf{mul}_{\{2\}}(x,\mathsf{s}(y)),y)),\\
\mathscr{T}(\ \mathsf{s}(\mathsf{add}_{\{2\}}(\mathsf{mul}_{\{2\}}(x,\mathsf{s}(y)),y))\ )\\
\qquad = \langle\ \mathsf{s}(z)\ ;\ \overline{\mathsf{add}}_{\{2\}}(z,y) \to w, \overline{\mathsf{mul}}_{\{2\}}(w,\mathsf{s}(y)) \to x\ \rangle.
\end{array}$$

We finally define the partial-inversion from constructor TRSs into CTRSs.

Definition 7. *Let R be a constructor TRS over a signature $\mathcal{F}$. For a rewrite rule $\rho : f(w_1,\ldots,w_n) \to r \in R$ and a set I of indexes for f, its corresponding conditional rewrite rule $\mathcal{I}nv\mathcal{R}ule(\rho, I)$ of $\overline{f}_I$ is defined as follows:*

$$\begin{array}{l}
\mathcal{I}nv\mathcal{R}ule(\ f(w_1,\ldots,w_n) \to r,\ I) =\\
\qquad \overline{f}_I(C[y_1,\ldots,y_m], w_{I_1},\ldots,w_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}})\\
\qquad\qquad\qquad\qquad \Leftarrow u_1 \to y_1,\ldots,u_m \to y_m, \mathcal{C}ond
\end{array}$$

where $\mathscr{T}(\ \mathcal{L}ab(r,\mathcal{UV}ar(\rho,I))\ ) = \langle\ C[\![u_1,\ldots,u_m]\!]\ ;\ \mathcal{C}ond\ \rangle$, each y_i is a variable with $y_i \notin \mathcal{V}ar(w_1,\ldots,w_n)$, and $(\mathcal{V}ar(C[\![u_1,\ldots,u_m]\!],\mathcal{C}ond) \setminus \mathcal{V}ar(r)) \cap \mathcal{V}ar(y_1,\ldots,y_m,\ w_1,\ldots,w_n) = \emptyset$.

The partial-inverse CTRS $\mathcal{I}nv(R)$ is defined as follows:

$$\begin{array}{l}
\mathcal{I}nv(R) = \{\ \mathcal{I}nv\mathcal{R}ule(\rho,I) \mid \rho : f(w_1,\ldots,w_n) \to r \in R, I \subseteq \{1,\ldots,n\}\ \}\\
\quad \cup \{\ \overline{f}_I(f(x_1,\ldots,x_n), x_{I_1},\ldots,x_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(x_{\bar{I}_1},\ldots,x_{\bar{I}_{|\bar{I}|}}) \mid (f,I) \in \mathbb{I}_{\mathcal{D}_R}\ \} \cup R.
\end{array}$$

The extended signature $\overline{\mathcal{F}}$ is defined as $\overline{\mathcal{F}} = \mathcal{F} \uplus \{\ \overline{f}_I \mid (f,I) \in \mathbb{I}_{\mathcal{D}_R}\ \}$.

The generation of conditional rules by $\mathcal{I}nv\mathcal{R}ule$ corresponds to Step 2 and Step 3 in Subsection 3.2. Note that our approach does not require the left-linearity of input systems. It is clear that $\mathcal{I}nv\mathcal{R}ule(\rho, I)$ is exactly a deterministic conditional rewrite rule and then $\mathcal{I}nv(R)$ is exactly a deterministic CTRS over the signature $\overline{\mathcal{F}}$. For a CTRS $S^{\Leftarrow}$ and a set D of defined symbols of $S^{\Leftarrow}$, we use $(S^{\Leftarrow})|_D$ for all rules in $S^{\Leftarrow}$ that are necessary to calculate terms in $\mathcal{T}(D \cup \mathcal{C}_{S^{\Leftarrow}}, \mathcal{V})$.

Example 8. Consider R_1 again. For $(\mathsf{mul},\{2\})$ and $(\mathsf{mul},\{1,2\})$, we obtain by $\mathcal{I}nv$ the following CTRS:

$$\begin{array}{l}
R_5^{\Leftarrow} = \mathcal{I}nv(R_1)|_{\{\overline{\mathsf{mul}}_{\{2\}},\overline{\mathsf{mul}}_{\{1,2\}}\}}\\
\quad = R_3^{\Leftarrow} \cup \left\{ \begin{array}{l}
\overline{\mathsf{mul}}_{\{1,2\}}(0,0,y) \to \mathsf{tp}_0, \quad \overline{\mathsf{mul}}_{\{1,2\}}(0,x,0) \to \mathsf{tp}_0,\\
\overline{\mathsf{mul}}_{\{1,2\}}(\mathsf{s}(z),\mathsf{s}(x),\mathsf{s}(y)) \to \mathsf{tp}_0 \Leftarrow \mathsf{add}(\mathsf{mul}(x,\mathsf{s}(y)),y) \to z,\\
\overline{\mathsf{mul}}_{\{1,2\}}(\mathsf{mul}(x,y),x,y) \to \mathsf{tp}_0
\end{array} \right\} \cup R_1.
\end{array}$$

The following theorem shows that the CTRS obtained by $\mathcal{I}nv$ from a given constructor TRS can perform the inverse computation of innermost derivations of the TRS. For a TRS R, we write $\mathrel{\underset{\mathrm{in}}{\to}}_R$ as the *innermost reduction* of R.

Theorem 9. *Let R be a constructor TRS over a signature $\mathcal{F}$. Let $(f, I) \in \mathbb{I}_{\mathcal{D}_R}$ and $t, t_1, \ldots, t_n$ be normal forms of R. Then, $f(t_1, \ldots, t_n) \xrightarrow[\text{in}]{*}_R t$ if and only if $\overline{f}_I(t, t_{I_1}, \ldots, t_{I_{|I|}}) \to_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1}, \ldots, t_{\bar{I}_{|\bar{I}|}})$ with innermost evaluation of conditional parts and normalized substitution for extra variables.*

It is clear that every normalizing reduction sequence of convergent TRSs can be simulated by an innermost-reduction sequence. Hence, the above theorem implies the following corollary.

Corollary 10. *Let R be a convergent constructor TRS over a signature $\mathcal{F}$. Let $(f, I) \in \mathbb{I}_{\mathcal{D}_R}$ and $t, t_1, \ldots, t_n$ be normal forms of R. Then, $f(t_1, \ldots, t_n) \xrightarrow{*}_R t$ if and only if $\overline{f}_I(t, t_{I_1}, \ldots, t_{I_{|I|}}) \to_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1}, \ldots, t_{\bar{I}_{|\bar{I}|}})$.*

From the above corollary and the fact $\mathcal{T}(\mathcal{C}_R) \subseteq NF_R(\mathcal{F}, \mathcal{V})$, the CTRS $\mathcal{I}nv(R)$ is exactly a partial-inverse system of the convergent constructor TRS R with respect to $\mathbb{I}_{\mathcal{D}_R}$ in the sense of Definition 1. In addition, it is clear that $(\mathcal{I}nv(R))|_{D_\mathcal{I}}$ for some subset $D_\mathcal{I} \subseteq \mathbb{I}_{\mathcal{D}_R}$ is a partial-inverse CTRS of R with respect to $D_\mathcal{I}$.

3.4 Unraveling to Unconditional Systems

As the second stage of the compiler proposed in this paper, we here give the unraveling for deterministic CTRSs [17, 19]. For a set $A = \{a_1, \ldots, a_n\}$, $\vec{A}$ denotes a list $a_1, \ldots, a_n$ for a unique representation of A.

Definition 11 ([17]). *Let $R^{\Leftarrow}$ be a deterministic CTRS over a signature $\mathcal{F}$. For every conditional rewrite rule $\rho : l \to r \Leftarrow s_1 \to t_1, \ldots, s_k \to t_k$ in $R^{\Leftarrow}$, we prepare k fresh function symbols $\mathsf{u}^\rho_1, \ldots, \mathsf{u}^\rho_k$, called* $\mathbb{U}$ symbols*, neither of which appears in $\mathcal{F}$. Then, the set of rewrite rules determined by ρ is defined as follows:*

$$\mathbb{U}(\rho) = \{\ l \to \mathsf{u}^\rho_1(s_1, \vec{X_1}),\ \mathsf{u}^\rho_1(t_1, \vec{X_1}) \to \mathsf{u}^\rho_2(s_2, \vec{X_2}),\ \cdots,\ \mathsf{u}^\rho_k(t_k, \vec{X_k}) \to r\ \}$$

where $X_i = \mathcal{V}ar(l, t_1, \ldots, t_{i-1}) \cap \mathcal{V}ar(r, t_i, s_{i+1}, t_{i+1}, \ldots, s_k, t_k)$. The system $\mathbb{U}(R^{\Leftarrow}) = \bigcup_{\rho \in R^{\Leftarrow}} \mathbb{U}(\rho)$ is an EV-TRS over the extended signature $\mathcal{F}_{\mathbb{U}(R^{\Leftarrow})} = \mathcal{F} \uplus \bigcup_{\rho \in R^{\Leftarrow}} \{\ \mathsf{u}^\rho_i \mid 1 \le i \le |\rho|\ \}$ where $|\rho|$ denotes the number of the conditions of ρ.

It is clear that $R^{\Leftarrow}$ is a 3-CTRS if and only if $\mathbb{U}(R^{\Leftarrow})$ is a TRS.

Example 12. The CTRS $R_3^{\Leftarrow}$ in Subsection 3.2 is unraveled by $\mathbb{U}$ as follows:

$$\mathbb{U}(R_3^{\Leftarrow}) = \begin{cases} \overline{\mathsf{add}}_{\{2\}}(y, y) \to \mathsf{tp}_1(0), & \\ \overline{\mathsf{add}}_{\{2\}}(\mathsf{s}(z), y) \to \mathsf{u}_1(\overline{\mathsf{add}}_{\{2\}}(z, y)), & \mathsf{u}_1(\mathsf{tp}_1(x)) \to \mathsf{tp}_1(\mathsf{s}(x)), \\ \overline{\mathsf{mul}}_{\{2\}}(0, y) \to \mathsf{tp}_1(0), & \overline{\mathsf{mul}}_{\{2\}}(0, 0) \to \mathsf{tp}_1(x), \\ \overline{\mathsf{mul}}_{\{2\}}(\mathsf{s}(z), \mathsf{s}(y)) \to \mathsf{u}_2(\overline{\mathsf{add}}_{\{2\}}(z, y), y), & \\ \mathsf{u}_2(\mathsf{tp}_1(w), y) \to \mathsf{u}_3(\overline{\mathsf{mul}}_{\{2\}}(w, \mathsf{s}(y))), & \mathsf{u}_3(\mathsf{tp}_1(x)) \to \mathsf{tp}_1(\mathsf{s}(x)), \\ \overline{\mathsf{add}}_{\{2\}}(\mathsf{add}(x, y), y) \to \mathsf{tp}_1(x), & \overline{\mathsf{mul}}_{\{2\}}(\mathsf{mul}(x, y), y) \to \mathsf{tp}_1(x). \end{cases}$$

The unraveled CTRS cannot always simulate any rewrite sequence of an original CTRS completely. It holds for every deterministic CTRS $R^{\Leftarrow}$ over a signature $\mathcal{F}$ that for every terms s and t in $\mathcal{T}(\mathcal{F},\mathcal{V})$, $s \xrightarrow{*}_{R^{\Leftarrow}} t$ implies $s \xrightarrow{*}_{\mathbb{U}(R^{\Leftarrow})} t$, (called *simulation-preserving*). However, the converse does not hold in general (see Example 7.2.14 in [19]). The converse is called the *simulation-soundness* of $\mathbb{U}$ for $R^{\Leftarrow}$. That is, for every terms s and t in $\mathcal{T}(\mathcal{F},\mathcal{V})$, $s \xrightarrow{*}_{\mathbb{U}(R^{\Leftarrow})} t$ implies $s \xrightarrow{*}_{R^{\Leftarrow}} t$. Simulation-preserving and -soundness is called *simulation-completeness* of $\mathbb{U}$ for $R^{\Leftarrow}$, that is, for every terms s and t in $\mathcal{T}(\mathcal{F},\mathcal{V})$, $s \xrightarrow{*}_{\mathbb{U}(R^{\Leftarrow})} t$ if and only if $s \xrightarrow{*}_{R^{\Leftarrow}} t$. In Section 4, we will show two solutions of this problem.

The unraveled CTRSs may produce some garbage normal forms, which contain $\mathbb{U}$ symbols. When $\mathbb{U}$ is simulation-complete, we can easily recognize whether a obtained normal form is garbage or not.

4 Computation of Partial-Inverse EV-TRSs

The rewrite relation of the generated systems has two problems to be solved. One is the existence of extra variables, and the other is the simulation-*incompleteness* of the unraveling $\mathbb{U}$. In this section, we state briefly that the first problem does not matter, and then we deal with the second problem from two approaches: a restriction against the rewrite relation, and a syntactic constraint on CTRSs.

4.1 On Extra Variables

Extra variables cause infinitely-branching and non-termination of the rewrite relation. However, these troubles are solved by *narrowing* [10] *starting from ground terms* — every *EV-normalized* rewrite sequence with the ground initial term has a narrowing sequence which starts from the same initial term, and of which the last term can be matched with the last one of the rewrite sequence, and vice versa [16]. Here a rewrite sequence is said to be *EV-normalized*, denoted by $\xrightarrow[\mathrm{ev}]{}$, if a normal form is substituted for each extra variable in every reduction step of the rewrite sequence. In the subproof of Theorem 9 (see Appendix A), the inverse computation $\overline{f}_I(t,t_{I_1},\ldots,t_{I_{|I|}}) \rightarrow_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1},\ldots,t_{\bar{I}_{|\bar{I}|}})$ is constructed from the innermost derivation $f(t_1,\ldots,t_n) \xrightarrow[\mathrm{in}]{*}_R t$, substituting normal forms for extra variables. In addition, such a reduction can be easily simulated by the unraveled CTRS. These facts mean that every partial-inverse computation for innermost normalizing derivation can be represented by a (ground) EV-normalized rewrite sequence of the generated EV-TRSs.

Proposition 13. *Let R be a constructor TRS over a signature $\mathcal{F}$. Let $(f,I) \in \mathbb{I}_{\mathcal{D}_R}$ and $t,t_1,\ldots,t_n$ be normal forms of R. If $f(t_1,\ldots,t_n) \xrightarrow[\mathrm{in}]{*}_R t$, then there exists an EV-normalized rewrite sequence such that $\overline{f}_I(t,t_{I_1},\ldots,t_{I_{|I|}}) \xrightarrow[\mathrm{ev}]{*}_{\mathbb{U}(\mathcal{I}nv(R))} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1},\ldots,t_{\bar{I}_{|\bar{I}|}})$.*

Thus, every necessary rewrite sequence for partial-inverse computation can be simulated by a narrowing sequence starting from a ground term. Every step of narrowing is finitely branching up to renaming, and the non-termination is less serious in the narrowing starting from ground terms than in the ordinary narrowing. In fact, some EV-TRSs terminate with respect to narrowing starting from ground terms, while they do not terminates with respect to the ordinary narrowing.

4.2 On Simulation-Incompleteness of Unraveling

The second problem stated in the beginning of this section is caused from the disordered evaluation of rules that originate in conditional parts (as shown in Example 7.2.14 of [19]). It is solved by combining the *membership conditional* and *context-sensitive* reductions.

We first give definitions of three reductions for the unraveled CTRSs: the *membership conditional reduction* [24], the *context-sensitive reduction* [12] and their combined reduction.

Definition 14. *Let $R^{\Leftarrow}$ be a deterministic CTRS over a signature $\mathcal{F}$. The* membership conditional rewrite relation $\xrightarrow[\mathrm{M}]{}_{\mathbb{U}(R^{\Leftarrow})}$ *on $\mathcal{T}(\mathcal{F}_{\mathbb{U}(R^{\Leftarrow})},\mathcal{V})$ is defined as* $\xrightarrow[\mathrm{M}]{}_{\mathbb{U}(R^{\Leftarrow})} = \{\, (s,t) \mid s \to^{p}_{\mathbb{U}(R^{\Leftarrow})} t, (\,\forall u, u \lhd s|_p \textit{ implies } u \in \mathcal{T}(\mathcal{F},\mathcal{V})\,)\,\}$.

Let μ be a mapping from $\mathcal{F}_{\mathbb{U}(R^{\Leftarrow})}$ to a set of argument positions for f such that $\mu(\mathsf{u}) = \{1\}$ for all $\mathsf{u} \in \mathcal{F}_{\mathbb{U}(R^{\Leftarrow})} \setminus \mathcal{F}$, and $\mu(f) = \{1,\dots,n\}$ for all n-ary function symbol $f \in \mathcal{F}$. The set $\mathcal{O}_\mu(t)$ for a term $t \in \mathcal{T}(\mathcal{F}_{\mathbb{U}(R^{\Leftarrow})},\mathcal{V})$ is defined recursively as follows:

- $\mathcal{O}_\mu(x) = \emptyset$ *where x is a variable, and*
- $\mathcal{O}_\mu(f(t_1,\dots,t_n)) = \{\, i_j q \mid 1 \le j \le m, q \in \mathcal{O}_\mu(t_{i_j})\,\}$ *where $f \in \mathcal{F}_{\mathbb{U}(R^{\Leftarrow})}$ and $\mu(f) = \{i_1,\dots,i_m\}$.*

The context-sensitive rewrite relation $\xrightarrow[\mathrm{CS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$ *on $\mathcal{T}(\mathcal{F}_{\mathbb{U}(R^{\Leftarrow})},\mathcal{V})$ is defined as* $\xrightarrow[\mathrm{CS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)} = \{\, (s,t) \mid s \to^{p}_{\mathbb{U}(R^{\Leftarrow})} t, p \in \mathcal{O}_\mu(s)\,\}$.

The membership context-sensitive (MCS) rewrite relation $\xrightarrow[\mathrm{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$ *is defined as* $\xrightarrow[\mathrm{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)} = (\xrightarrow[\mathrm{M}]{}_{\mathbb{U}(R^{\Leftarrow})}) \cap (\xrightarrow[\mathrm{CS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)})$.

Remark that these three reductions on narrowing are defined similarly. It is clear that $\xrightarrow[\mathrm{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)} \subseteq \xrightarrow[\mathrm{M}]{}_{\mathbb{U}(R^{\Leftarrow})} \subseteq \to_{\mathbb{U}(R^{\Leftarrow})}$, and $\xrightarrow[\mathrm{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)} \subseteq \xrightarrow[\mathrm{CS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)} \subseteq \to_{\mathbb{U}(R^{\Leftarrow})}$. From these facts, both the termination of $\to_{\mathbb{U}(R^{\Leftarrow})}$ and the μ-termination of $\xrightarrow[\mathrm{CS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$ guarantee the termination of $\xrightarrow[\mathrm{M}]{}_{\mathbb{U}(R^{\Leftarrow})}$, $\xrightarrow[\mathrm{CS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$ and $\xrightarrow[\mathrm{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$. To prove the termination of them, we can use existing tools, such as AProVE [5], TTT [8], MU-TERM [13]. The MCS reduction above is implemented as the particular case of MEP [4], and then we can use the technique in [4] to prove the termination of $\xrightarrow[\mathrm{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$.

We here discuss the simulation-completeness with respect to the MCS reduction. Consider the rule $\rho : \mathsf{f}(x,y) \to x \Leftarrow \mathsf{g}(x) \to z, \mathsf{g}(y) \to z$ and the corresponding unraveled rules $\mathbb{U}(\rho) = \{\, \mathsf{f}(x,y) \to \mathsf{u}^{\rho}_1(\mathsf{g}(x),x,y),\ \mathsf{u}^{\rho}_1(z,x,y) \to \mathsf{u}^{\rho}_2(\mathsf{g}(y),x,z),\ \mathsf{u}^{\rho}_2(z,x,z) \to x\,\}$. The $\mathbb{U}$ symbol u^{ρ}_1 is used to evaluate in the first argument the first condition $\mathsf{g}(x) \to z$, and to deliver the value of variables x and y via the rest of its arguments (that is, the second and third). From this observation, no redex in either of k-th argument ($2 \le k$) of u should be reduced until u is reduced, and u should not be evaluated until the evaluation of the first argument is finished. These evaluations are kept by the context-sensitive and the membership reductions, respectively. Hence, conditional parts of rules in $R^{\Leftarrow}$ are evaluated in proper order on $\xrightarrow[\text{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$.

Theorem 15. *For every deterministic CTRS $R^{\Leftarrow}$, $\mathbb{U}$ is simulation-complete with respect to $\xrightarrow[\text{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$.*

Proof (Sketch). $\mathbb{U}$ has simulation-preserving and simulation-soundness can be proved straightforward by induction on the length of the rewrite sequences of $\xrightarrow[\text{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$. □

As another solution of the second problem stated in the beginning of this section, we have shown some results on the simulation-completeness of $\mathbb{U}$ with respect to the ordinary rewrite relation [17].

Theorem 16 ([17]). *Let $R^{\Leftarrow}$ be a deterministic CTRS.*

- *If either $\mathbb{U}(R^{\Leftarrow})$ is a left-linear TRS or $\mathbb{U}(R^{\Leftarrow})$ is right-linear and non-erasing, then $\mathbb{U}$ is simulation-complete for $R^{\Leftarrow}$ with respect to $\to_{\mathbb{U}(R^{\Leftarrow})}$.*
- *If $\mathbb{U}(R^{\Leftarrow})$ is left-linear, then $\mathbb{U}$ is simulation-complete for $R^{\Leftarrow}$ with respect to $\xrightarrow[\text{ev}]{}_{\mathbb{U}(R^{\Leftarrow})}$.*

Note that syntactic conditions of CTRSs that the unraveled CTRSs are left-linear, right-linear and non-erasing, respectively, are shown in [17]. Theorem 16 may not be applicable to other unravelings, while $\xrightarrow[\text{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$ may provide the simulation-completeness to them. According to Proposition 13, the restriction of the rewrite sequences to $\xrightarrow[\text{ev}]{}_{\mathbb{U}(R^{\Leftarrow})}$ does not affect the computation of the generated EV-TRSs.

From Theorem 9 and the discussion in this section, we can conclude that our method is correct.

5 Improving Efficiency of Partial-Inverse Computation

In this section, we show that the efficiency of partial-inverse computation can be improved by the innermost strategy without loss of completeness if the generated systems are right-linear.

It has been shown in [22] that for every right-linear overlay TRS, all normal forms of terminating terms can be obtained by innermost strategy. As shown in the following theorem, this result has been extended.

Theorem 17 ([20]). *Let R be an ILRJ and right-linear TRS, and s be a terminating term. For all normal forms t of s, $s \xrightarrow[\text{in}]{*}_R t$.*

Here the *inside critical pairs* of a CTRS $R^{\Leftarrow}$ are the critical pairs obtained from rules overlap at non-root positions. The CTRS $R^{\Leftarrow}$ is said to be *inside left-to-right joinable* (ILRJ, for short) if every inside critical pair $\langle s,t\rangle$ ($s \leftarrow^{\varepsilon<} \cdot \to^{\varepsilon} t$) satisfies $s \xrightarrow{*}_{R^{\Leftarrow}} t$. Note that overlay systems are ILRJ.

Our partial-inverse EV-TRSs are sometimes not overlay. For example, $R_5^{\Leftarrow}$ in Example 8 is not overlay because they have the defined symbols add and mul in the first arguments of the lhs's of the inverse-property rules. From the same reason, the combination of forward and inverse programs such as $R_1 \cup \mathbb{U}(R_3^{\Leftarrow})$ are not always overlay. However, $R_3^{\Leftarrow}$ in Subsection 3.2 and $\mathbb{U}(R_3^{\Leftarrow})$ in Example 12 are ILRJ. In fact, the generated partial-inverse systems are always ILRJ.

Theorem 18. *Let R be a constructor TRS over a signature $\mathcal{F}$. Assume that for every rule $l \to r \in R$, the rhs r is weakly normalizing for the innermost reduction. Then, $\mathcal{I}nv(R) \cup R$ and $\mathbb{U}(\mathcal{I}nv(R)) \cup R$ are ILRJ.*

Proof. By the construction of $\mathcal{I}nv(R)$, inside overlaps in $\mathcal{I}nv(R) \cup R$ happens between rules $f(w_1,\ldots,w_n) \to r \in R$ and $\overline{f}_I(f(x_1,\ldots,x_n), x_{I_1},\ldots,x_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(x_{\bar{I}_1},\ldots,x_{\bar{I}_{|\bar{I}|}}) \in \mathcal{I}nv(R)$. Hence, we have inside critical pairs only in the form $\langle \overline{f}_I(r, w_{I_1},\ldots,w_{I_{|I|}}), \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}})\rangle$. By the assumption, there exists a normal form t of R such that $r \xrightarrow[\text{in}]{*}_R t$. Then, we have $f(w_1,\ldots,w_n) \xrightarrow[\text{in}]{}_R r \xrightarrow[\text{in}]{*}_R t$. It follows Theorem 9 that $\overline{f}_I(t, w_{I_1},\ldots,w_{I_{|I|}}) \to_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}})$, and hence $\overline{f}_I(r, w_{I_1},\ldots,w_{I_{|I|}}) \xrightarrow{*}_{\mathcal{I}nv(R)\cup R} \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}})$. Therefore, $\mathcal{I}nv(R) \cup R$ is ILRJ. The case of $\mathbb{U}(\mathcal{I}nv(R)) \cup R$ is similar to the above case. □

In this case , it is clear that $(\mathcal{I}nv(R))|_{D_\mathcal{I}} \cup R$ and $\mathbb{U}((\mathcal{I}nv(R))|_{D_\mathcal{I}}) \cup R$ for $D_\mathcal{I} \subseteq \mathbb{I}_{\mathcal{D}_R}$ are also ILRJ. Now we suppose that forward computation is convergent, and then the assumption holds. Hence, the assumption is not really a restriction for the generated systems. It also holds that the CTRS $S^{\Leftarrow} \cup (R \cup (\mathcal{I}nv(R))|_{D_\mathcal{I}})$ (or $S^{\Leftarrow} \cup (R \cup \mathbb{U}((\mathcal{I}nv(R))|_{D_\mathcal{I}}))$ is ILRJ if $S^{\Leftarrow}$ is ILRJ, because the assumption that they are *constructor-sharing systems* is adequate.

From Theorem 17 and 18, innermost strategy is effective to improve the efficiency of reductions by the right-linear partial-inverse systems, without loss of the reachability to solutions.

The MCS reduction $\xrightarrow[\text{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$ does not eliminate any necessary reduction sequence starting from a given term. Innermost strategy does not also eliminate such a sequence, when all of the conditions in Theorem 17 are satisfied. Therefore, in such cases, the MCS reduction with innermost strategy (that is, $\xrightarrow[\text{MCS,in}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$) is not less efficient than either $\xrightarrow[\text{in}]{}_{\mathbb{U}(R^{\Leftarrow})}$ or $\xrightarrow[\text{MCS}]{}_{\mathbb{U}(R^{\Leftarrow})(\mu)}$.

6 Related Works and Conclusion

Full-inversion compilers have been studied in [6, 7, 11, 21] which are applicable to several functional languages, and of which the correctness was not discussed (not proved). By contrast, we have shown the correctness of our method. Moreover, the discussion in Section 5 seems the first work on improving the efficiency of inverse computation. A partial-inversion compiler is considered in [21], which is for the programming language *Refal* (as like constructor normal CTRSs), and in which the non-determinism of inverses is solved by representing output of functions as a set. *Bidirectional transformation* [9] based on the *bidirectional updating* in the field of database, uses *bidirectional languages* which is a similar notion to the partial inversion.

There is another approach to inverse computation. *Inverse interpreters* are procedures that compute unknown inputs from the program and a given output. Several kinds of interpreters have been studied in [1, 3, 23]. Inverse interpreters seem to deal easily with partial-inverse problem. The algorithm in [3] (which is consists of inference rules) essentially resembles our method in the sense that the inputs are left-linear (ground-)convergent constructor TRSs. The algorithm terminates if the input TRS is *constructing*. We believe that the EV-TRS generated by our compiler from a constructing TRS is terminating with respect to narrowing starting from ground terms. Moreover, we have the example TRS R_4 which is not constructing but whose inverse is terminating.

The inverse computation in this paper can handle general solutions by variables that represent arbitrary terms. The compiler proposed in this paper is of course applicable to functions which returns multiple values, because the compiler can handle them by rules in the form of $f(\cdots) \rightarrow \mathit{tuple}(v_1, \ldots, v_m)$. The resulted TRS of the unraveling may be optimized. For example, u_1 and u_3 in $\mathbb{U}(R_3^{\Leftarrow})$ are redundant.

We have encountered some examples that the efficiency of $\xrightarrow[\mathrm{MCS,in}]{}_{\mathbb{U}(\mathcal{I}nv(R))(\mu)}$ is equal to that of $\xrightarrow[\mathrm{in}]{}_{\mathbb{U}(\mathcal{I}nv(R))}$. It is a future work to analyze the detail of the efficiency. We are also interested in relationships between syntactic properties of an input TRS and the generated EV-TRS, for example, a condition of R inducing the right-linearity of $\mathbb{U}(\mathcal{I}nv(R))$.

Which is more valuable, full inverses or partial inverses? Full inverses are included in partial inverses, and partial inverses can compute by the corresponding full-inverses because the results of the full inverses contain all solutions of partial inverses. Full inverses seems to be less efficient than partial inverses, but we know some desirable properties of syntactic relationships between forward and full-inverse programs. Analysis for this problem is another future work.

References

1. Abramov, S., Glück, R.: The universal resolving algorithm: Inverse computation in a functional language. In Proceedings of the 5th International Conference on Mathematics of Program Construction. Volume 1837 of LNCS, Springer (2000) 187–212.
2. Baader, F., Nipkow, T.: *Term Rewriting and All That*. Cambridge University Press (1998).
3. Dershowitz, N., Mitra, S.: Jeopardy. In *Proceedings of the 10th International Conference on Rewriting Techniques and Applications*. Volume 1631 of LNCS, Springer (1999) 16–29.
4. Durán, F., Lucas, S., Meseguer, J., Marché, C., Urbain, X.: Proving termination of membership equational programs. In *Proceedings of the ACM SIGPLAN 2004 Symposium on Partial Evaluation and Program Manipulation*. ACM Press (2004) 147–158.
5. Giesl, J., Thiemann, R., Schneider-Kamp, P., Falke, S.: Automated termination proofs with AProVE. In *Proceedings of the 15th International Conference on Rewriting Techniques and Applications*. Volume 3091 of LNCS, Springer (2004) 210–220.
6. Glück, R., Kawabe, M.: A program inverter for a functional language with equality and constructors. In *Proceedings of the first Asian Symposium on Programming Languages and Systems*. Volume 2895 of LNCS, Springer (2003) 246–264.
7. Harrison, P.G.: Function inversion. In *Proceedings of the IFIP TC2 Workshop on Partial Evaluation and Mixed Computation*. North-Holland (1988) 153–166.
8. Hirokawa, N., Middeldorp, A.: Tsukuba termination tool. In *Proceedings of the 14th International Conference on Rewriting Techniques and Applications*. Volume 2706 of LNCS, Springer (2003) 311–320.

9. Hu, Z., Mu, S.-C., Takeichi, M.: A programmable editor for developing structured documents based on bidirectional transformations. In *Proceedings of the 2004 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-based Program Manipulation*. ACM Press (2004) 178–189.
10. Hullot, J.-M.: Canonical forms and unification. In *Proceedings of the 5th International Conference on Automated Deduction*. Volume 87 of LNCS, Springer (1980) 318–334.
11. Khoshnevisan, H., Sephton, K.M.: InvX: An automatic function inverter. In *Proceedings of the 3rd International Conference on Rewriting Techniques and Applications*. Volume 355 of LNCS, Springer (1989) 564–568.
12. Lucas, S.: Context-sensitive computations in functional and functional logic programs. Journal of Functional and Logic Programming **1998** (1). The MIT Press (1998) 1–61.
13. Lucas, S.: MU-TERM: A tool for proving termination of context-sensitive rewriting. In *Proceedings of the 15th International Conference on Rewriting Techniques and Applications*. Volume 3091 of LNCS, Springer (2004) 200–209.
14. Marchiori, M.: Unravelings and ultra-properties. In *Proceedings of the 5th International Conference on Algebraic and Logic Programming*. Volume 1139 of LNCS, Springer (1996) 107–121.
15. Nishida, N., Sakai, M., Sakabe, T.: Generation of inverse term rewriting systems for pure treeless functions. In *Proceedings of the International Workshop on Rewriting in Proof and Computation*. Sendai, Japan (2001) 188–198.
16. Nishida, N., Sakai, M., Sakabe, T.: Narrowing-based simulation of term rewriting systems with extra variables and its termination proof. In *Functional and Constraint Logic Programming*. Volume 86 of ENTCS, Issue 3, Elsevier (2003) 18 pages.
17. Nishida, N., Sakai, M., Sakabe, T.: On simulation-completeness of unraveling for conditional term rewriting systems. Technical Report SS 2004-18 of IEICE (2004) 25–30.
18. Ohlebusch, E.: Termination of logic programs: Transformational methods revisited. Applicable Algebra in Engineering, Communication and Computing **12** (2001) 73–116.
19. Ohlebusch, E.: *Advanced Topics in Term Rewriting*. Springer (2002).
20. Okamoto, K., Sakai, M., Sakabe, T.: Completeness of innermost strategy for inside-LR-joinable right-linear terminating TRSs. In *Record of 2003 Tokai-section Joint Conference of the Eight Institutes of Electrical and Related Engineers*. (2003) 564 (in Japanese). This is an extended result of [22], and the English version of this paper is available from `http://www.sakabe.i.is.nagoya-u.ac.jp/~sakai/papers/`.
21. Romanenko, A.: Inversion and metacomputation. In *Proceedings of the Symposium on Partial Evaluation and Semantics-Based Program Manipulation*. Volume 26 of SIGPLAN Notices, ACM Press (1991) 12–22.
22. Sakai, M., Okamoto, K., Sakabe, T.: Innermost reductions find all normal forms on right-linear terminating overlay TRSs. In *Proceedings of the 3rd International Workshop on Reduction Strategies in Rewriting and Programming*. Valencia, Spain (2003) 79–88.
23. Secher, J.P., Sørensen, M.H.: From checking to inference via driving and dag grammars. In *Proceedings of the 2002 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation*. Volume 37 of SIGPLAN Notices, ACM (2002) 41–51.
24. Toyama, Y.: Confluent term rewriting systems with membership conditions. In *Proceedings of the 1st International Workshop on Conditional Term Rewriting Systems*. Volume 308 of LNCS, Springer (1987) 228–241.

A Proof of Theorem 9

In order to give a proof of Theorem 9, we need two lemmas (Lemma 24 and 25) in addition to some notations, concrete definitions of reductions, and some propositions associated with $\mathscr{T}$, reductions and substitutions.

A.1 Preparation

Let $R^{\Leftarrow}$ be a CTRS over signature $\mathcal{F}$. For terms s and t, we write $s \xrightarrow{*}^{!}_{R^{\Leftarrow}} t$ if $s \xrightarrow{*}_{R^{\Leftarrow}} t$ and $t \in NF_{R^{\Leftarrow}}(\mathcal{F},\mathcal{V})$, and similarly for other rewrite relations.

Let $\mathcal{F}$ be a signature and σ be a substitution. The *domain* and *range* of σ are denoted by $\mathcal{D}om(\sigma)$ and $\mathcal{R}an(\sigma)$, respectively. The *variable range* of σ is denoted by $\mathcal{VR}an(\sigma)$: $\mathcal{VR}an(\sigma) = \bigcup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(x\sigma)$. Let $T \subseteq \mathcal{T}(\mathcal{F},\mathcal{V})$. We call σ a *T-substitution* if $\mathcal{R}an(\sigma) \subseteq T$. When $\mathcal{D}om(\sigma) = \{x_1,\ldots,x_n\}$ and $\sigma(x_i) \equiv t_i$ for every $i \in \{1,\ldots,n\}$, we may write $\{\, x_1 \mapsto t_i, \ldots, x_n \mapsto t_n \,\}$ for σ. Let $R^{\Leftarrow}$ be a CTRS, $\rightarrow$ be a binary relation on terms, and let $l \rightarrow r \Leftarrow \mathcal{C}ond \in R^{\Leftarrow}$ such that $\mathcal{C}ond = s_1 \rightarrow t_1, \ldots, s_k \rightarrow t_k$. We write $\mathcal{C}ond(\sigma, \rightarrow)$ if $s_i\sigma \rightarrow t_i\sigma$ holds for every $i \in \{1,\ldots,k\}$.

The following proposition shows a property associated with the case (d) in Definition 5.

Proposition 19. *Let $\mathcal{F}$ ($= \mathcal{C} \uplus \mathcal{D}$) be a signature, t be a term, and X be a set of variables. If $\mathcal{V}ar(t) \cap X = \emptyset$, then $\mathscr{T}(\mathcal{L}ab(t,X)) = \langle\, t \,;\, \epsilon \,\rangle$.*

Proof (Sketch). From the definition, it holds clearly that for any n-ary defined symbol f, and for any terms $t_1,\ldots,t_n$, $\mathcal{V}ar(f(t_1,\ldots,t_n)) \cap X = \emptyset$ implies $\mathcal{L}ab(f(t_1,\ldots,t_n),X) = f_{\{1,\ldots,n\}}(\mathcal{L}ab(t_1,X),\ldots,\mathcal{L}ab(t_n,X))$. By using this claim, this proposition can be easily proved by induction on the term structure. □

The conditional parts of rules in the generated CTRS $\mathcal{I}nv(R)$ are in the form as follows.

Proposition 20. *Let R be a constructor TRS over a signature $\mathcal{F}$. Let $l \rightarrow r \Leftarrow s_1 \rightarrow t_1, \ldots, s_k \rightarrow t_k \in \mathcal{I}nv(R) \setminus R$. Then, the left- and right-hand sides satisfy the following:*

(a) $l \equiv f_I(u, w_1, \ldots, w_{|I|})$ and $u, w_1, \ldots, w_{|I|} \in \mathcal{T}(\mathcal{C}_R,\mathcal{V})$, and
(b) $r \equiv \mathsf{tp}_{|\bar{I}|}(w'_1, \ldots, w'_{\bar{I}_{|\bar{I}|}})$ and $w'_1, \ldots, w'_{\bar{I}_{|\bar{I}|}} \in \mathcal{T}(\mathcal{C}_R,\mathcal{V})$.

For every $i \in \{1,\ldots,k\}$, the condition $s_i \rightarrow t_i$ is either in the following forms:

(c) $s_i \in \mathcal{T}(\mathcal{F},\mathcal{V})$, and t_i is a variable appearing once among l, $t_1,\ldots,t_{i-1}$, or
(d) $s_i \equiv \overline{g}_J(y, s'_1, \ldots, s'_{|J|})$, $t_i \equiv \mathsf{tp}_{|\bar{J}|}(u_1,\ldots,u_{|\bar{J}|})$, $s'_j \in \mathcal{T}(\mathcal{F},\mathcal{V})$, $u_j \in \mathcal{T}(\mathcal{C}_R,\mathcal{V})$ and y is a variable appearing once among $t_1,\ldots,t_{i-1}$.

Proof (Sketch). Conditions in rules of $\mathcal{I}nv(R)$ are built in either (c) of Definition 5 or the definition of $\mathcal{I}nv\mathcal{R}ule$. In these two cases, we can see that conditions are built, satisfying either of the above forms. □

As shown by the CTRS R_5 in Example 8, our partial-inverse CTRS $\mathcal{I}nv(R)$ sometimes have parts of forward computations (terms to be evaluated by R) in conditional parts of some rules. The reduction sequences starting from such terms must be evaluated by the innermost strategy to be coincided with the corresponding parts of the forward computation in Theorem 9 which are evaluated by the innermost strategy. However, the naively-defined innermost reduction for CTRSs is illnatured because innermost redexes are undecidable in general. Therefore, we use a specially defined innermost-reduction for the execution of the CTRS $\mathcal{I}nv(R)$, where innermost redexes of R are executed preferentially in the evaluation. That is hierarchization of the conditional reduction on $\mathcal{I}nv(R)$ which consists of the innermost reduction on R, and the ordinary reduction by rules in $\mathcal{I}nv(R) \setminus R$.

Definition 21 (EV-normalized and R-innermost reduction). *Let R be a TRS over a signature $\mathcal{F}$, and $S^{\Leftarrow}$ be a CTRS over $\mathcal{F}$. The* n-level EV-normalized and R-innermost rewrite relation $\xrightarrow[(n),\mathrm{evn},R\text{-in}]{}_{S^{\Leftarrow}}$ *of $S^{\Leftarrow}$ is inductively defined as follows:*

- $\xrightarrow[(0),\mathrm{evn},R\text{-in}]{}_{S^{\Leftarrow}} = \emptyset$, *and*
- $\xrightarrow[(k+1),\mathrm{evn},R\text{-in}]{}_{S^{\Leftarrow}} = \{\ (C[l\sigma]_p, C[r\sigma]_p) \mid \rho : l \to r \Leftarrow \mathcal{C}ond \in R \cup S^{\Leftarrow}$, $\mathcal{C}ond(\sigma, \xrightarrow[(k),\mathrm{evn},R\text{-in}]{*}_{S^{\Leftarrow}})$, $\mathcal{R}an(\sigma|_{\mathcal{EV}ar(\rho)}) \subseteq NF_{R \cup S^{\Leftarrow}}(\mathcal{F},\mathcal{V})$, $(\forall t,\ t \lhd l\sigma$ *implies* $t \in NF_R(\mathcal{F},\mathcal{V})\ )\ \}$.

The EV-normalized and R-innermost rewrite relation $\xrightarrow[\mathrm{evn},R\text{-in}]{}_{S^{\Leftarrow}}$ *of $S^{\Leftarrow}$ is defined as* $\xrightarrow[\mathrm{evn},R\text{-in}]{}_{S^{\Leftarrow}} = \bigcup_{k \geq 0} \xrightarrow[(k),\mathrm{evn},R\text{-in}]{}_{S^{\Leftarrow}}$.

It is clear that $\xrightarrow[\mathrm{in}]{}_R \subseteq \xrightarrow[\mathrm{evn},R\text{-in}]{}_{S^{\Leftarrow}}$. Since the CTRS $\mathcal{I}nv(R)$ contains all rules of R, the following proposition holds.

Proposition 22. *Let R be a constructor TRS over a signature $\mathcal{F}$. For all terms s and t in $\mathcal{T}(\mathcal{F},\mathcal{V})$,*

(a) $s \to_R t$ if and only if $s \to_{\mathcal{I}nv(R)} t$, and
(b) $s \xrightarrow[\mathrm{in}]{}_R t$ if and only if $s \xrightarrow[\mathrm{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} t$.

Proof. The *only-if* parts of the both hold trivially from the fact that $R \subseteq \mathcal{I}nv(R)$. We consider the *if* part of (a). Suppose that $s \to_{\mathcal{I}nv(R)} t$. Since R has no extra variable and s has no defined symbol of $\mathcal{I}nv(R) \setminus R$, the rule used in $s \to_{\mathcal{I}nv(R)} t$ must be in R. Therefore, we have $s \to_R t$. The *if* part of (b) can be shown similarly. □

The reduction $\xrightarrow[\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)}$ has the following property.

Proposition 23. *Let R be a constructor TRS over a signature $\mathcal{F}$. Let $f \in \mathcal{D}_R$, I be a set of indexes for f, and $s, s_1, \ldots, s_{|I|}$, $t_1, \ldots, t_{|\bar{I}|} \in \mathcal{T}(\mathcal{F},\mathcal{V})$. If $\overline{f}_I(s, s_1, \ldots, s_{|I|}) \xrightarrow[\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$ holds, then there exist a (conditional) rewrite rule $\overline{f}_I(u, u_1, \ldots, u_{|I|}) \to \mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|}) \Leftarrow \mathcal{C}ond \in \mathcal{I}nv(R) \setminus R$ and an $NF_R(\mathcal{F},\mathcal{V})$-substitution σ such that*

– $\overline{f}_I(s, s_1, \ldots, s_{|I|}) \xrightarrow[\text{in}]{*}{}^!_R \overline{f}_I(u, u_1, \ldots, u_{|I|})\sigma$
$\xrightarrow[\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)\setminus R} \mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|})\sigma \equiv \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$,
– $t_1, \ldots, t_{|\bar{I}|} \in NF_{\mathcal{I}nv(R)}(\overline{\mathcal{F}}, \mathcal{V})$, *and*
– $\mathcal{C}ond(\sigma, \xrightarrow[\text{in}]{*}{}^!_R \cup \Big(\xrightarrow[\text{in}]{*}{}^!_R \cdot (\xrightarrow[\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)} \setminus \xrightarrow[\text{in}]{}_R)\Big))$.

Proof. We prove this proposition by induction on the level k of the reduction $\xrightarrow[(k),\text{evn},R\text{-in}]{*}{}_{\mathcal{I}nv(R)}$. In the sequel, we abbreviate $\xrightarrow[\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)} \setminus \xrightarrow[\text{in}]{}_R$ to $\xrightarrow[\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)\setminus R}$.

Suppose that $\overline{f}_I(s, s_1, \ldots, s_{|I|}) \xrightarrow[(k),\text{evn},R\text{-in}]{*}{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$. It follows from the construction of $\mathcal{I}nv(R)$ and the definition of $\xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)}$ that there exist a rule $\overline{f}_I(u, u_1, \ldots, u_{|I|}) \to \mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|}) \Leftarrow \mathcal{C}ond \in \mathcal{I}nv(R) \setminus R$ and a $\mathcal{T}(\mathcal{F}, \mathcal{V})$-substitution σ such that

– $\overline{f}_I(s, s_1, \ldots, s_{|I|}) \xrightarrow[\text{in}]{*}{}^!_R \overline{f}_I(u, u_1, \ldots, u_{|I|})\sigma$
$\xrightarrow[(k),\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)\setminus R} \mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|})\sigma \xrightarrow[(k),\text{evn},R\text{-in}]{*}{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$,
– $\mathcal{R}an(\sigma|_{\mathcal{V}ar(u,u_1,\ldots,u_{|I|})}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$ and
– $\mathcal{C}ond(\sigma, \xrightarrow[(k-1),\text{evn},R\text{-in}]{*}{}_{\mathcal{I}nv(R)})$.

In the sequel, we show that $\mathcal{R}an(\sigma|_{\mathcal{V}ar(\mathcal{C}ond,w_1,\ldots,w_{|\bar{I}|})}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$ (that is, σ is an $NF_R(\mathcal{F}, \mathcal{V})$-substitution) and $\mathcal{C}ond(\sigma, \xrightarrow[\text{in}]{*}{}^!_R \cup \Big(\xrightarrow[\text{in}]{*}{}^!_R \cdot \xrightarrow[\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)\setminus R}\Big))$.

The case that the applied rule is unconditional is trivial. We consider the remaining case. Let $\mathcal{C}ond = l_1 \to r_1, \ldots, l_m \to r_m$. We first show that $l_1\sigma \;(\xrightarrow[\text{in}]{*}{}^!_R \cup \Big(\xrightarrow[\text{in}]{*}{}^!_R \cdot \xrightarrow[\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)\setminus R}\Big))\; r_1\sigma$ and $\mathcal{R}an(\sigma|_{\mathcal{V}ar(l_1,r_1)}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$. Since the CTRS $\mathcal{I}nv(R)$ is deterministic, we have $\mathcal{V}ar(l_1) \subseteq \mathcal{V}ar(u, u_1, \ldots, u_{|I|})$ and hence $\mathcal{R}an(\sigma|_{\mathcal{V}ar(l_1)}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$.

– Consider the subcase that $l_1 \to r_1$ is in the form of Proposition 20 (c). In this subcase, we can suppose that $l_1 \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and r_1 is a variable appearing once in either of $u, u_1, \ldots, u_{|I|}$. Then, $r_1\sigma$ is a normal form of R and hence $\mathcal{R}an(\sigma|_{\mathcal{V}ar(r_1)}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$. It follows from Proposition 22 that $l_1\sigma \xrightarrow[\text{in}]{*}{}^!_R r_1\sigma$.
– Consider the remaining subcase that $l_1 \to r_1$ is in the form of Proposition 20 (d). In this subcase, we can suppose that
  - $l_1 \equiv \overline{g}_J(z, s'_1, \ldots, s'_{|J|})$,
  - z is a variable in either of $u, u_1, \ldots, u_{|I|}$ and
  - $r_1 \equiv \mathsf{tp}_{|\bar{J}|}(t'_1, \ldots, t'_{|\bar{J}|})$.

  Since $\mathcal{C}ond(\sigma, \xrightarrow[(k-1),\text{evn},R\text{-in}]{*}{}_{\mathcal{I}nv(R)})$ holds, $l_1\sigma \xrightarrow[(k-1),\text{evn},R\text{-in}]{*}{}_{\mathcal{I}nv(R)} r_1\sigma$, and hence $\overline{g}_J(z\sigma, s'_1\sigma, \ldots, s'_{|J|}\sigma) \xrightarrow[(k-1),\text{evn},R\text{-in}]{*}{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{J}|}(t'_1\sigma, \ldots, t'_{|\bar{J}|}\sigma)$. By the induction hypothesis, there exist an $NF_R(\mathcal{F}, \mathcal{V})$-substitution σ' and a rule $\overline{g}_J(u', u', \ldots, u'_{|J|}) \to \mathsf{tp}_{|\bar{J}|}(w'_1, \ldots, w'_{|\bar{J}|}) \Leftarrow \mathcal{C}ond' \in \mathcal{I}nv(R) \setminus R$ such that
  - $\overline{g}_J(z\sigma, s'_1\sigma, \ldots, s'_{|J|}\sigma) \xrightarrow[\text{in}]{*}{}^!_R \overline{g}_J(u', u'_1, \ldots, u'_{|J|})\sigma'$
  $\xrightarrow[\text{evn},R\text{-in}]{\varepsilon}{}_{\mathcal{I}nv(R)\setminus R} \mathsf{tp}_{|\bar{J}|}(w'_1, \ldots, w'_{|\bar{J}|})\sigma' \equiv \mathsf{tp}_{|\bar{J}|}(t'_1\sigma, \ldots, t'_{|\bar{J}|}\sigma)$,

- $t'_1\sigma, \ldots, t'_{|\bar{J}|}\sigma \in NF_{\mathcal{I}nv(R)}(\overline{\mathcal{F}}, \mathcal{V})$, and
- $\mathcal{C}ond(\sigma', \xrightarrow[\text{in}]{*}{}^!_R \cup \left(\xrightarrow[\text{in}]{*}{}^!_R \cdot \xrightarrow[\text{evn},R\text{-in}]{}{}^{\varepsilon}_{\mathcal{I}nv(R)\setminus R}\right))$.

Hence, we have $\mathcal{R}an(\sigma|_{\mathcal{V}ar(r_1)}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$.

Thus, we have $l_1\sigma \ (\xrightarrow[\text{in}]{*}{}^!_R \cup \left(\xrightarrow[\text{in}]{*}{}^!_R \cdot \xrightarrow[\text{evn},R\text{-in}]{}{}^{\varepsilon}_{\mathcal{I}nv(R)\setminus R}\right))\ r_1\sigma$ and $\mathcal{R}an(\sigma|_{\mathcal{V}ar(l_1,r_1)}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$. Similarly for every $i \in \{2, \ldots, m\}$, we can show that $l_i\sigma \ (\xrightarrow[\text{in}]{*}{}^!_R \cup \left(\xrightarrow[\text{in}]{*}{}^!_R \cdot \xrightarrow[\text{evn},R\text{-in}]{}{}^{\varepsilon}_{\mathcal{I}nv(R)\setminus R}\right))\ r_i\sigma$ and $\mathcal{R}an(\sigma|_{\mathcal{V}ar(l_i,r_i)}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$.

Let $X = \mathcal{V}ar(u, u_1, \ldots, u_{|I|}, \mathcal{C}ond)$. It follows from the definition of the reduction $\xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)}$ that $\mathcal{R}an(\sigma|_{\mathcal{V}ar(w_1,\ldots,w_{|\bar{I}|})\setminus X}) \subseteq NF_{\mathcal{I}nv(R)}(\overline{\mathcal{F}}, \mathcal{V})$. On the other hand, it follows from Proposition 20 (b) that $w_1, \ldots, w_{|I|}$ are constructor terms of R. These imply that $\mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|})\sigma \in NF_{\mathcal{I}nv(R)}(\overline{\mathcal{F}}, \mathcal{V})$, and hence $\mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|})\sigma \equiv \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$.

Therefore, we have $\mathcal{R}an(\sigma|_{\mathcal{V}ar(\mathcal{C}ond,w_1,\ldots,w_{|\bar{I}|})}) \subseteq NF_R(\mathcal{F}, \mathcal{V})$ and $l_i\sigma \ (\xrightarrow[\text{in}]{*}{}^!_R \cup \left(\xrightarrow[\text{in}]{*}{}^!_R \cdot \xrightarrow[\text{evn},R\text{-in}]{}{}^{\varepsilon}_{\mathcal{I}nv(R)\setminus R}\right))\ r_i\sigma$ for $i \in \{1, \ldots, m\}$. □

A.2 Lemmas

Lemma 24. *Let R be a constructor TRS over a signature $\mathcal{F}$, t be a term, X be a set of variables, $\mathscr{T}(\mathcal{L}ab(t, X)) = \langle\, u\, ;\, \mathcal{C}ond\,\rangle$, θ be an $NF_R(\mathcal{F}, \mathcal{V})$-substitution such that $\mathcal{D}om(\theta) \cap (\mathcal{V}ar(u, \mathcal{C}ond) \setminus \mathcal{V}ar(t)) = \emptyset$, and s be a normal form of R. If $t\theta \xrightarrow[\text{in}]{*}_R s$, then there exists an $NF_R(\mathcal{F}, \mathcal{V})$-substitution σ such that $\mathcal{D}om(\sigma) \subseteq \mathcal{V}ar(u, \mathcal{C}ond) \setminus \mathcal{V}ar(t)$, $u(\sigma \cup \theta) \xrightarrow[\text{in}]{*}_R s$, and $\mathcal{C}ond(\sigma \cup \theta, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$.*

Proof. We prove this lemma by induction on the lexicographic combination of the number k of steps of $t\theta \xrightarrow[\text{in}]{k}_R s$, and the structure of the term t. We consider the following cases:

(i) $t \in \mathcal{V}$,
(ii) $\mathtt{root}(t) \in \mathcal{C}_R$,
(iii) $\mathtt{root}(t) \in \mathcal{D}_R$ and $\mathcal{V}ar(t) \cap X = \emptyset$, and
(iv) $\mathtt{root}(t) \in \mathcal{D}_R$ and $\mathcal{V}ar(t) \cap X \neq \emptyset$

In the cases (i) and (iii), it is trivial since $t \equiv u$ and $\mathcal{C}ond = \epsilon$ (by Proposition 19). In the case (ii), it can be easily proved by using the induction hypothesis.

We consider the remaining case (iv). Suppose that $t \equiv f(t_1, \ldots, t_n)$ with $f \in \mathcal{D}_R$, and $\mathcal{V}ar(t) \cap X \neq \emptyset$. From (c) of Definition 5 and Proposition 19, we can write $u \equiv y \in \mathcal{V}$,

$$\mathcal{C}ond = \overline{f}_I(y, t_{I_1}, \ldots, t_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(u_1, \ldots, u_{|\bar{I}|}), \mathcal{C}ond'_1, \ldots, \mathcal{C}ond'_{|\bar{I}|},$$

$$u_j = C_j[y_{j,1}, \ldots, y_{j,m_j}],$$

$$\mathscr{T}(\mathcal{L}ab(t_{\bar{I}_j}, X)) = \langle\, C_j[\![u_{j,1}, \ldots, u_{j,m_j}]\!]\, ;\, \mathcal{C}ond_j\,\rangle \text{ for } 1 \leq j \leq |\bar{I}|, \text{ and,}$$

$$\mathcal{C}ond'_j = \mathcal{C}ond_j, u_{j,1} \to y_{j,1}, \ldots, u_{j,m_j} \to y_{j,m_j}.$$

where y and $y_{1,1},\ldots,y_{|\bar{I}|,m_{|\bar{I}|}}$ are fresh variables, and variables introduced in $\mathscr{T}(\mathcal{L}ab(t_{\bar{I}_i},X))$ and $\mathscr{T}(\mathcal{L}ab(t_{\bar{I}_j},X))$ $(i \neq j)$ are disjoint, that is,

$$y \notin \mathcal{V}ar(t_1,\ldots,t_n,u_1,\ldots,u_{|\bar{I}|},\mathcal{C}ond'_1,\ldots,\mathcal{C}ond'_{|\bar{I}|}) \text{ and} \tag{1}$$

$$\Big(\mathcal{V}ar(u_j,\mathcal{C}ond_j)\setminus\mathcal{V}ar(t_{\bar{I}_j})\Big)\cap\mathcal{V}ar(t_{\bar{I}_{j'}},u_{j'},\mathcal{C}ond_{j'})=\emptyset \text{ for } j\neq j'. \tag{2}$$

For the sequence $f(t_1,\ldots,t_n)\theta \xrightarrow[\text{in}]{k}_R s$, there exist normal forms s_i of R and numbers k',k'' such that $k'+k''=k$ and

$$f(t_1,\ldots,t_n)\theta \xrightarrow[\text{in}]{k'}_R f(s_1,\ldots,s_n) \xrightarrow[\text{in}]{k''}_R s.$$

Hence, there exist numbers k_i such that $k_i \leq k'$ and $t_i\theta \xrightarrow[\text{in}]{k_i}_R s_i$. It is clear that $\mathcal{D}om(\theta)\cap(\mathcal{V}ar(C_j[\![u_{j,1},\ldots,u_{j,m_j}]\!],\mathcal{C}ond_j\setminus\mathcal{V}ar(t_{\bar{I}_j}))=\emptyset$. By the induction hypothesis, there exists an $NF_R(\mathcal{F},\mathcal{V})$-substitution σ_j such that $\mathcal{C}ond_j(\sigma_j\cup\theta,\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$, $(C_j[\![u_{j,1},\ldots,u_{j,m_j}]\!])(\sigma_j\cup\theta)\xrightarrow[\text{in}]{*}_R s_{\bar{I}_j}$, and

$$\mathcal{D}om(\sigma_j)\subseteq\mathcal{V}ar(\mathcal{C}ond_j)\setminus\mathcal{V}ar(t_{\bar{I}_j}). \tag{3}$$

Now we can suppose that $s_{\bar{I}_j}\equiv C'_j[\![s'_{j,1},\ldots,s'_{j,m_j}]\!]$ where $C'_j\equiv C_j(\sigma_j\cup\theta)$. Then, we have $u_{j,k}(\sigma_j\cup\theta)\xrightarrow[\text{in}]{*}_R s'_{j,k}$.

Let $\sigma'_j=\sigma_j\cup\{\,y_{j,k}\mapsto s'_{j,k}\mid 1\leq k\leq m_j\,\}$. Since $y_{j,k}$ is fresh, σ'_j is a substitution. It is clear that σ'_j is an $NF_R(\mathcal{F},\mathcal{V})$-substitution. It is also clear that $\mathcal{C}ond(\sigma'_j\cup\theta,\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$ and $u_{j,k}(\sigma'_j\cup\theta)\xrightarrow[\text{in}]{*}_R s'_{j,k}\equiv y_{j,k}(\sigma'_j\cup\theta)$. It follows from Proposition 20 that u_j is a constructor term of R, and hence $u_j(\sigma'_j\cup\theta)\equiv s_{\bar{I}_j}$. Hence, we have $\mathcal{C}ond_j(\sigma'_j\cup\theta,\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$ and $(C_j[\![u_{j,1},\ldots,u_{j,m_j}]\!])(\sigma'_j\cup\theta)$ $\xrightarrow[\text{in}]{*}_R u_j(\sigma'_j\cup\theta)$, and hence $\mathcal{C}ond'_j(\sigma'_j\cup\theta,\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$.

Since $s_1,\ldots,s_n$ are normal forms of R, we consider the following two cases, that is, whether $f(s_1,\ldots,s_n)$ is a redex of R or not.

– Consider the subcase that $f(s_1,\ldots,s_n)$ is not a redex of R, that is, a normal form of R. Now we have $s\equiv f(s_1,\ldots,s_n)$. Let $\sigma=\{y\mapsto s\}\cup\bigcup_{j=1}^{|\bar{I}|}\sigma'_j$. It follows from (1)–(3) that $y\notin\mathcal{D}om(\sigma_j)$ and $\mathcal{D}om(\sigma_j)\cap\mathcal{D}om(\sigma_{j'})=\emptyset$ for $j\neq j'$. Hence σ is a substitution. It is clear that $\mathcal{D}om(\sigma)\subseteq\mathcal{V}ar(\mathcal{C}ond)\setminus\mathcal{V}ar(t)$, $u(\sigma\cup\theta)\xrightarrow[\text{in}]{*}_R s$, and $\mathcal{C}ond_j(\sigma\cup\theta,\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. It is also clear that σ is an $NF_R(\mathcal{F},\mathcal{V})$-substitution. On the other hand, we have the inverse-property rule $\overline{f}_I(f(x_1,\ldots,x_n),x_{I_1},\ldots,x_{I_{|I|}})\to\mathsf{tp}_{|\bar{I}|}(x_{\bar{I}_1},\ldots,x_{\bar{I}_{|\bar{I}|}})\in\mathcal{I}nv(R)$ and we have $R\subseteq\mathcal{I}nv(R)$, and hence

$$\begin{aligned}\overline{f}_I(y,t_{I_1},\ldots,t_{I_{|I|}})(\sigma\cup\theta) &\equiv \overline{f}_I(y\sigma,t_{I_1}\theta,\ldots,t_{I_{|I|}}\theta)\\ &\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} \overline{f}_I(s,s_{I_1},\ldots,s_{I_{|I|}})\\ &\xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(s_{\bar{I}_1},\ldots,s_{\bar{I}_{|\bar{I}|}})\\ &\equiv \mathsf{tp}_{|\bar{I}|}(u_1,\ldots,u_{|\bar{I}|})\sigma.\end{aligned}$$

– Consider the remaining subcase that $f(s_1,\ldots,s_n)$ is a redex of R. In this subcase, there exist a rewrite rule $\rho : f(w_1,\ldots,w_n) \to r \in R$ and an $\mathit{NF}_R(\mathcal{F},X)$-substitution θ_0 such that

$$\mathcal{D}om(\theta_0) \subseteq \mathcal{V}ar(w_1,\ldots,w_n), \text{ and} \tag{4}$$

$$f(s_1,\ldots,s_n) \equiv f(w_1,\ldots,w_n)\theta_0 \xrightarrow[\text{in}]{\varepsilon}_R r\theta_0 \xrightarrow[\text{in}]{k''-1}_R s.$$

Let $\mathscr{T}(\mathcal{L}ab(r,\mathcal{UV}ar(\rho,I))) = \langle C_0[\![r_1,\ldots,r_m]\!] ; \mathcal{C}ond_0 \rangle$. Then, we have a rule

$$\begin{aligned}\overline{f}_I(C_0[z_1,\ldots,z_m], w_{I_1},\ldots,w_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}}) \\ \Leftarrow r_1 \to z_1,\ldots,r_m \to z_m, \mathcal{C}ond_0 \in \mathcal{I}nv(R)\end{aligned}$$

which satisfies $(\mathcal{V}ar(C_0[\![r_1,\ldots,r_m]\!], \mathcal{C}ond_0) \setminus \mathcal{V}ar(r)) \cap \mathcal{V}ar(w_1,\ldots,w_n) = \emptyset$. By the induction hypothesis, there exists an $\mathit{NF}_R(\mathcal{F},\mathcal{V})$-substitution σ_0 such that $\mathcal{D}om(\sigma_0) \subseteq \mathcal{V}ar(\mathcal{C}ond_0) \setminus \mathcal{V}ar(r)$, $C_0[\![r_1,\ldots,r_m]\!](\sigma_0 \cup \theta_0) \xrightarrow[\text{in}]{*}_R s$, and $\mathcal{C}ond_0(\sigma_0 \cup \theta_0, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. Since $C_0[,\ldots,]$ is a constructor context and $\sigma_0 \cup \theta_0$ is an $\mathit{NF}_R(\mathcal{F},\mathcal{V})$-substitution, we have $s \equiv C'_0[s_{0,1},\ldots,s_{0,m}]$ where $C'_0 \equiv C_0(\sigma_0 \cup \theta_0)$. Hence, we have $r_i(\sigma_0 \cup \theta_0) \xrightarrow[\text{in}]{*}_R s_{0,i}$. Let $\sigma'_0 = \{\, z_i \mapsto s_{0,i} \mid 1 \le i \le m \,\} \cup \sigma_0$. Then, σ'_0 is clearly an $\mathit{NF}_R(\mathcal{F},\mathcal{V})$-substitution. It is obvious that $r_i(\sigma'_0 \cup \theta) \xrightarrow[\text{in}]{*}_R s_{0,i} \equiv z_i(\sigma'_0 \cup \theta)$ and $\mathcal{C}ond_0(\sigma'_0 \cup \theta, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. Hence we have

$$\begin{aligned}&\overline{f}_I(C_0[\![z_1,\ldots,z_m]\!], w_{I_1},\ldots,w_{I_{|I|}})(\sigma'_0 \cup \theta) \\ &\qquad \xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}})(\sigma'_0 \cup \theta).\end{aligned}$$

This implies $\overline{f}_I(s, s_{I_1},\ldots,s_{I_{|I|}}) \xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(s_{\bar{I}_1},\ldots,s_{\bar{I}_{|\bar{I}|}})$.

Let $\sigma = \{y \mapsto s\} \cup \bigcup_{j=1}^{|\bar{I}|} \sigma'_j$. Similarly in the former subcase, σ is a substitution. It is clear that $\mathcal{D}om(\sigma) \subseteq \mathcal{V}ar(\mathcal{C}ond) \setminus \mathcal{V}ar(t)$, $u\sigma \equiv s$, and $\mathcal{C}ond_j(\sigma \cup \theta, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. It is also clear that σ is an $\mathit{NF}_R(\mathcal{F},\mathcal{V})$-substitution.

Now we have the following sequence:

$$\begin{aligned}\overline{f}_I(y, u_{I_1},\ldots,u_{I_{|I|}})\sigma &\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} \overline{f}_I(s, s_{I_1},\ldots,s_{I_{|I|}}) \\ &\xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(s_{\bar{I}_1},\ldots,s_{\bar{I}_{|\bar{I}|}}) \equiv \mathsf{tp}_{|\bar{I}|}(u_1,\ldots,u_{|\bar{I}|})\sigma.\end{aligned}$$

In the both subcases above, we have shown that the first condition in $\mathcal{C}ond$ is satisfied by σ and $\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)}$. Therefore, we have $u(\sigma \cup \theta) \xrightarrow[\text{in}]{*}_R s$ and $\mathcal{C}ond(\sigma \cup \theta, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$ holds. $\square$

Lemma 25. *Let R be a constructor TRS over a signature $\mathcal{F}$, t be a term, and X be a set of variables. Let $\mathscr{T}(\mathcal{L}ab(t,X)) = \langle u ; \mathcal{C}ond \rangle$. Let σ be an $\mathit{NF}_R(\mathcal{F},\mathcal{V})$-substitution. If $\mathcal{C}ond(\sigma, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$, then $t\sigma \xrightarrow[\text{in}]{*}_R u\sigma$.*

Proof. We prove this lemma by induction on the lexicographic combination of the level k of rewrite relation $\mathcal{C}ond(\sigma, \xrightarrow[(k),\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$, and the structure of the term t. We consider the following cases:

(i) $t \in \mathcal{V}$,
(ii) $\texttt{root}(t) \in \mathcal{C}_R$,
(iii) $\texttt{root}(t) \in \mathcal{D}_R$ and $\mathcal{V}ar(t) \cap X = \emptyset$, and
(iv) $\texttt{root}(t) \in \mathcal{D}_R$ and $\mathcal{V}ar(t) \cap X \neq \emptyset$

In the cases (i) and (iii), it is trivial since $t \equiv u$. In the case (ii), it can be easily proved by using the induction hypothesis.

We consider the remaining case. Suppose that $t \equiv f(t_1, \ldots, t_n)$ with $f \in \mathcal{D}_R$, and $\mathcal{V}ar(t) \cap X \neq \emptyset$. From (c) of Definition 5 and Proposition 19, we can write $u \equiv y \in \mathcal{V}$,

$$\mathcal{C}ond = \overline{f}_I(y, t_{I_1}, \ldots, t_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(u_1, \ldots, u_{|\bar{I}|}), \mathcal{C}ond'_1, \ldots, \mathcal{C}ond'_{|\bar{I}|},$$

$$u_j = C_j[y_{j,1}, \ldots, y_{j,m_j}],$$

$$\mathscr{T}(\mathcal{L}ab(t_{\bar{I}_j}, X)) = \langle\, C_j[\![u_{j,1}, \ldots, u_{j,m_j}]\!] \,;\, \mathcal{C}ond_j \,\rangle \text{ for } 1 \le j \le |\bar{I}|, \text{ and,}$$

$$\mathcal{C}ond'_j = \mathcal{C}ond_j, u_{j,1} \to y_{j,1}, \ldots, u_{j,m_j} \to y_{j,m_j}.$$

where y and $y_{1,1}, \ldots, y_{|\bar{I}|,m_{|\bar{I}|}}$ are fresh variables, and variables introduced in $\mathscr{T}(\mathcal{L}ab(t_{\bar{I}_i}, X))$ and $\mathscr{T}(\mathcal{L}ab(t_{\bar{I}_j}, X))$ $(i \neq j)$ are disjoint, that is,

$$y \notin \mathcal{V}ar(t_1, \ldots, t_n, u_1, \ldots, u_{|\bar{I}|}, \mathcal{C}ond'_1, \ldots, \mathcal{C}ond'_{|\bar{I}|}) \text{ and}$$

$$\Big(\mathcal{V}ar(u_j, \mathcal{C}ond_j) \setminus \mathcal{V}ar(t_{\bar{I}_j})\Big) \cap \mathcal{V}ar(t_{\bar{I}_{j'}}, u_{j'}, \mathcal{C}ond_{j'}) = \emptyset \text{ for } j \neq j'.$$

It follows from $\mathcal{C}ond(\sigma, \xrightarrow[(k),\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$ that $\mathcal{C}ond'_j(\sigma, \xrightarrow[(k),\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$, and hence $\mathcal{C}ond_j(\sigma, \xrightarrow[(k),\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$ and $u_{j,k}\sigma \xrightarrow[\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} y_{j,k}\sigma$. It follows from $u_{j,k}\sigma \xrightarrow[\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} y_{j,k}\sigma$ that $(C_j[\![u_{j,1}, \ldots, u_{j,m_j}]\!])\sigma \xrightarrow[\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} u_j\sigma \in NF_R(\mathcal{F}, \mathcal{V})$. Hence, it follows from Proposition 22 that $(C_j[\![u_{j,1}, \ldots, u_{j,m_j}]\!])\sigma \xrightarrow[\mathrm{in}]{*}_R u_j\sigma$. By the induction hypothesis, we have $t_{\bar{I}_j}\sigma \xrightarrow[\mathrm{in}]{*}_R (C_j[\![u_{j,1}, \ldots, u_{j,m_j}]\!])\sigma$, and hence $t_{\bar{I}_j}\sigma \xrightarrow[\mathrm{in}]{*}_R u_j\sigma$.

From Proposition 23, we have an $NF_R(\mathcal{F}, \mathcal{V})$-substitution θ and the following sequence:

$$\begin{aligned} \overline{f}_I(y, t_{I_1}, \ldots, t_{I_{|I|}})\sigma &\xrightarrow[\mathrm{in}]{*}_R \overline{f}_I(s, s_{I_1}, \ldots, s_{I_{|I|}}) \\ &\equiv l\theta \xrightarrow[(k),\mathrm{evn},R\text{-in}]{\varepsilon}_{\mathcal{I}nv(R)} r\theta \equiv \mathsf{tp}_{|\bar{I}|}(u_1, \ldots, u_{|\bar{I}|})\sigma \end{aligned}$$

where $l \to r \Leftarrow \mathcal{C}ond_0 \in \mathcal{I}nv(R) \setminus R$, $\mathcal{C}ond_0(\theta, \xrightarrow[(k-1),\mathrm{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$, and s_{I_j} is a normal form of R. Hence we have $t_{I_j}\sigma \xrightarrow[\mathrm{in}]{*}_R s_{I_j}$. We also have $y\sigma \equiv s$ since σ is an $NF_R(\mathcal{F}, \mathcal{V})$-substitution.

Next, we consider the form of the applied rule $l \to r \Leftarrow \mathcal{C}ond_0$ at the last one-step $\xrightarrow[(k),\mathrm{evn},R\text{-in}]{\varepsilon}_{\mathcal{I}nv(R)}$. Let $s_{\bar{I}_j} \equiv u_{\bar{I}_j}\sigma$.

- Consider the subcase that $l \to r \Leftarrow \mathcal{C}ond_0$ is the following inverse-property rule:

$$\overline{f}_I(f(x_1, \ldots, x_n), x_{I_1}, \ldots, x_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(x_{\bar{I}_1}, \ldots, x_{\bar{I}_{|\bar{I}|}}) \in \mathcal{I}nv(R).$$

In this subcase, we have $y\sigma \equiv f(s_1,\ldots,s_n)$. Therefore, we have

$$t\sigma \equiv f(t_1,\ldots,t_n)\sigma \xrightarrow[\text{in}]{*}_R f(s_1,\ldots,s_n) \equiv y\sigma \equiv s.$$

– Consider the remaining subcase. We suppose that $l \to r \Leftarrow \mathit{Cond}_0$ is the following rule (which is not the inverse-property rule):

$$\overline{f}_I(r', w_{I_1},\ldots,w_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}}) \Leftarrow \mathit{Cond}_0 \in \mathcal{I}nv(R) \setminus R.$$

In addition, we have by definition the rule $\rho : f(w_1,\ldots,w_n) \to r \in R$ such that $\mathscr{T}(\mathcal{L}ab(r, \mathcal{UV}ar(\rho, I))) = \langle\, C[\![r_1,\ldots,r_m]\!]\ ;\ \mathit{Cond}'_0\,\rangle$, $r' \equiv C[z_1,\ldots,z_m]$ and $\mathit{Cond}_0 = r_1 \to z_1,\ldots,r_m \to z_m, \mathit{Cond}'_0$. Now we have

$$\begin{aligned}\overline{f}_I(y\sigma, s_{I_1},\ldots,s_{I_{|I|}}) &\equiv \overline{f}_I(r', w_{I_1},\ldots,w_{I_{|I|}})\theta \\ &\xrightarrow[(k),\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(w_{\bar{I}_1},\ldots,w_{\bar{I}_{|\bar{I}|}})\theta \\ &\equiv \mathsf{tp}_{|\bar{I}|}(s_{\bar{I}_1},\ldots,s_{\bar{I}_{|\bar{I}|}})\end{aligned}$$

and $\mathit{Cond}_0(\theta, \xrightarrow[(k-1),\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. This implies that $r_i\theta \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} z_i\theta$ and $\mathit{Cond}'_0(\theta, \xrightarrow[(k-1),\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. It follows from Proposition 22 that we have $r_i\theta \xrightarrow[\text{in}]{*}_R z_i\theta$, and hence $(C[\![r_1,\ldots,r_m]\!])\theta \xrightarrow[\text{in}]{*}_R r'\theta \equiv y\sigma$. Moreover, by the induction hypothesis, we have $r\theta \xrightarrow[\text{in}]{*}_R (C[\![r_1,\ldots,r_m]\!])\theta$. Therefore, the following sequence holds:

$$\begin{aligned}t\sigma &\equiv f(t_1,\ldots,t_n)\sigma \xrightarrow[\text{in}]{*}_R f(s_1,\ldots,s_n) \\ &\equiv f(w_1,\ldots,w_n)\theta \xrightarrow[\text{in}]{}_R r\theta \xrightarrow[\text{in}]{*}_R (C[\![r_1,\ldots,r_m]\!])\theta \xrightarrow[\text{in}]{*}_R y\sigma.\end{aligned}$$

□

A.3 Proof of Theorem 9

We finally prove Theorem 9.

Theorem 9. *Let R be a constructor TRS over a signature $\mathcal{F}$. Let $(f, I) \in \mathbb{I}_{\mathcal{D}_R}$ and $s, t_1,\ldots,t_n$ be normal forms of R. Then, $f(t_1,\ldots,t_n) \xrightarrow[\text{in}]{*}_R s$ if and only if $\overline{f}_I(s, t_{I_1},\ldots,t_{I_{|I|}}) \xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1},\ldots,t_{\bar{I}_{|\bar{I}|}})$.*

Proof. This theorem can be proved by using Lemma 24 and 25 as follows.

Let $t \equiv f(x_1,\ldots,x_n)$ and θ be an $\mathit{NF}_R(\mathcal{F},\mathcal{V})$-substitution such that $\theta = \{\, x_i \mapsto t_i \mid 1 \le i \le n \,\}$ (that is, $t\theta \equiv f(t_1,\ldots,t_n)$). From the definition of $\mathscr{T}$, we can assume that all of the following:

(a) $\mathscr{T}\big(f(x_1,\ldots,x_n), \{x_j \mid j \in \bar{I}\}\big) = \langle\, y\,;\, \mathit{Cond}\,\rangle$, $\mathit{Cond} = \overline{f}_I(y, x_{I_1},\ldots,x_{I_{|I|}}) \to \mathsf{tp}_{|\bar{I}|}(x_{\bar{I}_1},\ldots,x_{\bar{I}_{|\bar{I}|}})$, and
(b) y is a fresh variable, that is, $y \notin \bigcup_{i=1}^{n} \mathcal{V}ar(t_i, x_i)$.

We first prove the *only-if*-part. Suppose that $f(t_1,\ldots,t_n) \xrightarrow[\text{in}]{*}_R s$. Since $\mathcal{D}om(\theta) = \{x_1,\ldots,x_n\}$ and $\mathcal{V}ar(u, \mathit{Cond}) \setminus \mathcal{V}ar(t) = \{y\}$, we have $\mathcal{D}om(\theta) \cap$

$(\mathcal{V}ar(u,\mathcal{C}ond) \setminus \mathcal{V}ar(t)) = \emptyset$. Hence, from Lemma 24, there exists an $NF_R(\mathcal{F},\mathcal{V})$-substitution σ such that $y(\sigma\cup\theta) \xrightarrow[\text{in}]{*}_R s$, $\mathcal{C}ond(\sigma\cup\theta, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$ and $\mathcal{D}om(\sigma) \subseteq \{y\}$. Since $y(\sigma\cup\theta) \xrightarrow[\text{in}]{*}_R s$ and σ is an $NF_R(\mathcal{F},\mathcal{V})$-substitution, we have $y(\sigma\cup\theta) \equiv s$, and hence $\sigma = \{y \mapsto s\}$ and

$$\begin{aligned}\overline{f}_I(s, t_{I_1}, \ldots, t_{I_{|I|}}) &\equiv \overline{f}_I(y, x_{I_1}, \ldots, x_{I_{|I|}})(\sigma \cup \theta)\\ &\xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(x_1, \ldots, x_n)(\sigma \cup \theta) \equiv \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1}, \ldots, t_{\bar{I}_{|\bar{I}|}})\end{aligned}$$

from $\mathcal{C}ond(\sigma \cup \theta, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. Therefore, it follows from Proposition 23 that we have $\overline{f}_I(s, t_{I_1}, \ldots, t_{I_{|I|}}) \xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1}, \ldots, t_{\bar{I}_{|\bar{I}|}})$.

Next we prove the *if*-part. Suppose that $\overline{f}_I(s, t_{I_1}, \ldots, t_{I_{|I|}}) \xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_{\bar{I}_1}, \ldots, t_{\bar{I}_{|\bar{I}|}})$. Let $\sigma = \{y \mapsto s\}$. Since $y \notin \mathcal{D}om(\theta)$ from the assumption (b), $\sigma\cup\theta$ is a substitution. From the assumption, we also have $\overline{f}_I(y, x_{I_1}, \ldots, x_{I_{|I|}})(\sigma\cup\theta) \xrightarrow[\text{evn},R\text{-in}]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(x_1, \ldots, x_n)(\sigma \cup \theta)$, that is, $\mathcal{C}ond(\sigma \cup \theta, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. Therefore, it follows from Lemma 25 and the fact that $y\sigma \equiv s$ that

$$f(t_1, \ldots, t_n) \equiv f(x_1, \ldots, x_n)(\sigma \cup \theta) \xrightarrow[\text{in}]{*}_R y\sigma \equiv s.$$

□

A.4 On Corollary 10

We can easily prove the *if*-part of Corollary 10 from Theorem 9 and the following proposition which the convergency of $\to_R$ leads.

Proposition 26. *Let R be a convergent constructor TRS over a signature $\mathcal{F}$. Let $(f, I) \in \mathbb{I}_{\mathcal{D}_R}$, $s, s_1, \ldots, s_{|I|}$ be terms in $\mathcal{T}(\mathcal{F},\mathcal{V})$, and $t_1, \ldots, t_{|\bar{I}|}$ be normal forms of R. If $\overline{f}_I(s, s_1, \ldots, s_{|I|}) \xrightarrow{*}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$, then $\overline{f}_I(s, s_1, \ldots, s_{|I|})$ $(\xrightarrow[\text{in}]{*}{}^!_R \cdot \xrightarrow[\text{evn},R\text{-in}]{\varepsilon}_{\mathcal{I}nv(R)})$ $\mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$.*

Proof. We prove this by induction on the level k of rewrite relation $\xrightarrow[(k)]{*}_{\mathcal{I}nv(R)}$. The sequence $\overline{f}_I(s, s_1, \ldots, s_{|I|}) \xrightarrow[(k)]{*}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$ can be represented as follows:

$$\begin{aligned}\overline{f}_I(s, s_1, \ldots, s_{|I|}) &\xrightarrow{*}_R \overline{f}_I(u, u_1, \ldots, u_{|I|})\theta\\ &\xrightarrow[(k)]{}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|})\theta \xrightarrow{*}_R \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})\end{aligned}$$

where we have a rule $\overline{f}_I(u, u_1, \ldots, u_{|I|}) \to \mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|}) \Leftarrow \mathcal{C}ond \in \mathcal{I}nv(R)$, $\mathcal{C}ond(\theta, \xrightarrow[(k')]{*}_{\mathcal{I}nv(R)})$ and $k' \leq k-1$. From Proposition 20 (b), $\mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|}) \in \mathcal{T}(\mathcal{F},\mathcal{V})$ holds. It follows from the convergency of R that there exists a $NF_R(\mathcal{F},\mathcal{V})$-substitution σ such that $\mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|})\sigma \equiv \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|})$. By the induction hypothesis, we can easily shown that $\mathcal{C}ond(\sigma, \xrightarrow[\text{evn},R\text{-in}]{*}_{\mathcal{I}nv(R)})$. It follows from Proposition 20 (a) and the convergency of R that $\overline{f}_I(s, s_1, \ldots, t_{|I|}) \xrightarrow[\text{in}]{*}{}^{!}_R \overline{f}_I(u, u_1, \ldots, u_{|I|})\sigma$. Therefore, we have the following sequence:

$$\begin{aligned}\overline{f}_I(s, s_1, \ldots, s_{|I|}) &\xrightarrow[\text{in}]{*}{}^{!}_R \overline{f}_I(u, u_1, \ldots, u_{|I|})\sigma \\ &\xrightarrow[\text{evn},R\text{-in}]{}{}^{\varepsilon}_{\mathcal{I}nv(R)} \mathsf{tp}_{|\bar{I}|}(w_1, \ldots, w_{|\bar{I}|})\sigma \equiv \mathsf{tp}_{|\bar{I}|}(t_1, \ldots, t_{|\bar{I}|}).\end{aligned}$$

□

B Examples

We show the complete result of the inversion for R_1 in Example 2. It is constructed as shown in Fig. 3.

To compare with a common definition of division, consider the following another definition of multiplication which is slightly different from R_1:

$$R_6 = \begin{cases} \mathsf{add}(0, y) \to y, \\ \mathsf{add}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{add}(x, y)), \\ \mathsf{mult}(0, 0) \to 0, \\ \mathsf{mult}(0, \mathsf{s}(y)) \to 0, \\ \mathsf{mult}(\mathsf{s}(x), 0) \to 0, \\ \mathsf{mult}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{s}(\mathsf{add}(y, \mathsf{mult}(x, \mathsf{s}(y)))). \end{cases}$$

The partial inverse EV-TRS of the above TRS with respect to $\{(\mathsf{mult}, \{2\})\}$ is constructed by $\mathcal{I}nv$ and $\mathbb{U}$ as follows:

$$\mathbb{U}(\mathcal{I}nv(R_6)|_{\{(\mathsf{mult},\{2\})\}}) = \begin{cases} \overline{\mathsf{add}}_{\{1\}}(y, 0) \to \mathsf{tp}_1(y), \\ \overline{\mathsf{add}}_{\{1\}}(\mathsf{s}(z), \mathsf{s}(x)) \to \mathsf{u}_4(\overline{\mathsf{add}}_{\{1\}}(z, x)), \\ \mathsf{u}_4(\mathsf{tp}_1(y)) \to \mathsf{tp}_1(y), \\ \overline{\mathsf{mult}}_{\{2\}}(0, 0) \to \mathsf{tp}_1(0), \\ \overline{\mathsf{mult}}_{\{2\}}(0, \mathsf{s}(y)) \to \mathsf{tp}_1(0), \\ \overline{\mathsf{mult}}_{\{2\}}(0, 0) \to \mathsf{tp}_1(\mathsf{s}(x)), \\ \overline{\mathsf{mult}}_{\{2\}}(\mathsf{s}(z), \mathsf{s}(y)) \to \mathsf{u}_5(\overline{\mathsf{add}}_{\{1\}}(z, y), y), \\ \mathsf{u}_5(\mathsf{tp}_1(w), y) \to \mathsf{u}_6(\overline{\mathsf{mult}}_{\{2\}}(w, \mathsf{s}(y))), \\ \mathsf{u}_6(\mathsf{tp}_1(x)) \to \mathsf{tp}_1(\mathsf{s}(x)), \\ \overline{\mathsf{add}}_{\{1\}}(\mathsf{add}(x, y), x) \to \mathsf{tp}_1(y), \\ \overline{\mathsf{mult}}_{\{2\}}(\mathsf{mult}(x, y), y) \to \mathsf{tp}_1(x). \end{cases}$$

On the other hand, the following TRS is a common definition of the division:

$$R_7 = \begin{cases} \mathsf{minus}(x, 0) \to x, \\ \mathsf{minus}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{minus}(x, y), \\ \mathsf{div}(0, \mathsf{s}(y)) \to x, \\ \mathsf{div}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{s}(\mathsf{div}(\mathsf{minus}(x, y), \mathsf{s}(y))). \end{cases}$$

Comparing the above two systems, $\overline{\mathsf{add}}_{\{1\}}$ and minus, and $\overline{\mathsf{mult}}_{\{2\}}$ and div are the same definitions, respectively, by identifying the redundancy of $\mathbb{U}$ symbols u_4, u_5 and u_6.

$$
\mathcal{I}nv(R_1) = \begin{cases}
\overline{\mathsf{add}}_{\emptyset}(y) \to \mathsf{tp}_2(\mathsf{0}, y), \\
\overline{\mathsf{add}}_{\emptyset}(\mathsf{s}(z)) \to \mathsf{tp}_2(\mathsf{s}(x), y) \Leftarrow \overline{\mathsf{add}}_{\emptyset}(z) \to \mathsf{tp}_2(x, y), \\
\overline{\mathsf{add}}_{\emptyset}(\mathsf{add}(x, y)) \to \mathsf{tp}_2(x, y), \\[1ex]
\overline{\mathsf{mul}}_{\emptyset}(\mathsf{0}) \to \mathsf{tp}_2(\mathsf{0}, y), \\
\overline{\mathsf{mul}}_{\emptyset}(\mathsf{0}) \to \mathsf{tp}_2(x, \mathsf{0}), \\
\overline{\mathsf{mul}}_{\emptyset}(\mathsf{s}(z)) \to \mathsf{tp}_2(\mathsf{s}(x), \mathsf{s}(y)) \\
\qquad\qquad \Leftarrow \overline{\mathsf{add}}_{\emptyset}(z) \to \mathsf{tp}_2(w, y), \overline{\mathsf{mul}}_{\emptyset}(w) \to \mathsf{tp}_2(x, \mathsf{s}(y)), \\
\overline{\mathsf{mul}}_{\emptyset}(\mathsf{mul}(x, y)) \to \mathsf{tp}_2(x, y), \\[1ex]
\overline{\mathsf{add}}_{\{1\}}(y, \mathsf{0}) \to \mathsf{tp}_1(y), \\
\overline{\mathsf{add}}_{\{1\}}(\mathsf{s}(z), \mathsf{s}(x)) \to \mathsf{tp}_1(y) \Leftarrow \overline{\mathsf{add}}_{\{1\}}(z, x) \to \mathsf{tp}_1(y), \\
\overline{\mathsf{add}}_{\{1\}}(\mathsf{add}(x, y), x) \to \mathsf{tp}_1(y), \\[1ex]
\overline{\mathsf{mul}}_{\{1\}}(\mathsf{0}, \mathsf{0}) \to \mathsf{tp}_1(y), \\
\overline{\mathsf{mul}}_{\{1\}}(\mathsf{0}, x) \to \mathsf{tp}_1(\mathsf{0}), \\
\overline{\mathsf{mul}}_{\{1\}}(\mathsf{s}(z), \mathsf{s}(x)) \to \mathsf{tp}_1(\mathsf{s}(y)) \\
\qquad\qquad \Leftarrow \overline{\mathsf{add}}_{\emptyset}(z) \to \mathsf{tp}_2(w, y), \overline{\mathsf{mul}}_{\{1\}}(w, x) \to \mathsf{tp}_1(\mathsf{s}(y)), \\
\overline{\mathsf{mul}}_{\{1\}}(\mathsf{mul}(x, y), x) \to \mathsf{tp}_1(y), \\[1ex]
\overline{\mathsf{add}}_{\{2\}}(y, y) \to \mathsf{tp}_1(\mathsf{0}), \\
\overline{\mathsf{add}}_{\{2\}}(\mathsf{s}(z), y) \to \mathsf{tp}_1(\mathsf{s}(x)) \Leftarrow \overline{\mathsf{add}}_{\{2\}}(z, y) \to \mathsf{tp}_1(x), \\
\overline{\mathsf{add}}_{\{2\}}(\mathsf{add}(x, y), y) \to \mathsf{tp}_1(x), \\[1ex]
\overline{\mathsf{mul}}_{\{2\}}(\mathsf{0}, y) \to \mathsf{tp}_1(\mathsf{0}), \\
\overline{\mathsf{mul}}_{\{2\}}(\mathsf{0}, \mathsf{0}) \to \mathsf{tp}_1(x), \\
\overline{\mathsf{mul}}_{\{2\}}(\mathsf{s}(z), \mathsf{s}(y)) \to \mathsf{tp}_1(\mathsf{s}(x)) \\
\qquad\qquad \Leftarrow \overline{\mathsf{add}}_{\{2\}}(z, y) \to \mathsf{tp}_1(w), \overline{\mathsf{mul}}_{\{2\}}(w, \mathsf{s}(y)) \to \mathsf{tp}_1(x), \\
\overline{\mathsf{mul}}_{\{2\}}(\mathsf{mul}(x, y), y) \to \mathsf{tp}_1(x), \\[1ex]
\overline{\mathsf{add}}_{\{1,2\}}(y, \mathsf{0}, y) \to \mathsf{tp}_0, \\
\overline{\mathsf{add}}_{\{1,2\}}(\mathsf{s}(z), \mathsf{s}(x), y) \to \mathsf{tp}_0 \Leftarrow \mathsf{add}(x, y) \to z, \\
\overline{\mathsf{add}}_{\{1,2\}}(\mathsf{add}(x, y), x, y) \to \mathsf{tp}_0, \\[1ex]
\overline{\mathsf{mul}}_{\{1,2\}}(\mathsf{0}, \mathsf{0}, y) \to \mathsf{tp}_0, \\
\overline{\mathsf{mul}}_{\{1,2\}}(\mathsf{0}, x, \mathsf{0}) \to \mathsf{tp}_0, \\
\overline{\mathsf{mul}}_{\{1,2\}}(\mathsf{s}(z), \mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{tp}_0 \Leftarrow \mathsf{add}(\mathsf{mul}(x, \mathsf{s}(y)), y) \to z, \\
\overline{\mathsf{mul}}_{\{1,2\}}(\mathsf{mul}(x, y), x, y) \to \mathsf{tp}_0, \\[1ex]
\mathsf{add}(\mathsf{0}, y) \to y, \\
\mathsf{add}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{add}(x, y)), \\[1ex]
\mathsf{mul}(\mathsf{0}, y) \to \mathsf{0}, \\
\mathsf{mul}(x, \mathsf{0}) \to \mathsf{0}, \\
\mathsf{mul}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{s}(\mathsf{add}(\mathsf{mul}(x, \mathsf{s}(y)), y)).
\end{cases}
$$

Fig. 3. The partial inverse CTRS of R_1.