Leftmost Outside-In Narrowing Calculi

Tetsuo Ida [†] [‡] and Koichi Nakahara [††]

March, 1994

ISE–TR–94–107

Abstract

We present narrowing calculi that are used for implementing functional-logic programming languages. The narrowing calculi are based on the notion of the leftmost outside-in reduction of Huet and Lévy. We note the correspondence between the narrowing and reduction derivations, and define the leftmost outside-in narrowing derivation. We then give a narrowing calculus OINC that generates the leftmost outside-in narrowing derivations. It consists of several inference rules that perform the leftmost outside-in narrowing. We prove the completeness of OINC using an ordering defined over a narrowing derivation space. In order to use the calculus OINC as a model of computation of functional-logic programming we extend OINC to incorporate strict equality. The extension results in a new narrowing calculus s-OINC. We show also that s-OINC enjoys the same completeness property as OINC.

[†]Inst. of Information Sciences and Electronics, Univ. of Tsukuba, Tsukuba, Ibaraki 305, Japan

[‡]ida@is.tsukuba.ac.jp

[††]koh@softlab.is.tsukuba.ac.jp

1 Introduction

Recently narrowing has received considerable research interest in declarative programming community as it was found to be an important computing mechanism of functional-logic programming languages[2, 5, 6, 7, 11, 14, 16]. In this paper we present narrowing calculi that are used for implementing functional-logic programming languages. Our narrowing calculi are based on the notion of the outside-in reduction of Huet and Lévy[9] for orthogonal term rewriting systems. Huet and Lévy showed that for a given reduction derivation of a term s to its normal form (if it exists) there exists a leftmost outside-in reduction derivation from s to its normal form. This derivation is also called standard due to this property.

To generate a standard reduction derivation from a term s to its normal form is impossible in general without look-ahead. Practical implications of the standard derivation lie in the following facts. First, for a sub-class of orthogonal term rewriting systems called strongly sequential systems, there exists a strategy, i.e., effective means to locate redexes that should be reduced next without look-ahead, by which we generate a standard reduction derivation. This strategy is often called a call-by-need strategy since it selects a redex only when its contraction is definitely needed in each reduction. Secondly, it provides a theoretical basis of the lazy evaluation in the framework of (first-order) functional programming.

By the correspondence of reduction and narrowing, in particular by the use of a so-called lifting lemma, we can obtain a narrowing derivation that corresponds to the standard reduction derivation. This narrowing derivation, which we call a leftmost outside-in narrowing derivation, deserves a special investigation, as the leftmost outside-in reduction derivation does in reduction. Let a method of narrowing that generates the leftmost outside-in narrowing derivation be called leftmost outside-in narrowing. The leftmost outside-in narrowing behaves very much like the leftmost outside-in reduction. It narrows the subterms at the same positions that are contracted by the reduction using the same rewrite rules. It performs 'lazy narrowing'. Furthermore to process narrowable expressions in outside-in manner is amenable to implementation of narrowing.

There exists an important difference between reduction and narrowing derivations, however. A narrowable term is not stable under contextual narrowing. That is, descendants of a narrowable term may become non-narrowable after its superterm is narrowed, whereas in an orthogonal system descendants of a redex remain to be a redex by the reduction of its superterm. Hence for a given standard reduction derivation its lifted narrowing derivation is not necessarily the one in which only 'needed' narrowable terms are contracted. This phenomenon was observed by several researchers and lead them to discover new methods of narrowing. You presented an outer narrowing for constructor-based orthogonal systems[18] and Antoy et al. presented needed narrowing strategy for strongly sequential constructor-based systems[1]. Darlington and Guo noted the similarity between reduction and narrowing derivations, and developed a narrowing algorithm for constructor-based orthogonal term rewriting systems [4]. Their algorithm

is essentially the same as our leftmost outside-in narrowing restricted to constructor-based systems.

The leftmost outside-in narrowing is similar in its behaviour to Antoy's needed narrowing, but it differs in that we consider 'needed-ness' of a narrowable term with respect to previous context in which its ancestors are defined. Furthermore we present the leftmost outside-in narrowing as a computation of a calculus consisting of several inference rules that altogether perform the leftmost outside-in narrowing. Our narrowing calculi are more general than the needed narrowing or the outer narrowing in that they are defined for arbitrary orthogonal term rewriting systems. For constructor-based systems our calculi generate the same narrowing derivations as the ones by outer and needed narrowing.

The organization of the paper is as follows. We first explain narrowing as a computation of an inference system NC that stands for Narrowing Calculus. From the calculus NC we develop another narrowing calculi that perform leftmost outside-in narrowing. In Section 4 we give the inductive definitions of a leftmost outside-in narrowing derivation. In Section 5 we present a calculus called OINC that generates a leftmost outside-in derivation. The calculus OINC enjoys the soundness and completeness. In Section 6 we prove the completeness of OINC, employing a lifting argument in which we relate a standard reduction derivation and a leftmost outside-in narrowing derivation. In Section 7 we incorporate strict equality into OINC. We show that this extended calculus, to be called s-OINC, is a natural and efficient model of computation for functional-logic programming, and further that s-OINC is complete.

Because of the lack of space all proofs of the propositions are omitted.

2 Preliminaries

Let $\mathcal{F}$ be a set of function symbols, and $\mathcal{V}$ a set of variables, satisfying $\mathcal{F} \cap \mathcal{V} = \emptyset$. Terms are defined as usual over a set of alphabet $\mathcal{F} \cup \mathcal{V}$. The set of terms is denoted by $\mathcal{T}(\mathcal{F}, \mathcal{V})$, or simply by $\mathcal{T}$. A term t is called linear when no variable occurs in t more than once. A set $\mathcal{F}$ is divided into disjoints sets $\mathcal{F}_\mathcal{C}$ and $\mathcal{F}_\mathcal{D}$; $\mathcal{F}_\mathcal{C}$ is a set of constructors and $\mathcal{F}_\mathcal{D}$ is a set of defined function symbols. When there is no danger of confusion we call a constructor symbol simply a constructor and a defined function symbol simply a function symbol. A term whose root symbol is a constructor is called a constructor term, and a term in $\mathcal{T}(\mathcal{F}_\mathcal{C}, \mathcal{V})$ is called a data term.

$\mathcal{V}(\mathcal{A})$ denotes a set of variables occurring in a syntactic object $\mathcal{A}$. The set $\mathcal{O}(t)$ of positions of a term t is a set of sequences of positive integers that address subterms of t. An empty sequence is denoted by ε. A position u in $\mathcal{O}(t)$ addresses a subterm $t|_u$, where $t|_u$ is defined as follows. Let $t \triangleq f(t_1, \ldots, t_n)$. Then, $t|_u \triangleq t$ if $u = \varepsilon$ and $t|_u \triangleq t_i|_{u'}$ if $u = i.u'$. A position u in $\mathcal{O}(t)$ is called a non-variable position if $t|_u$ is not a variable. The set of non-variable positions of t is denoted by $\overline{\mathcal{O}}(t)$. A term obtained from t by replacing $t|_u$, where $u \in \mathcal{O}(t)$, by a term s is denoted by $t[s]_u$. An equation $s = t$, where $s = t \in \mathcal{T}$, is a special term whose root symbol

is $=$ (used as an infix operator, and allowed only at the root position).

A substitution is a mapping from $\mathcal{V}$ to $\mathcal{T}$. The domain of a substitution θ is defined as $\mathcal{D}\theta = \{x \mid \theta x \not\equiv x, x \in \mathcal{V}\}$, and the codomain of θ as $Cod\ \theta = \{\theta x \mid x \in \mathcal{D}\theta\}$. We identify a substitution θ with the set $\{x \mapsto \theta x \mid x \in \mathcal{D}\theta\}$. An empty substitution is defined as the empty set $\emptyset$. A substitution is extended to an endomorphism over $\mathcal{T}$ as usual. Let $V \subseteq \mathcal{V}$. The restriction of θ to V is denoted by $\theta\restriction_V$. We write $\theta_1 = \theta_2[V]$ when $\theta_1\restriction_V = \theta_2\restriction_V$ holds. The composition of θ_2 and θ_1, (first apply θ_1, then θ_2) is denoted by $\theta_2\theta_1$. When $\sigma\theta_1 = \theta_2$ for some substitution σ, we write $\theta_1 \preceq \theta_2$. When $\sigma\theta_1 = \theta_2[V]$ holds for some substitution σ, we write $\theta_1 \preceq \theta_2[V]$.

3 Calculus NC

Narrowing is a combination of instantiating an equation $s = t$ by applying a most general substitution θ and of subsequent rewriting of the equation $\theta s = \theta t$ by a rewrite rule to form a new equation $s_1 = t_1$. Narrowing is successively applied, to obtain an equation $s_n = t_n$ both sides of which are unifiable. The whole process of rewriting an equation $s = t (\triangleq s_0 = t_0)$ to $s_n = t_n$ is also called narrowing.

3.1 NC over equations

For our purpose, narrowing is best presented in the form of a calculus $(\mathcal{G}, \mathcal{L})$, where $\mathcal{G}$ is a set of objects manipulated in this calculus, and $\mathcal{L}$ is a set of inference rules that operate on $\mathcal{G}$. We first present a calculus that operates equations.

Definition 3.1 (NC over equations) Let $\mathcal{R}$ be an arbitrary term rewriting system. A calculus NC for $\mathcal{R}$ is a pair $(\mathcal{G}, NC)$, where $\mathcal{G}$ and NC are following.

- $\mathcal{G}$ is a set of equations and a special term 'true'.
- NC consists of the following inference rules.
  - [n] narrowing

$$\frac{s = t}{\theta s[\sigma r]_u = \theta t} \quad \text{or} \quad \frac{t = s}{\theta t = \theta s[\sigma r]_u}$$

    if there exist
    - a new variant $l \to r$ of a rewrite rule in $\mathcal{R}$,
    - $u \in \overline{\mathcal{O}}(s)$,
    - a substitution $\theta \cup \sigma$ such that
      - $\mathcal{D}\theta \subseteq \mathcal{V}(s|_u)$,
      - $\mathcal{D}\sigma \subseteq \mathcal{V}(l)$,
      - $(\theta s)|_u \equiv \sigma l$,

· θ is a most general substitution that satisfies the above conditions.

– [f] reflection

$$\frac{s = t}{\text{true}} \qquad \text{if } \theta s \equiv \theta t \text{ for a most general unifier } \theta.$$

The inference rule [n] states explicitly the involved manipulation of equations; a most general substitution is applied to s, and then the subterm at u, i.e., $(\theta s)|_u$ is identified such that the left-hand side of the rule $l \to r$ is matched against $(\theta s)|_u$ using the substitution σ. $\mathcal{D}\theta$ contains only variables in the equation to be narrowed.

In the above we say that the equation $s = t (t = s)$ is narrowed at the position $1.u(2.u)$. A term $s|_u$ is called narrowable expression (narex in short). A term is called narrowable if it has a narex as a subterm. A term that has no narex is called non-narrowable. Furthermore, a substitution θ is called non-narrowable if $Cod\ \theta$ does not contain narrowable terms.

Let e and e' be equations. We write $e \overset{\text{n}}{\rightsquigarrow}_\theta e'$ if e' is obtained from e by a single application of the inference rule [n], and likewise for $e \overset{\text{f}}{\rightsquigarrow}_\theta e'$. The substitution θ in both cases are those formed in the inference step and whose domain is restricted to the set of the variables occurring in the equation e. We also write $e \overset{\text{n}}{\rightsquigarrow}_\theta e'$ if e' is obtained from e by zero or more applications of the inference rule [n]. The substitution θ is a composition of the substitutions formed in the $\overset{\text{n}}{\rightsquigarrow}$ steps, and whose domain is restricted to $\mathcal{V}(e)$. Abusing the notation, we also let $NC = \{[\text{n}], [\text{f}]\}$, and write $e \overset{NC}{\rightsquigarrow}_\theta e'$ when e' is obtained from e by single application of the inference rule in NC and likewise for $e \overset{NC}{\rightsquigarrow}_\theta e'$. A sequence $e_0 \overset{\text{n}}{\rightsquigarrow}_{\theta_1} e_1 \overset{\text{n}}{\rightsquigarrow} \cdots \overset{NC}{\rightsquigarrow}_{\theta_n} e_n$ is called an NC-derivation. The NC-derivation whose last step of the derivation is $\overset{\text{f}}{\rightsquigarrow}$, i.e.,

$$e_0 \overset{\text{n}}{\rightsquigarrow}_{\theta_1} e_1 \overset{\text{n}}{\rightsquigarrow} \cdots \overset{\text{n}}{\rightsquigarrow}_{\theta_n} e_n \overset{\text{f}}{\rightsquigarrow}_{\theta_{n+1}} \text{true}$$

is said to be successful. A successful NC-derivation starting from e_0 gives a solution $(\theta_{n+1}\theta_n \cdots \theta_1)\restriction_{\mathcal{V}(e_0)}$ of the equation e_0, and we write $e_0 \overset{NC}{\rightsquigarrow}_\theta \text{true}$.

Example 3.1 Let $\mathcal{R}$ be a TRS given by:

$$\mathcal{R} = \begin{cases} f(w) \to w, \\ g(1) \to 1. \end{cases}$$

Given an equation $f(g(x)) = f(y)$, there are 13 successful NC-derivations. We give below the following two successful NC-derivations.

$$f(g(x)) = f(y) \overset{\text{f}}{\rightsquigarrow}_{\{y \mapsto g(x)\}} \text{true}, \tag{1}$$

$$f(g(x)) = f(y) \overset{\text{n}}{\rightsquigarrow}_\emptyset g(x) = f(y) \overset{\text{n}}{\rightsquigarrow}_{\{x \mapsto 1\}} 1 = f(y) \overset{\text{n}}{\rightsquigarrow}_\emptyset 1 = y \overset{\text{f}}{\rightsquigarrow}_{\{y \mapsto 1\}} \text{true}. \tag{2}$$

The NC-derivations (1) and (2) yield solutions $\{y \mapsto g(x)\}$ and $\{x \mapsto 1, y \mapsto 1\}$, respectively. The former solution contains a term that is still narrowable, whereas the latter solution contains only normal forms. From the viewpoint of equation solving, the former solution would be satisfactory, but from the viewpoint of programming the latter solution $\{x \mapsto 1, y \mapsto 1\}$ is desirable. We will take the view point of programming languages and consider the latter solution as our solution. In order to guarantee that solutions are normal forms, we will later introduce strict equality together with certain syntactic restrictions on TRSs.

3.2 Correspondence between narrowing and reduction derivations

By the construction of narrowing, we have a correspondence between the NC-derivation

$$e_0 \overset{n}{\leadsto}_{\theta_1} e_1 \overset{n}{\leadsto} \cdots \overset{n}{\leadsto} e_{n-1} \overset{n}{\leadsto}_{\theta_n} e_n$$

and the reduction derivation

$$\sigma_0 e_0 \to_{\mathcal{R}} \sigma_1 e_1 \to\!\!\!\!\to_{\mathcal{R}} \sigma_{n-1} e_{n-1} \to_{\mathcal{R}} e_n$$

where $\sigma_i = \theta_n \cdots \theta_{i+1}$ for $i = 0, \ldots, n-1$,

in which the same rewrite rules are employed at the same positions in each step of both derivations.

The last step $\overset{f}{\leadsto}$ of a successful NC-derivation corresponds to the reduction in which a rewrite rule $x = x \to \text{true}$ is used. Let $\mathcal{R}_+$ denote a TRS extended with the rewrite rule $x = x \to \text{true}$. Then, the whole successful NC-derivation $e_0 \overset{n}{\leadsto}_{\theta}$ true can be made to correspond to the reduction derivation $\theta e_0 \to\!\!\!\!\to_{\mathcal{R}_+}$ true. Whenever we say an NC-derivation corresponds to a reduction derivation (and vice versa), we implicitly assume that the same rewrite rules are employed at the same positions in each step of both derivations.

3.3 NC over goals

We next extend NC to deal with sequences of equations. A (possibly empty) sequence of equations and true's is called a goal. An empty sequence is denoted by $\Box$. Goals are objects that are manipulated by the calculus NC (over goals). In this paper we are primarily interested in the narrowing that can be a computing mechanism of functional-logic programs. By this we mean the following:

1. We restrict ourselves to orthogonal term rewriting systems (abbreviated as OTRS, hereafter).
2. We regard data terms as an answer of the evaluation of functional-logic programs.

The first restriction would be justified since most of (first-order) functional programs are regarded as an OTRS. The second restriction seems natural since data terms are objects which we represent our data with. The first point affects almost all of our discussion in this paper, whereas the second point does not until Section 7 where we discuss strict equality.

Definition 3.2 (NC over goals) Let $\mathcal{R}$ be an OTRS. A calculus NC for $\mathcal{R}$ is a pair $(\mathcal{G}, NC)$, where $\mathcal{G}$ and NC are following.

- $\mathcal{G}$ is a set of goals.
- NC is a set of inference rules defined as follows:
  - [n] narrowing

$$\frac{E, s = t, E'}{\tau E, \tau s[r]_u = \tau t, \tau E'} \quad \text{or} \quad \frac{E, t = s, E'}{\tau E, \tau t = \tau s[r]_u, \tau E'}$$

    if there exist
    - a new variant $l \to r$ of a rewrite rule in $\mathcal{R}$,
    - $u \in \overline{\mathcal{O}}(s)$,
    - a most general unifier τ such that $(\tau s)|_u \equiv \tau l$.
  - [f] reflection

$$\frac{E, s = t, E'}{\theta E, \text{true}, \theta E'}$$ if s and t are unifiable with a most general unifier θ.

Notations:

- The same symbol **NC** is used for the narrowing calculus over goals, and from now on **NC** is a calculus over goals.
- The symbol $\top$ is used to represent generically a sequence of zero or more true's.

In Definition 3.1 we gave the substitutions θ and σ in defining the inference rule [n]. We let $\tau \triangleq \theta \cup \sigma$. Then it is clear that the narrowing rule [n] can be stated as in the above. The relations over a set of goals $\overset{NC}{\rightsquigarrow}$, $\overset{NC}{\rightsquigarrow}$, $\overset{\text{n}}{\rightsquigarrow}$, $\overset{\text{f}}{\rightsquigarrow}$ etc. are defined similarly to the corresponding ones over a set of equations. Let A be an NC-derivation $A : G \overset{NC}{\rightsquigarrow} G'$. The goal G is called an initial goal of the NC-derivation A. If $\overset{NC}{\rightsquigarrow}$ is empty in the above NC-derivation, the derivation is empty. The empty NC-derivation is denoted by 0.

The correspondence between narrowing and reduction derivations can be extended in an obvious way to the derivations over goals. We should note, however, that the induced reduction relation $\to$ is with respect to $\mathcal{R}_+$ not to $\mathcal{R}$.

4 Leftmost outside-in narrowing

The calculus **NC** is too general as a computation model in that it does not incorporate a method by which we can locate a narex. Selection of a narex could be specified by a computation rule

which is often called a strategy. A computation rule could specify, for example, that the outermost-leftmost narexes be processed first among the narexes. In this paper we are aiming at a calculus in which this kind of a computation rule is built in. Towards that goal, we first define a special class of successful NC-derivations called leftmost outside-in derivations.

4.1 A space of successful NC-derivations

Huet and Lévy defined the leftmost outside-in reduction derivation inductively on the length of derivations using a set of external positions. The set of external positions are determined with respect to the derivations. In this paper we give an alternative (but equivalent) definition of the leftmost outside-in reduction and narrowing derivations inductively on the complexity measure of derivations. We first define a space of successful derivations over which total ordering of derivations is defined.

Let $(D, \lhd)$ be an ordered set, where D is a set of successful NC-derivations and $\lhd$ is a (total) ordering over the set D defined as follows.

Definition 4.1 Let G be a goal of **NC** and $A : G \overset{NC}{\rightsquigarrow}_\theta \top \in D$. The complexity $\|A\|$ of A is defined as a triple $\langle \#\mathrm{n}, \mathcal{D}\theta, \#G \rangle$, where

- $\#\mathrm{n}$ is the number of applications of the inference rule [n] in the derivation A,
- $\#G$ is the number of occurrences of variables and function symbols in G excluding the symbols $=$ and true.

Definition 4.2

- The ordering $\prec\!\!\!\prec$ on complexities of a successful NC-derivation is the lexicographic ordering of $<$ on natural numbers, (proper) set inclusion $\subset$, and the ordering $<$ on natural numbers.
- The ordering $\lhd$ on D is defined as follows:
  Let $A, A' \in D$. $A' \lhd A$ if $\|A'\| \prec\!\!\!\prec \|A\|$.

Note that the ordering $\lhd$ is well-founded, and hence $(D, \lhd)$ is a well-founded set.

Suppose that we are given a successful NC-derivation $A : G \overset{NC}{\rightsquigarrow}_\theta \top$. We try to find another successful NC-derivation starting from some goal G' that yields the solution θ' such that $\theta' \preceq \theta$ and $A' \lhd A$. If such a derivation A' exists, and we know a method of transforming A to A', we then reason about A' instead of A. Since $(D, \lhd)$ is a well-founded set, this process will terminate. Indeed, there are special pairs of successful NC-derivations related by the relations that are subsets of $\lhd$. The following lemmas (Lemmas 4.1, 4.2, 4.4 and 4.5) enable us to enumerate those pairs of successful NC-derivations.

The leftmost outside-in NC-derivation is defined inductively on the relation $\lhd$.

Lemma 4.1 Let

$$A : \top, s = x, E \overset{\mathrm{f}}{\rightsquigarrow}_{\eta(=\{x \mapsto s\})} \top, \eta E \overset{NC}{\rightsquigarrow}_{\theta} \top, \text{ where } x \notin \mathcal{V}(s)$$

be an NC-derivation and

$$A' : \eta E \overset{NC}{\rightsquigarrow}_{\theta} \top$$

be the NC-derivation taken from the sub-derivation $\top, \eta E \overset{NC}{\rightsquigarrow}_{\theta} \top$. Then $A' \lhd A$.

Proof: Since $x \notin \mathcal{V}(\eta E)$, we have $x \notin \mathcal{D}\theta$, and hence $\mathcal{D}\theta \subset \mathcal{D}(\theta\eta)$. From this $A' \lhd A$ follows.∎

We write $A' \overset{\mathrm{v1}}{\lhd} A$, if A and A' are NC-derivations given above.

Lemma 4.2 Let

$$A : \top, x = t, E \overset{\mathrm{f}}{\rightsquigarrow}_{\eta(=\{x \mapsto t\})} \top, \eta E \overset{NC}{\rightsquigarrow}_{\theta} \top, \text{ where } x \notin \mathcal{V}(t)$$

be an NC-derivation and

$$A' : \eta E \overset{NC}{\rightsquigarrow}_{\theta} \top$$

be the NC-derivation taken from the sub-derivation $\top, \eta E \overset{NC}{\rightsquigarrow}_{\theta} \top$. Then $A' \lhd A$.

Proof: Similarly to Lemma 4.1.∎

We write $A' \overset{\mathrm{v2}}{\lhd} A$, if A and A' are NC-derivations given above.

In the following lemmas (Lemmas 4.4 and 4.5) a preference is given to the left-hand sides of the equations. This is made possible because of the syntactic restrictions on the goals that we will impose in Section 4.2. The following definition of descendant is used in the following lemmas.

Definition 4.3 Let $G(\triangleq e_1, \ldots, e_n) \overset{NC}{\rightsquigarrow} G'(\triangleq e'_1, \ldots, e'_n)$ be an NC-derivation.

- The term $e'_i, i \in \{1, \ldots, n\}$ is called an immediate descendant, written as $e_i \hookrightarrow e'_i$.
- A term e' is a descendant of e if $e \hookrightarrow^* e'$, where $\hookrightarrow^*$ is the reflexive and transitive closure of $\hookrightarrow$.

To prove Lemmas 4.4 and 4.5, we need the following lemma.

Lemma 4.3 Let $\mathcal{R}$ be a TRS and G be a goal $s_1 = t_1, \ldots, s_n = t_n$ such that $\mathcal{V}(s_i) \cap \mathcal{V}(t_j) = \emptyset$ for any $i, j \in \{1, \ldots, n\}$ and $\mathcal{V}(t_i) \cap \mathcal{V}(t_j) = \emptyset$ for any $i \neq j$. If there exists a successful NC-derivation

$$G \overset{\mathrm{n}}{\rightsquigarrow} G' \overset{\mathrm{f}}{\rightsquigarrow} \top, \tag{3}$$

then there exists a successful NC-derivation

$$G \overset{\mathrm{n}}{\rightsquigarrow} \overset{\mathrm{f}}{\rightsquigarrow} G_1 \ldots \overset{\mathrm{n}}{\rightsquigarrow} \overset{\mathrm{f}}{\rightsquigarrow} G_{n-1} \overset{\mathrm{n}}{\rightsquigarrow} \overset{\mathrm{f}}{\rightsquigarrow} \top, \tag{4}$$

such that

- $\overset{\mathrm{f}}{\leadsto}$-steps on each equation are moved as leftward as possible in the derivation (4), and
- in each corresponding $\overset{\mathrm{n}}{\leadsto}$-steps the same rewrite rules are employed at the same positions.

Proof: By repeated applications of Switching Lemma A.1 given in the appendix.

We call derivation (4) the $\overset{\mathrm{f}}{\leadsto}$-eager derivation of derivation (3).

Lemma 4.4 Let $\mathcal{R}$ be a TRS and G be a goal $\top, f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n), E$ such that $\mathcal{V}(f(s_1,\ldots,s_n)) \cap \mathcal{V}(f(t_1,\ldots,t_n)) = \emptyset$, and $f(t_1,\ldots,t_n)$ is linear. If there exists an NC-derivation

$$A : G \overset{\mathrm{n}}{\leadsto}_{\theta_1} \top, f(s'_1,\ldots,s'_n) = f(t_1,\ldots,t_n), \theta_1 E \overset{\mathrm{f}}{\leadsto}_{\theta_2} \top, \theta_2\theta_1 E \tag{5}$$

$$\overset{NC}{\leadsto}_{\theta_3} \top, \tag{6}$$

where the descendants of $f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n)$ are not narrowed at position 1 then there exists an NC-derivation

$$A' : s_1 = t_1,\ldots,s_n = t_n, E \overset{NC}{\leadsto}_{\eta} \top, \eta E \tag{7}$$

$$\overset{NC}{\leadsto}_{\theta'_3} \top, \tag{8}$$

such that

- the derivation $s_1 = t_1,\ldots,s_n = t_n \overset{NC}{\leadsto}_{\eta} \top$ extracted from the subderivation (7) is the $\overset{\mathrm{f}}{\leadsto}$-eager derivation of $s_1 = t_n,\ldots,s_n = t_n \overset{\mathrm{n}}{\leadsto} s'_1 = t_n,\ldots,s'_n = t_n \overset{\mathrm{f}}{\leadsto} \top$, and
- in each step of the subderivation (8), the same rewrite rules are employed at the same position of the same equation in each goal of both subderivations (6) and (8).

Furthermore, we have $A' \lhd A$.

Proof: Note that $\eta = \theta_2\theta_1$ modulo renaming $[\mathcal{V}(G)]$ by Lemma 4.3. From the NC-derivation A, we can construct an NC-derivation

$$A'' : s_1 = t_1,\ldots,s_n = t_n, E \overset{\mathrm{n}}{\leadsto}_{\theta_1} s'_1 = t_1,\ldots,s'_n = t_n, \theta_1 E \overset{\mathrm{f}}{\leadsto}_{\theta_2} \theta_2\theta_1 E \overset{NC}{\leadsto}_{\theta_3} \top.$$

By the assumption on the goal G and by the repeated applications of Lemma 4.3, we have the correspondence between the deivations A and A'. It is straightfoward to show $A' \lhd A$. ∎

We write $A' \overset{\mathrm{d}}{\lhd} A$, if A and A' are NC-derivations given above.

Lemma 4.5 If there exists an NC-derivation

$$A : (G \triangleq)\top, f(s_1,\ldots,s_n) = t, E \overset{\mathrm{n}}{\rightsquigarrow}_{\theta_1} \top, f(s'_1,\ldots,s'_n) = t, \theta_1 E \tag{9}$$

$$\overset{\mathrm{n}}{\rightsquigarrow}_{\sigma} \top, \sigma r = t, \sigma\theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_2} \top, \tag{10}$$

in which the descendant of $f(s_1,\ldots,s_n) = t$ is narrowed for the first time at the position 1, in some step in A using a rewrite rule $f(l_1,\ldots,l_n) \to r$, then there exists an NC-derivation

$$A' : s_1 = l_1,\ldots,s_n = l_n, r = t, E \overset{NC}{\rightsquigarrow}_{\eta} \top, \eta r = t, \sigma'\theta_1 E, \tag{11}$$

$$\overset{NC}{\rightsquigarrow}_{\theta'_2} \top, \tag{12}$$

such that

- the derivation $s_1 = l_1,\ldots,s_n = l_n \overset{NC}{\rightsquigarrow}_{\eta} \top$ extracted from the subderivation (11) is the $\overset{\mathrm{f}}{\rightsquigarrow}$-eager derivation of $s_1 = l_n,\ldots,s_n = l_n \overset{\mathrm{n}}{\rightsquigarrow} s'_1 = l_n,\ldots,s'_n = l_n \overset{\mathrm{f}}{\rightsquigarrow} \top$, and
- in each step of the subderivation (12), the same rewrite rules are employed at the same position of the same equation in each goal of both subderivations (10) and (12).

Furthermore, we have $A' \lhd A$.

Proof: The existence of the derivation A' is assured by Lemma 4.3. The number of $\overset{\mathrm{n}}{\rightsquigarrow}$-steps in A' is one less than that in A. Hence, $A' \lhd A$.∎

We write $A' \overset{\mathrm{on}}{\lhd} A$, if A and A' are NC-derivations given above. Let $\overset{OI}{\lhd} = \lhd \cup \overset{\mathrm{v1}}{\lhd} \cup \overset{\mathrm{v2}}{\lhd} \cup \overset{\mathrm{d}}{\lhd} \cup \overset{\mathrm{on}}{\lhd}$, and $\overset{OI*}{\lhd}$ denote the reflexive and transitive closure of $\overset{OI}{\lhd}$. Using the above lemmas we can transform A to A' that is $A' \overset{OI*}{\lhd} A$.

4.2 Initial NC-derivation

We next discuss what class of successful NC-derivations we will transform with the relation $\overset{OI}{\lhd}$. Since we are interested in narrowing that can be used in functional-logic programming, we restrict ourselves to a class of NC-derivations, to be called initial NC-derivations. We first give necessary definitions.

Definition 4.4 A goal G is called right-normal if

- G is $\Box$, or
- G is a sequence consisting only of equations and the right-hand side of all the equations in G is a ground normal form.

The restriction of right-normality on goals is slightly more general than what we actually need in functional-logic programming. We are interested in solving a strict equation $s \equiv t =$ true. The idea of using a strict equation in functional-logic programming languages originates in a logic plus functional language K-LEAF[6] and has been exploited by several researchers[15, 12, 3, 1]. Solving a strict equation $s \equiv t =$ true is to find a normalized substitution θ such that θs and θt have a common reduct that is a data term. To cope with the strict equations, a TRS $\mathcal{R}$ is extended with rewrite rules for strict equality

$$\mathcal{R}_c = \begin{cases} \{c \equiv c \to \text{true}\} & \text{if the arity of } c \text{ is } 0 \\ \{c(x_1, \ldots, x_n) \equiv c(y_1, \ldots, y_n) \to x_1 \equiv y_1 \wedge \ldots \wedge x_n \equiv y_n\} & \text{if the arity of } c \text{ is } n > 0, \end{cases}$$

together with the rewrite rule $\text{true} \wedge x \to x$. Let $\mathcal{R}^{\equiv} = \mathcal{R} \cup \bigcup_{c \in \mathcal{F}_C} \mathcal{R}_c \cup \{\text{true} \wedge x \to x\}$. Furthermore, we abbreviate $s \equiv t =$ true as $s \equiv t$ in a goal.

The choice of right-normal initial goals as our objects of narrowing leads to a class of goals called proper goals.

Definition 4.5

- A successful NC-derivation $G \overset{NC}{\rightsquigarrow} \top$ is called initial if G is right-normal.
- Let $D_0 (\subseteq D)$ be a set of initial NC-derivations, and

$$\mathcal{G}_0 = \{G \mid G \text{ is an initial goal of } A' \text{ such that } A' \overset{OI*}{\lhd} A, A \subseteq D_0\}.$$

  An element of $\mathcal{G}_0$ is called a proper goal.

A proper goal has the following properties.

Definition 4.6 Let G be a goal $E, s = t, E'$, and $\text{Left}(G, t) = \{E, s\}$.

- The equation $s = t$ in G is left-independent if $\mathcal{V}(\text{Left}(G, t)) \cap \mathcal{V}(t) = \emptyset$.
- A goal G is called left-independent if all the equations in G are left-independent.

By the definition of proper goals, we can easily see that the following proposition holds.

Proposition 4.1 Let $\mathcal{R}$ be an OTRS. A proper goal G satisfies the following properties.

(G1) G is left-independent.

(G2) The right-hand side of every equation in G is linear and non-narrowable.

The following lemma on the solutions of initial NC-derivations will be used later.

Lemma 4.6 Let $\mathcal{R}$ be an OTRS, and G be a right-normal goal. If $G \overset{NC}{\rightsquigarrow}_\theta \top$ then the solution θ is non-narrowable.

4.3 Leftmost outside-in NC-derivations

We are now ready to define a leftmost outside-in NC-derivation of a proper goal.

Definition 4.7 Let $\mathcal{R}$ be an OTRS, and G be a proper goal. An leftmost outside-in (abbreviated as LOI, hereafter) NC-derivation $G \overset{NC}{\rightsquigarrow} \top$ is defined inductively (with respect to $\lhd$) as follows.

- An empty NC-derivation 0 is LOI.
- $A : \top, s = x, E \overset{NC}{\rightsquigarrow}_\theta \top$ is LOI if
  - A is written as $\top, s = x, E \overset{\mathrm{f}}{\rightsquigarrow}_{\theta_1(=\{x \mapsto s\})} \top, \theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_2} \top$, and
  - A', such that $A' \overset{\mathrm{v1}}{\lhd} A$, is LOI.
- $A : \top, x = t, E \overset{NC}{\rightsquigarrow}_\theta \top$ where $t \notin \mathcal{V}$ is LOI if
  - A is written as $\top, x = t, E \overset{\mathrm{f}}{\rightsquigarrow}_{\theta_1(=\{x \mapsto t\})} \top, \theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_2} \top$ and
  - A', such that $A' \overset{\mathrm{v2}}{\lhd} A$, is LOI.
- A: $\top, f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n), E \overset{NC}{\rightsquigarrow}_\theta \top$ is LOI if
  - A is written as
    $$\top, f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n), E$$
    $$\overset{\mathrm{n}}{\rightsquigarrow}_{\theta_1} f(s'_1, \ldots, s'_n) = f(t_1, \ldots, t_n), \theta_1 E \overset{\mathrm{f}}{\rightsquigarrow}_{\theta_2} \theta_2\theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_3} \top, \text{ and}$$
  - A', such that $A' \overset{\mathrm{d}}{\lhd} A$, is LOI.
- $A : \top, f(s_1, \ldots, s_n) = t, E \overset{NC}{\rightsquigarrow}_\theta \top$, where $t \notin \mathcal{V}$, is LOI if
  - A is written as
    $$\top, f(s_1, \ldots, s_n) = t, E$$
    $$\overset{\mathrm{n}}{\rightsquigarrow}_{\theta_1} f(s'_1, \ldots, s'_n) = t, \theta_1 E \overset{\mathrm{n}}{\rightsquigarrow}_\sigma \top, \sigma r = t, \sigma\theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_2} \top, \text{ and}$$
  - A', such that $A' \overset{\mathrm{on}}{\lhd} A$, is LOI.

We next define an LOI reduction derivation via an LOI NC-narrowing derivation. A reduction derivation

$$s(\triangleq s_0) \to s_1 \twoheadrightarrow \hat{s}(\triangleq s_n), \text{ where } \hat{s} \text{ is a normal form of a term } s, \tag{13}$$

is LOI if the corresponding NC-derivation A constructed in the following way is LOI. Let $\{x_1, \ldots, x_k\} = \mathcal{V}(s)$, $c_1, \ldots, c_k$ be distinct fresh constants, and $\sigma = \{x_1 \mapsto c_1, \ldots, x_k \mapsto c_k\}$.

$$A : \sigma s_0 = \sigma\hat{s} \overset{\mathrm{n}}{\rightsquigarrow}_\emptyset \sigma s_1 = \sigma\hat{s} \overset{\mathrm{n}}{\rightsquigarrow}_\emptyset \sigma s_n = \sigma\hat{s} \overset{\mathrm{f}}{\rightsquigarrow}_\emptyset \top. \tag{14}$$

The reduction derivation (13) is called LOI if the NC-derivation (14) is LOI.

The LOI reduction derivation starting from a term s is very important for the following reasons.

(i) It represents the class of the reduction derivations starting from s.

(ii) The LOI reduction derivation from s to its normal form is optimal in the sense that it contracts only the redexes that are needed to get to the normal form.

For this reason Huet and Lévy called an LOI reduction derivation standard [9]. The clause (i) in the above is formally stated as follows.

Theorem 4.1 (Standardization theorem [9]) Let $\mathcal{R}$ be an OTRS. Every reduction derivation class contains a unique standard reduction derivation.

The following lemma is a corollary of Theorem 4.1.

Lemma 4.7 Let $\mathcal{R}$ be an OTRS, and G be a right-normal goal. If there exists a reduction derivation $G \twoheadrightarrow_{\mathcal{R}_+} \top$ then there exists a standard reduction derivation $G \twoheadrightarrow_{\mathcal{R}_+} \top$.

By the following lemma known as a (kind of) lifting lemma in the theories of TRSs and logic programming, we can obtain a version of a standardization theorem for narrowing. Let $\mathrm{rhs}(G)$ denote a set of terms that are the right-hand sides of all the equations in the goal G.

Lemma 4.8 (Lifting Lemma) Let $\mathcal{R}$ be an OTRS. Suppose we have a proper goal G, and a non-narrowable substitution θ such that $(\mathcal{D}\theta \cup \mathcal{V}(\mathcal{C}od\ \theta)) \cap \mathcal{V}(\mathrm{rhs}(G)) = \emptyset$. If there exists a successful LOI NC-derivation

$$A : \theta G \overset{NC}{\rightsquigarrow}_{\eta} \top$$

then there exists a successful LOI NC-derivation

$$G \overset{NC}{\rightsquigarrow}_{\sigma} \top,$$

such that $\sigma \preceq \eta\theta[\mathcal{V}(G)]$.

Proof: The proof is given in Appendix C.

Theorem 4.2 (Standardization theorem for NC-derivations) Let $\mathcal{R}$ be an OTRS and G be a right-normal goal. If there exists a successful NC-derivation $G \overset{NC}{\rightsquigarrow}_{\theta} \top$, then there exists an LOI NC-derivation $G \overset{NC}{\rightsquigarrow}_{\sigma} \top$ such that $\sigma \preceq \theta$.

Proof: By the correspondence of narrowing and reduction derivations, there exists a reduction derivation $\theta G \twoheadrightarrow_{\mathcal{R}_+} \top$. By Lemma 4.7 there exists a standard reduction derivation $\theta G \twoheadrightarrow_{\mathcal{R}_+} \top$. By Lemma 4.6 the substitution θ is non-narrowable. By Lifting Lemma 4.8 there exists a successful LOI NC-derivation $G \overset{NC}{\rightsquigarrow}_{\sigma} \top$, where $\sigma \preceq \theta$. ∎

Example 4.1 Let $\mathcal{R}$ be an OTRS

$$\begin{cases} f(g(d), w) \to a, \\ g(c) \to g(d). \end{cases}$$

The following NC-derivation

$$f(g(x), g(y)) = a \overset{\mathrm{n}}{\leadsto}_{\{x \mapsto c\}} f(g(d), g(y)) = a \overset{\mathrm{n}}{\leadsto}_{\{y \mapsto c\}} f(g(d), g(d)) = a$$

$$\overset{\mathrm{n}}{\leadsto}_{\emptyset} a = a \overset{\mathrm{f}}{\leadsto}_{\emptyset} \top \tag{15}$$

yields a solution $\theta = \{x \mapsto c, y \mapsto c\}$.

The derivation (15) is not LOI since the NC-derivation

$$\top, g(y) = w, a = a \overset{\mathrm{n}}{\leadsto}_{\{y \mapsto c\}} \top, g(d) = w, a = a \overset{\mathrm{f}}{\leadsto}_{\{w \mapsto d\}} \top$$

is not LOI.

The LOI NC-derivation that yields a solution $\sigma = \{x \mapsto c\} \preceq \theta$ is following.

$$f(g(x), g(y)) = a \overset{\mathrm{n}}{\leadsto}_{\{x \mapsto c\}} f(g(d), g(y)) = a \overset{\mathrm{n}}{\leadsto}_{\emptyset} a = a \overset{\mathrm{f}}{\leadsto}_{\emptyset} \top \tag{16}$$

The derivation (16) is not the only LOI NC-derivation.

$$f(g(x), g(y)) = a \overset{\mathrm{n}}{\leadsto}_{\{x \mapsto d\}} a = a \overset{\mathrm{f}}{\leadsto}_{\emptyset} \top, \tag{17}$$

that yields a solution $\{x \mapsto d\}$ is also LOI.

These are two solutions of the goal $f(g(x), g(y)) = a$. In order to obtain a solution $\{x \mapsto c\}$, narrowing of $g(x)$ is needed. However, to obtain a solution $\{x \mapsto d\}$, narrowing of $g(x)$ is not needed. So the needed-ness of a term to be narrowed depends on the rewrite rule to be applied. The notion of needed-ness defined for reduction does not well fit in narrowing.

4.4 Completeness of LOI NC

We now show that LOI narrowing is complete with respect to normalizable solutions for OTRSs.

The completeness of narrowing is formally stated as follows.

Definition 4.8 Let $\mathcal{R}$ be an OTRS and G a goal.

- Narrowing is complete if for every substitution σ such that $\sigma g \to_{\mathcal{R}+} \top$, there exists a narrowing derivation $G \leadsto_{\tau}^{*} \top$ such that $\tau \preceq_{\mathcal{R}} \sigma[\mathcal{V}(G)]$. Here, $\preceq_{\mathcal{R}}$ is defined as follows: let θ_1 and θ_2 be substitutions. $\theta_1 =_{\mathcal{R}} \theta_2$ if $\theta_1 x =_{\mathcal{R}} \theta_2 x$ for all $x \in \mathcal{V}$, and $\theta_1 \preceq_{\mathcal{R}} \theta_2$ if $\rho\theta_1 =_{\mathcal{R}} \theta_2$, for some substitution ρ.

- In particular, LOI narrowing is complete if for every substitution σ such that $\sigma G \to_{\mathcal{R}+} \top$, there exists an LOI NC-derivation $G \overset{NC}{\leadsto}_{\tau} \top$ such that $\tau \preceq_{\mathcal{R}} \sigma[\mathcal{V}(G)]$.

Using the following well-known completeness theorem of narrowing for confluent TRSs, we can obtain the completeness result of LOI narrrowing.

Theorem 4.3 (completeness of NC) Let $\mathcal{R}$ be a confluent TRS and $G(\triangleq s = t)$ be an (arbitrary) goal. For any normalizable solution θ such that $\theta s =_{\mathcal{R}} \theta t$, there exists a successful NC-derivation $G \overset{NC}{\leadsto}_{\sigma} \top$ such that $\sigma \preceq_{\mathcal{R}} \theta$.

Theorem 4.3 is an easy consequence of the lifting lemma (due to Hullot [10]) for a confluent TRS. The theorem, together with a rigorous proof of Hullot's lifting lemma, is given by Middeldorp and Hamoen [13].

Theorem 4.4 (Completeness of LOI NC) LOI narrowing is complete with respect to normalizable solutions of right-normal goals for OTRSs.

Proof: For every normalizable solution θ of a goal G, there exists an NC-derivation $G \overset{NC}{\leadsto}_{\sigma} \top$ such that $\sigma \preceq_{\mathcal{R}} \theta$ by Theorem 4.3. By Theorem 4.2 there exists an LOI NC-derivation $G \overset{NC}{\leadsto}_{\sigma'} \top$ such that $\sigma' \preceq \sigma$. ∎

5 Calculus OINC

In this section we present a calculus **OINC** that generates LOI NC-derivations. In the calculus **OINC** the inference rules [n] and [f] of **NC** are decomposed into several more primitive inference rules. Furthermore, a computation rule by which to locate narexes is built in **OINC**. Standardization Theorem 4.2 for NC-derivations allows us to deal only with LOI NC-derivations for the completeness result of **OINC**. Hence in **OINC**, all the goals are proper and the equations of the goals are processed from left to right. As in **NC**, we give the calculus **OINC** in the form of an inference system.

Definition 5.1 (OINC) Let $\mathcal{R}$ be an OTRS. A calculus **OINC** for $\mathcal{R}$ is a pair $(\mathcal{G}, OINC)$, where

- $\mathcal{G}$ is a set of proper goals,
- $OINC$ is a set of inference rules defined as follows.

  - [on] outermost narrowing

$$\frac{f(s_1,\ldots,s_n) = t, E}{s_1 = l_1,\ldots,s_n = l_n, r = t, E} \qquad t \notin \mathcal{V}$$

    if there exists a new variant $f(l_1,\ldots,l_n) \to r$ of a rewrite rule in $\mathcal{R}$.

- [d] decomposition

$$\frac{f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n), E}{s_1 = t_1,\ldots,s_n = t_n, E}$$

- [v] variable elimination
  - * [v1]

$$\frac{t = x, E}{\sigma E, \text{where } \sigma{=}\{x \mapsto t\}}$$

  - * [v2]

$$\frac{x = t, E}{\sigma E, \text{where } \sigma{=}\{x \mapsto t\}} \qquad t \notin \mathcal{V}$$

Note:

- There exists indeterminacy between the choice of [on] and [d].
- We do not need an inference rule

$$\frac{t = f(s_1,\ldots,s_n), E}{s_1 = l_1,\ldots,s_n = l_n, t = r, E}$$

  if there exists a new variant $f(l_1,\ldots,l_n) \to r$ of a rewrite rule in $\mathcal{R}$,

  since a narrowable term is never generated on the right-hand side of the equations of a goal.
- The term true is never generated in the inference steps of **OINC**. Hence, unless true's are in an initial goal, true's are never in the goals. Since true's are superfluous in the calculus **OINC**, we remove the true's in the initial goals and assume that all goals (including initial goals) of **OINC** do not contain true's.
- By Proposition 4.1 we have $x \notin \mathcal{V}(t)$ in [v1] and [v2]. Hence, the so-called occur check, i.e., the check of $x \notin \mathcal{V}(t)$, is unnecessary in applying the inference rules [v1] and [v2].

An equation of the form $t = x$ is always processed by the inference rule [v1]. In our context, this is the meaning of lazy narrowing. In some implementation of **OINC**, substitutions are maintained separately, and terms are essentially represented in directed acyclic graphs (dag). The inference rule [v1] coupled with the dag representation of terms may also be regarded the essence of the lazy narrowing.

Relations over goals $\overset{\text{on}}{\rightsquigarrow}$, $\overset{\text{d}}{\rightsquigarrow}$, $\overset{\text{v1}}{\rightsquigarrow}$, $\overset{\text{v2}}{\rightsquigarrow}$, $\overset{OINC}{\rightsquigarrow}$ and their reflexive and transitive closures are defined as in the calculus **NC**. We should note the correspondence between $\overset{\text{on}}{\rightsquigarrow}$, $\overset{\text{d}}{\rightsquigarrow}$, $\overset{\text{v1}}{\rightsquigarrow}$, $\overset{\text{v2}}{\rightsquigarrow}$, and $\overset{OINC}{\rightsquigarrow}$ and $\overset{\text{on}}{\lhd}, \overset{\text{d}}{\lhd}, \overset{\text{v1}}{\lhd}, \overset{\text{v2}}{\lhd}$ and $\overset{OI}{\lhd}$, respectively. By the definitions of the inference rules of **OINC** and the relations $\overset{\text{v1}}{\lhd}, \overset{\text{v2}}{\lhd}, \overset{\text{d}}{\lhd}$ and $\overset{\text{on}}{\lhd}$, it is clear that if G is proper and $G \overset{OINC}{\rightsquigarrow} G'$, then G' is proper.

Example 5.1 We use the OTRS in Example 4.1. The following are OINC-derivations that yield solutions $\{x \mapsto c\}$ and $\{x \mapsto d\}$.

$$
\begin{array}{lll}
f(g(x), g(y)) = a & \overset{\text{on}}{\rightsquigarrow} & g(x) = g(d), g(y) = w, a = a \\
& \overset{\text{on}}{\rightsquigarrow} & x = c, g(d) = g(d), g(y) = w, a = a \\
& \overset{\text{v2}}{\rightsquigarrow}_{\{x \mapsto c\}} & g(d) = g(d), g(y) = w, a = a \\
& \overset{\text{d}}{\rightsquigarrow} & d = d, g(y) = w, a = a \\
& \overset{\text{d}}{\rightsquigarrow} & g(y) = w, a = a \\
& \overset{\text{v1}}{\rightsquigarrow}_{\{w \mapsto g(y)\}} & a = a \\
& \overset{\text{d}}{\rightsquigarrow} & \Box.
\end{array}
$$

$$
\begin{array}{lll}
f(g(x), g(y)) = a & \overset{\text{on}}{\rightsquigarrow} & g(x) = g(d), g(y) = w, a = a \\
& \overset{\text{d}}{\rightsquigarrow} & x = d, g(y) = w, a = a \\
& \overset{\text{v2}}{\rightsquigarrow}_{\{x \mapsto d\}} & g(y) = w, a = a \\
& \overset{\text{v1}}{\rightsquigarrow}_{\{w \mapsto g(y)\}} & a = a \\
& \overset{\text{d}}{\rightsquigarrow} & \Box.
\end{array}
$$

Before we proceed further with **OINC**, we check that **OINC** computes a correct solution.

Definition 5.2 Let $\mathcal{R}$ be an OTRS and G be a proper goal of **OINC**.

- A substitution θ is a solution of G (with respect to **OINC**) if $G \overset{OINC}{\rightsquigarrow}_{\theta} \Box$. Note that $\mathcal{D}\theta \subseteq \mathcal{V}(G)$ by the definition of a substitution formed in the derivation.
- A solution of G is correct if $\theta G \twoheadrightarrow_{\mathcal{R}_+} \top$.

The following proposition shows that the calculus **OINC** indeed computes a correct solution.

Proposition 5.1 (Soundness of OINC) Let $\mathcal{R}$ be an OTRS, and G be a proper goal. If there exists an OINC-derivation $G \overset{OINC}{\rightsquigarrow}_{\theta} \Box$, then there exists a reduction derivation $\theta G \twoheadrightarrow_{\mathcal{R}_+} \top$.

Proof: The proof is given in Appendix B.∎

6 Completeness of OINC

The calculus **OINC** is a complete implementation of **NC**. The completeness of **OINC** states that given a right-normal G, for any successful NC-derivation $G \overset{NC}{\rightsquigarrow}_{\theta} \top$ there exists an OINC-derivation

$G \overset{OINC}{\rightsquigarrow}_{\sigma} \Box$, such that $\sigma \preceq \theta$.

The solution θ is non-narrowable (hence normalized) by Lemma 4.6.

The completeness proof proceeds as follows.

1. We have already seen that for any successful NC-derivation there exists an LOI NC-derivation.
2. We transform a successful LOI NC-derivation A to another successful NC-derivation A' that is less complex than A.
3. We connect A and A' by the inference steps of **OINC**.
4. By an inductive argument, we show that A is replaced by an OINC-derivation.

This proof method was employed by Hölldobler in proving the completeness of an inference system TRANS [8].

Proposition 6.1 Let $\mathcal{R}$ be an OTRS, and G be a proper goal of **NC**. If there exists an LOI NC-derivation $G \overset{NC}{\rightsquigarrow}_\theta \top$, then there exists an OINC-derivation $\overline{G} \overset{OINC}{\rightsquigarrow}_\theta \Box$, where $\overline{G}$ is a goal obtained from G by removing true's in G.

Proof: By the transfinite induction on $\lhd$. Obviously the result holds for the base case. Let $A : G \overset{NC}{\rightsquigarrow} \top$ and G be a proper goal $\top, s = t, E$. Assume that the result holds for any NC-derivation A' such that $A' \lhd A$. We distinguish the following four cases.

(1) t is a variable x.

The term s is not a variable x by the property (G1). Hence, by assumption, A is written as follows.

$$A : \top, s = x, E \overset{\mathrm{f}}{\rightsquigarrow}_{\theta_1(=\{x \mapsto s\})} \top, \theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_2} \top$$

By Lemma 4.1, there exists an NC-derivation $A' : \theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_2} \top$ such that $A' \overset{\mathrm{v1}}{\lhd} A$. Since A is LOI, A' is LOI. By the inference rule [v1] of **OINC**, we have

$$s = x, \overline{E} \overset{\mathrm{v1}}{\rightsquigarrow}_{\theta_1} \theta_1 \overline{E}.$$

Therefore, by the induction hypothesis, we have an OINC-derivation

$$s = x, \overline{E} \overset{\mathrm{v1}}{\rightsquigarrow}_{\theta_1} \theta_1 \overline{E} \overset{OINC}{\rightsquigarrow}_{\theta_2} \Box.$$

(2) t is a non-variable term.

(2-1) s is a variable x.

Since t is not narrowable by the property (G2) and $t \not\equiv x$, by the property (G1) the LOI NC-derivation A is written as follows.

$$A : \top, x = t, E \overset{\mathrm{f}}{\rightsquigarrow}_{\theta_1(=\{x \mapsto t\})} \top, \theta_1 E \overset{NC}{\rightsquigarrow}_{\theta_2} \top.$$

The rest of the proof is the same as that of the case (1) using Lemma 4.2.

(2-2) s is a term $f(s_1, \ldots, s_n)$. We further distinguish the following two cases.

(2-2a) The LOI NC-derivation A is written as follows.

$$A : \top, f(s_1, \ldots, s_n) = t, E \overset{NC}{\rightsquigarrow}_\theta \top,$$

where a new variant $f(l_1, \ldots, l_n) \to r$ of a rewrite rule in $\mathcal{R}$ is used to narrow the descendant of $f(s_1, \ldots, s_n) = t$. By Lemma 4.5, there exists an NC-derivation

$$A' : s_1 = l_1, \ldots, s_n = l_n, r = t, E \overset{NC}{\rightsquigarrow}_\theta \top$$

such that $A' \overset{\text{on}}{\lhd} A$. Since A is LOI, A' is LOI. By the induction hypothesis there exists an OINC-derivation

$$f(s_1, \ldots, s_n) = t, \overline{E} \overset{\text{on}}{\rightsquigarrow} s_1 = l_1, \ldots, s_n = l_n, r = t, \overline{E} \overset{OINC}{\rightsquigarrow}_\theta \Box.$$

(2-2b) The LOI NC-derivation A is written as follows.

$$A : \top, f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n), E \overset{NC}{\rightsquigarrow} \top,$$

where the descendants of $f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$ is never narrowed at the position 1.

By Lemma 4.4, there exists an NC-derivation

$$A' : s_1 = t_1, \ldots, s_n = t_n, E \overset{NC}{\rightsquigarrow}_\theta \top$$

such that $A' \overset{\text{d}}{\lhd} A$. Since A is LOI, A' is LOI.

Hence, we have an OINC-derivation

$$f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n), \overline{E} \overset{\text{d}}{\rightsquigarrow} s_1 = t_1, \ldots, s_n = t_n, \overline{E} \overset{OINC}{\rightsquigarrow}_\theta \Box$$

by the induction hypothesis. ∎

Theorem 6.1 (Completeness of OINC) Let $\mathcal{R}$ be an OTRS and G be a right-normal goal. If there exists an NC-derivation $G \overset{NC}{\rightsquigarrow}_\theta \top$, then there exists an OINC-derivation $G \overset{OINC}{\rightsquigarrow}_\sigma \Box$ such that $\sigma \preceq \theta$.

Proof: By Theorem 4.2 there exists an LOI NC-derivation $A : G \overset{NC}{\rightsquigarrow}_\sigma \top$ such that $\sigma \preceq \theta$. By Proposition 6.1 there exists an OINC-derivation $G \overset{OINC}{\rightsquigarrow}_\sigma \Box$. ∎

7 Calculus s-OINC

We next extend **OINC** in order to handle strict equations efficiently. To motivate the extension let us take an example.

Example 7.1 With respect to $\mathcal{R}^{\equiv}$ in Example 3.1, we solve a goal $f(g(x)) \equiv f(y)$ in **OINC**.

$$f(g(x)) \equiv f(y) \overset{\text{on}}{\rightsquigarrow} f(g(x)) = 1, f(y) = 1, \text{true} = \text{true}$$

$$\overset{\text{on}}{\rightsquigarrow} g(x) = w_1, w_1 = 1, f(y) = 1, \text{true} = \text{true}$$

$$\overset{\text{vl}}{\rightsquigarrow}_{\{w_1 \mapsto g(x)\}} g(x) = 1, f(y) = 1, \text{true} = \text{true} \overset{OINC}{\rightsquigarrow} \Box. \tag{18}$$

This OINC-derivation seems to be very redundant. With some insight we can think of an inference step that can bypass some of the above steps. Let us try to apply a kind of $\overset{n}{\rightsquigarrow}$-step on the strict equation directly.

$$f(g(x)) \equiv f(y) \rightsquigarrow g(x) = w, w \equiv f(y).$$

We denote this step by $\overset{ons}{\rightsquigarrow}$. Then we can obtain a new derivation.

$$f(g(x)) \equiv f(y) \overset{ons}{\rightsquigarrow} g(x) = w_1, w_1 \equiv f(y) \overset{vl}{\rightsquigarrow}_{\{w_1 \mapsto g(x)\}} g(x) \equiv f(y)$$

$$\overset{OINC}{\rightsquigarrow}_{\{x \mapsto 1,\ldots\}} 1 \equiv f(y) \overset{ons}{\rightsquigarrow} y = w_2, 1 \equiv w_2 \rightsquigarrow \cdots \rightsquigarrow \Box. \tag{19}$$

We see that in the derivation (19) the equation $g(x) = w_1$ is generated in one step, whereas in the derivation (18) it is generated in two $\overset{n}{\rightsquigarrow}$-steps.

The problem with OINC in handling the strict equations is not only the number of redundant steps, but the difficulty of choosing an adequate rewrite rule for strict equations when there are many constructor symbols $c \in \mathcal{F}_C$. In the above example, in the derivation (18) we select the rewrite rule $1 \equiv 1 \rightarrow$ true immediately in the first step of the derivation. In practice, this is impossible without try-and-backtracks. We will circumvent these difficulties in the following way. The basic idea for taking the shortcut that we saw above is to narrow the left- and right-hands of the strict equations independently. Suppose that a goal $s \equiv t$ is given. We narrow s and t independently until s and t become constructor terms, say $c(s')$ and $c(t')$, respectively (if t becomes $c'(t')$ where $c \not\equiv c'$, this derivation will never become successful). Then we repeat this process with s' and t'.

We are now ready to give the inference rules for strict equations.

7.1 Inference-rules for strict equations

- [ons] outermost narrowing for strict equations

$$\frac{f(s_1,\ldots,s_n) \equiv t, E}{s_1 = l_1,\ldots,s_n = l_n, r \equiv t, E} \quad \text{or} \quad \frac{s \equiv f(t_1,\ldots,t_n), E}{t_1 = l_1,\ldots,t_n = l_n, s \equiv r, E}$$

  if there exist a new variant $f(l_1,\ldots,l_n) \rightarrow r$ of a rewrite rule in $\mathcal{R}$.

- [ds] decomposition for strict equations

$$\frac{c(s_1,\ldots,s_n) \equiv c(t_1,\ldots,t_n), E}{s_1 \equiv t_1,\ldots,s_n \equiv t_n, E}$$

  where $c \in \mathcal{F}_C$.

- [ims] imitation for strict equations

$$\frac{c(s_1,\ldots,s_n) \equiv y, E}{\theta(s_1 \equiv y_1,\ldots,s_n \equiv y_n, E)} \quad \text{or} \quad \frac{y \equiv c(t_1,\ldots,t_n), E}{\theta(y_1 \equiv t_1,\ldots,y_n \equiv t_n, E)}$$

  where $c \in \mathcal{F}_C$ and $\theta = \{y \mapsto c(y_1,\ldots,y_n)\}$.

- [ts] elimination of trivial strict equations

$$\frac{x \equiv y, E}{\sigma E}$$

$$\text{where } \sigma = \begin{cases} \{x \mapsto y\} & \text{if } x \not\equiv y \\ \emptyset & \text{otherwise.} \end{cases}$$

Since the strict equations are symmetrical in narrowing, we need two rules in the inference rules [ons] and [ims]. Note also that thanks to the inference rules for the strict equations we no longer need rewrite rules for the strict equations.

Except for the inference rule [ims], the new inference rules are easy to understand. The inference rule [ims] is used to narrow the subterms of a constructor term. Let us take an example.

Example 7.2 Let $\mathcal{F}_\mathcal{C} = \{1, c_1\}$, where c_1 is a constructor symbol of arity one. We use a TRS $\mathcal{R}$ of Example 3.1, and solve a goal $G \triangleq c_1(f(x)) \equiv y$. A successful derivation is as follows.

$$G \overset{\text{ims}}{\rightsquigarrow}_{\{y \mapsto c_1(y_1)\}} f(x) \equiv y_1 \overset{\text{ons}}{\rightsquigarrow} x = w, w \equiv y_1 \overset{\text{vl}}{\rightsquigarrow}_{\{w \mapsto x\}} x \equiv y_1 \overset{\text{ts}}{\rightsquigarrow}_{\{x \mapsto y_1\}} \Box.$$

The solution obtained in this derivation is $\{y \mapsto c_1(y_1), x \mapsto y_1\}$.

Let $s\text{-}OINC = \{[\text{ons}], [\text{ds}], [\text{ims}], [\text{ts}]\} \cup OINC$. We define a new calculus **s-OINC** $= (\mathcal{G}, s\text{-}OINC)$, where $\mathcal{G}$ is a set of proper goals as in **OINC**. The initial goal is a sequence of a right-normal equations, some of which may be a strict equation.

7.2 Completeness of s-OINC

We do not iterate the soundness theorem for **s-OINC**. The completeness theorem for **s-OINC** is obtained from the following proposition.

Proposition 7.1 Let $\mathcal{R}$ be an OTRS, and G be a proper goal. If there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$G \overset{OINC}{\rightsquigarrow}_\theta \Box$$

then there exists an **s-OINC** derivation with respect to $\mathcal{R}$

$$G \overset{s\text{-}OINC}{\rightsquigarrow}_\sigma \Box, \text{such that } \sigma \preceq \theta.$$

Proof: The proof is given in Appendix D.∎

Our final result is the following completeness theorem for **s-OINC**.

Theorem 7.1 (Completeness of s-OINC) Let $\mathcal{R}$ be an OTRS and G be a right-normal goal. If there exists an NC-derivation $G \overset{NC}{\rightsquigarrow}_\theta \top$, then there exists an s-OINC-derivation $G \overset{s\text{-}OINC}{\rightsquigarrow}_\sigma \Box$ such that $\sigma \preceq \theta$.

8 Conclusion

We have presented a leftmost outside-in narrowing and shown the narrowing calculus **OINC** based on the leftmost outside-in narrowing. The calculus **OINC** realizes lazy evaluation in narrowing in that it delays narrowing on narrowable terms that are to be bound to variables. Furthermore it enjoys the property of completeness for orthogonal TRSs with respect to normalized solutions.

In order to use the calculus **OINC** as a model of computation of functional-logic programming we extended **OINC** to incorporate strict equality. The extension results in a new narrowing calculus **s-OINC**. It has been also shown that the calculus **s-OINC** enjoys the same completeness property as **OINC**.

Finally, we would like to mention the implementation of the calculi that we discussed. An interpreter of **s-OINC** is straightforward to implement. Furthermore, a compiler for **s-OINC** (for constructor-based TRSs) targeted at modified WAM has been implemented [17].

References

[1] S. Antoy, R. Echahed, and M. Hanus. A needed narrowing strategy. In *Proc. 21st ACM Symposium on Principles of Programming Languages*, pages 268–279, Portland, 1994.

[2] A. Bockmayr, S. Krischer, and A. Werner. Narrowing strategies for arbitrary canonical rewrite systems. Technical report, Universität Karlsruhe, 1993.

[3] P. H. Cheong. Compiling lazy narrowing into Prolog. To appear in New Generating Computing, 199?

[4] J. Darlington and Y.-K. Guo. Narrowing and unification in functional programming- an evaluation mechanism for absolute set abstraction. In *Proceedings of Rewriting Techniques and Applications, Lecture Notes in Computer Science 355*, pages 92–108, 1989.

[5] L. Friborg. SLOG: a logic programming language interpreter based on clausal superposition and rewriting. In *Proceedings of the 2nd IEEE Symposium on Logic Programming, Boston*, pages 172–184, 1985.

[6] E. Giovannetti, G. Levi, C. Moiso, and C. Palamidessi. Kernel-LEAF: A logic plus functional language. *Journal of Computer and System Sciences*, 42(2):139–185, 1991.

[7] M. Hanus. Compiling logic programs with equality. In *Proceedings of the 2nd Workshop on Programming Language Implementation and Logic Programming, Lecture Notes in Computer Science 456*, pages 387–401, 1990.

[8] S. Hölldobler. Foundations of equational logic programming. *Lecture Notes in Artificial Intelligence*, 353, 1989.

[9] G. Huet and J. Lévy. Computations in orthogonal rewriting systems, I. In J.-L. Lassez and G. Plotkin, editors, *Computational logic: essays in honor of Alan Robinson*, pages 395–414. The MIT Press, 1991.

[10] J. Hullot. Canonical forms and unification. In *Proceedings of the 5th Conference on Automated Deduction, Lecture Notes in Computer Science 87*, pages 318–334, 1980.

[11] H. C.R. Lock. *The Implementation of Functional Logic Programming Languages.* PhD thesis, Universität Karlsruhe, 1992. Published as GMD-Bericht Nr. 208, by R. Oldenbourg Verlag, 1993.

[12] R. Loogen, F. L. Fraguas, and M. Rodríguez-Artalejo. A demand driven computation strategy for lazy narrowing. In *Proceedings of the 5th International Symposium of Programming Language Implementation and Logic Programming, Lecture Notes in Computer Science 714*, pages 184–200, 1993.

[13] A. Middeldorp and E. Hamoen. Completeness results for basic narrowing. *Applicable Algebra in Engineering, Communication and Computing*, 1994. To appear.

[14] J. J. Moreno-Navarro and M. Rodríguez-Artalejo. Logic programming with functions and predicates: The language BABEL. *Journal of Logic Programming*, 12:191–223, 1992.

[15] S. Narain. A technique for doing lazy evaluation in logic. *Journal of Logic Programming*, 3:259–276, 1986.

[16] U. S. Reddy. Narrowing as the operational semantics of functional languages. In *The Proceedings IEEE International Symposium on Logic Programming*, pages 138–151, Boston, 1985.

[17] T. Suzuki. Implementation issues of a functional-logic programming language. Master's thesis, University of Tsukuba, 1993.

[18] J.-H. You. Enumerating outer narrowing derivations for constructor-based term rewriting systems. *Journal of Symbolic Computation*, 7:319–341, 1989.

A Proof of Switching Lemma

Lemma A.1 (Switching Lemma) Let $\mathcal{R}$ be a TRS, and $G \triangleq s_1 = t_1, s_2 = t_2$ be a goal such that $\mathcal{V}(s_i) \cap \mathcal{V}(t_j) = \emptyset$ for $i,j \in \{1,2\}$ and $\mathcal{V}(t_1) \cap \mathcal{V}(t_2) = \emptyset$. If there exists an NC-derivation

$$A : G \overset{\mathrm{n}}{\leadsto}_{\theta_1|_{\mathcal{V}(s_1)}} \theta_1 s_1[r]_u = t_1, \theta_1 s_2 = t_2 \overset{\mathrm{f}}{\leadsto}_{\theta_2} \theta_2\theta_1 s_1[r]_u = t_2, \mathrm{true},$$

where in the first step $s_1|_u$ is narrowed using a new variant $l \to r$ of a rewrite rule in $\mathcal{R}$ and a most general unifier θ_1 of $s_1|u$ and l, then there exists an NC-derivation

$$A' : G \overset{\mathrm{f}}{\leadsto}_{\sigma_1} \sigma_1 s_1 = t_1, \mathrm{true} \overset{\mathrm{n}}{\leadsto}_{\sigma_2|_{\mathcal{V}(\sigma s_1)}} \sigma_2\sigma_1 s_1[r]_u = t_2, \mathrm{true},$$
where σ_2 is a most general unifier of $(\sigma_1 s_1)|u$ and l,

such that $\theta_2\theta_1 = \sigma_2\sigma_1$ modulo renaming.

Proof: Since $\theta_1 s_2$ and $t_2 (\equiv \theta_1 t_2)$ are unifiable, s_2 and t_2 are unifiable by a most general unifier σ_1. Hence we obtain the first step of the derivation A'. We next show $(\sigma_1 s_1)|_u$ is narrowable by the same rewrite rule $l \to r$. By assumption, we have $\theta_1(s_1|_u) \equiv \theta_1 l$ and $\theta_2\theta_1 s_2 \equiv \theta_2 t_2$. Since $\mathcal{D}\theta_1 \cap \mathcal{V}(t_2) = \emptyset$, we have $\theta_2\theta_1 s_2 \equiv \theta_2\theta_1 t_2$. Since σ_1 is a most general unifier of s_2 and t_2, there exists a substitution η such that

$$\eta\sigma_1 = \theta_2\theta_1. \tag{20}$$

On the other hand, we have

$$\eta\sigma_1(s_1|_u) \equiv \theta_2\theta_1(s_1|_u) \equiv \theta_2\theta_1 l \equiv \eta\sigma_1 l \equiv \eta l.$$

Hence $\sigma_1(s_1|_u)$ is narrowable by $l \to r$. There exist a most general unifier σ_2 of $\sigma_1(s_1|_u)$ and l, and a substitution η' such that

$$\eta = \eta'\sigma_2. \tag{21}$$

Hence, we obtain the second step of NC-derivation A'.

From (20) and (21), we obtain

$$\eta'\sigma_2\sigma_1 = \theta_2\theta_1. \tag{22}$$

Similarly, we can prove

$$\xi\theta_2\theta_1 = \sigma_2\sigma_1 \tag{23}$$

for some substitution ξ. From (22) and (23), we see $\theta_2\theta_1 = \sigma_2\sigma_1$ modulo renaming.

B Proof of the soundness of OINC

Proposition B.1 (Soundness of OINC) Let $\mathcal{R}$ be an OTRS, and G be a proper goal. If there exists an OINC-derivation $G \overset{OINC}{\rightsquigarrow}_\theta \Box$, then there exists a reduction derivation $\theta G \to_{\mathcal{R}_+} \top$.

Proof: By the induction on the length of the OINC-derivation. The result obviously holds for the base case. Let $A : G \overset{OINC}{\rightsquigarrow}_\sigma G' \overset{OINC}{\rightsquigarrow}_{\theta'} \Box$, and suppose the result holds for the derivation $G' \overset{OINC}{\rightsquigarrow}_{\theta'} \Box$. We distinguish the following cases depending on the first step of the derivation.

Case [v1]: Let the OINC-derivation A be $s = x, E \overset{\mathrm{v1}}{\rightsquigarrow}_{\sigma(=\{x \mapsto s\})} \sigma E \overset{OINC}{\rightsquigarrow}_{\theta'} \Box$. We have $\theta = \theta'\sigma = \theta' \cup \{x \mapsto \theta' s\}$. Since $\theta(s = x, E) \equiv \theta' s = \theta' s, \theta E$ by the left-independence of the goal, we have the following reduction

$$\theta' s = \theta' s, \theta E \to_{\mathcal{R}_+} \mathrm{true}, \theta E \to_{\mathcal{R}_+} \top.$$

The first reduction is by the use of $x = x \to \mathrm{true}$, and the second by the induction hypothesis.

Case [v2]: Similarly to the case [v1].

Case [on]: Let the OINC-derivation A be

$$f(s_1,\ldots,s_n) = t, E \overset{\text{on}}{\rightsquigarrow} s_1 = l_1,\ldots,s_n = l_n, r = t, E \overset{OINC}{\rightsquigarrow}_{\theta'} \Box,$$

where in the first step a rewrite rule $f(l_1,\ldots,l_n) \to r$ is used. We have $\theta = \theta'[\mathcal{V}(G)]$. By the induction hypothesis, there exists a reduction derivation

$$\theta s_1 = \theta' l_1,\ldots,\theta s_n = \theta' l_n, \theta' r = \theta t, \theta E \twoheadrightarrow_{\mathcal{R}_+} \top, \theta' r = \theta t, \theta E \twoheadrightarrow_{\mathcal{R}_+} \top. \tag{24}$$

Since we have $\theta s_i = \theta' l_i \twoheadrightarrow_{\mathcal{R}_+}$ true for every $i \in \{1,\ldots,n\}$, there exists a reduction derivation

$$\theta s_i = \theta' l_i \twoheadrightarrow_{\mathcal{R}_+} s_i' = l_i' \twoheadrightarrow_{\mathcal{R}_+} \text{true, where } s_i' \equiv l_i'. \tag{25}$$

By the non-ambiguity, l_i is a normal form. Hence, $l_i' \equiv \tau l_i$, where τ is such that

$$\theta' x \twoheadrightarrow_{\mathcal{R}} \tau x \text{ for all } x \in \mathcal{V}(l_i).$$

From (25), we have $\theta' s_i \twoheadrightarrow_{\mathcal{R}} \tau l_i$, and hence

$$\theta f(s_1,\ldots,s_n) = \theta t, \theta E \twoheadrightarrow_{\mathcal{R}} f(\tau l_1,\ldots,\tau l_n) = \theta t, \theta E \twoheadrightarrow_{\mathcal{R}_+} \tau r = \theta t, \theta E.$$

By (24) and the confluence of $\mathcal{R}$, we have $\theta(f(s_1,\ldots,s_n) = t, E) \twoheadrightarrow_{\mathcal{R}_+} \top$.

Case [d]: Let the OINC-derivation A be

$$\begin{array}{lll} f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n), E & \overset{\text{d}}{\rightsquigarrow} & s_1 = t_1,\ldots,s_n = t_n, E \\ & \overset{OINC}{\rightsquigarrow}_{\theta} & \Box. \end{array}$$

By the induction hypothesis, $\theta s_1 = \theta t_1,\ldots,\theta s_n = \theta t_n, \theta E \twoheadrightarrow_{\mathcal{R}_+} \top$. Hence, for $i = 1,\ldots,n$, we have $\theta s_i = \theta t_i \twoheadrightarrow_{\mathcal{R}_+}$ true. This implies that there exists a terms r_i such that $\theta s_i \twoheadrightarrow_{\mathcal{R}} r_i$ and $\theta t_i \twoheadrightarrow_{\mathcal{R}} r_i$. Hence, there exists a reduction derivation

$$\begin{array}{lll} \theta(f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n), E) & \twoheadrightarrow_{\mathcal{R}_+} & f(r_1,\ldots,r_n) = f(r_1,\ldots,r_n), \theta E \\ & \twoheadrightarrow_{\mathcal{R}_+} & \text{true}, \theta E \\ & \twoheadrightarrow_{\mathcal{R}_+} & \top. \ \blacksquare \end{array}$$

C Proof of Lifting Lemma

Lemma C.1 (Lifting Lemma) Let $\mathcal{R}$ be an OTRS. Suppose we have a proper goal G, and a non-narrowable substitution θ such that $(\mathcal{D}\theta \cup \mathcal{V}(\mathcal{C}od\ \theta)) \cap \mathcal{V}(\text{rhs}(G)) = \emptyset$. If there exists a successful LOI NC-derivation

$$A : \theta G \overset{NC}{\rightsquigarrow}_{\eta} \top$$

then there exists a successful LOI NC-derivation

$G \overset{NC}{\rightsquigarrow}_{\sigma} \top$ such that $\sigma \preceq \eta\theta[\mathcal{V}(G)]$.

Proof: By the transfinite induction on $\overset{OI}{\lhd}$. Obviously the result holds in the base case. Assume that the result holds for any NC-derivation A' such that $A' \overset{OI}{\lhd} A$.

We distinguish the following three cases.

(1) $G \triangleq \top, s = x, E$

By assumption we have

$$A : (\theta G \triangleq)\top, \theta s = x, \theta E \overset{\mathrm{f}}{\rightsquigarrow}_{\eta_1(=\{x \mapsto \theta s\})} \top, \eta_1\theta E \overset{NC}{\rightsquigarrow}_{\eta_2} \top$$

with $\eta = \eta_2\eta_1[\mathcal{V}(\theta G)]$.

Let A' be $\eta_1\theta E \overset{NC}{\rightsquigarrow}_{\eta_2} \top$. We have $A' \overset{\mathrm{v1}}{\lhd} A$, and hence A' is LOI.

On the other hand, we have

$$\top, s = x, E \overset{\mathrm{f}}{\rightsquigarrow}_{\sigma_1(=\{x \mapsto s\})} \top, \sigma_1 E.$$

We know that $\eta_1\theta = \theta\sigma_1$ by assumption.

By the induction hypothesis, there exists an LOI NC-derivation $\sigma_1 E \overset{NC}{\rightsquigarrow}_{\sigma_2} \top$ such that $\sigma_2 \preceq \eta_2\theta[\mathcal{V}(\sigma_1 E)]$. We have $\sigma_2\sigma_1 \preceq \eta_2\theta\sigma_1 = \eta_2\eta_1\theta[\mathcal{V}(G)]$.

Therefore, we have an LOI NC-derivation

$$\top, s = x, E \overset{NC}{\rightsquigarrow}_{\sigma} \top$$

such that $\sigma \preceq \eta\theta[\mathcal{V}(G)]$.

(2) $G \triangleq \top, x = t, E$ where $t \notin \mathcal{V}$ and $\theta x \equiv s$

By assumption we have

$$A : (\theta G \triangleq)\top, s = t, \theta E \overset{NC}{\rightsquigarrow}_{\eta} \top.$$

Since s and t are non-narrowable, they should be unifiable with a most general unifier η_1. The substitution η_1 is obviously non-narrowable.

The LOI NC-derivation A can be written as

$$\top, s = t, \theta E \overset{\mathrm{f}}{\rightsquigarrow}_{\eta_1} \top, \eta_1\theta E \overset{NC}{\rightsquigarrow}_{\eta_2} \top, \text{and } \eta = \eta_2\eta_1[\mathcal{V}(\theta G)]. \tag{26}$$

Let A' be $\eta_1\theta E \overset{NC}{\rightsquigarrow}_{\eta_2} \top$. We have $A' \overset{OI*}{\lhd} A$, and hence A' is LOI. On the other hand, we have

$$x = t, E \overset{\mathrm{f}}{\rightsquigarrow}_{\sigma_1(=\{x \mapsto t\})} \sigma_1 E.$$

We define a substitution τ as follows.

$$\tau z = \begin{cases} \eta_1 z & \text{if } z \in \mathcal{V}(t) \\ \eta_1 \theta z & \text{otherwise.} \end{cases}$$

Then, using the assumption on θ, we have

$$\eta_1 \theta = \tau \sigma_1. \tag{27}$$

Since η_1 and θ are non-narrowable, the substitution τ is non-narrowable and satisfies that $(\mathcal{D}\tau \cup \mathcal{V}(\mathcal{C}od\ \tau)) \cap \mathcal{V}(\mathrm{rhs}(\sigma_1 E)) = \emptyset$. By the induction hypothesis there exists an LOI NC-derivation

$$\sigma_1 E \overset{NC}{\rightsquigarrow}_{\sigma_2} \top$$

such that $\sigma_2 \preceq \eta_2 \tau[\mathcal{V}(\sigma_1 E)]$. Therefore, we have an LOI NC-derivation

$$\top, x = t, E \overset{\mathrm{f}}{\rightsquigarrow}_{\sigma_1} \top, \sigma_1 E \overset{NC}{\rightsquigarrow}_{\sigma_2} \top.$$

Since $\sigma = \sigma_2 \sigma_1 \restriction_{\mathcal{V}(G)}$ and $\sigma_2 \preceq \eta_2 \tau[\mathcal{V}(G)]$ because of $x \notin \mathcal{D}\sigma_2$, we have

$$\sigma \preceq \eta_2 \tau \sigma_1[\mathcal{V}(G)]. \tag{28}$$

From (27) and (28), we have $\sigma \preceq \eta_2 \eta_1 \theta[\mathcal{V}(G)]$, and hence $\sigma \preceq \eta\theta[\mathcal{V}(G)]$ from (26).

(3) $G \triangleq \top, f(s_1, \ldots, s_n) = t, E$ where $t \notin \mathcal{V}$
By assumption, A is

$$A : (\theta G \triangleq)\top, f(\theta s_1, \ldots, \theta s_n) = t, \theta E \overset{NC}{\rightsquigarrow}_{\eta} \top.$$

By the definition of LOI, we have the following two cases.

(3-a) $A' : \theta s_1 = l_1, \ldots, \theta s_n = l_n, r = t, \theta E \overset{NC}{\rightsquigarrow}_{\eta} \top$ such that $A' \overset{\mathrm{on}}{\lhd} A$
Since $\theta(s_1 = l_1, \ldots, s_n = l_n, r = t, E) \overset{NC}{\rightsquigarrow}_{\eta} \top$, there exists an LOI NC-derivation

$$(G' \triangleq) s_1 = l_1, \ldots, s_n = l_n, r = t, E \overset{NC}{\rightsquigarrow}_{\sigma'} \top \tag{29}$$

such that $\sigma' \preceq \eta\theta[\mathcal{V}(G')]$. From (29) we can construct an LOI NC-derivation as shown in Lemma 4.5.

$$\top, f(s_1, \ldots, s_n) = t, E \overset{NC}{\rightsquigarrow}_{\sigma} \top \text{ such that } \sigma \preceq \eta\theta[\mathcal{V}(G)].$$

(3-b) $t \triangleq f(t_1, \ldots, t_n)$, and $A' : \theta s_1 = t_1, \ldots, \theta s_n = t_n, \theta E \overset{NC}{\rightsquigarrow}_{\eta} \top$ such that $A' \overset{\mathrm{d}}{\lhd} A$
The rest of the proof is similar to the case (3-a). ∎

D Completeness proof of s-OINC

Let $|A|$ denote the length of the OINC-derivation A.

Lemma D.1 Let $\mathcal{R}$ be an OTRS. If there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A : (s_1 \equiv t_1 \wedge \ldots \wedge s_n \equiv t_n) = \text{true} \overset{OINC}{\rightsquigarrow}_{\theta} \Box \quad (n \geq 1)$$

then there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A' : s_1 \equiv t_1, \ldots, s_n \equiv t_n \overset{OINC}{\rightsquigarrow}_{\theta} \Box, \text{and } |A'| < |A|.$$

Proof: Induction on n. If $n = 1$, the result holds trivially. Suppose the result holds for $n - 1$. The derivation A is written as follows.

$$\begin{array}{ll}
(G \triangleq) & (s_1 \equiv t_1 \wedge \ldots \wedge s_n \equiv t_n) = \text{true} \\
\overset{\text{on}}{\rightsquigarrow} & s_1 \equiv t_1, (s_2 \equiv t_2 \wedge \ldots \wedge s_n \equiv t_n) = x, x = \text{true} \\
\overset{OINC}{\rightsquigarrow}_{\sigma_1} & \sigma_1(s_2 \equiv t_2 \wedge \ldots \wedge s_n \equiv t_n) = x, x = \text{true} \\
\overset{\text{v1}}{\rightsquigarrow}_{\sigma_2} & \sigma_1(s_2 \equiv t_2 \wedge \ldots \wedge s_n \equiv t_n) = \text{true} \\
\overset{OINC}{\rightsquigarrow}_{\sigma_3} & \Box.
\end{array}$$

By the induction hypothesis, there exists an OINC-derivation

$$\sigma_1(s_2 \equiv t_2), \ldots, \sigma_1(s_n \equiv t_n) \overset{OINC}{\rightsquigarrow}_{\sigma_3} \Box.$$

Thus we have

$$A' : s_1 \equiv t_1, \ldots, s_n \equiv t_n \overset{OINC}{\rightsquigarrow}_{\sigma_1} \sigma_1(s_2 \equiv t_2), \ldots, \sigma_1(s_n \equiv t_n) \overset{OINC}{\rightsquigarrow}_{\sigma_3} \Box.$$

We can easily check that $\sigma_3\sigma_2\sigma_1 \lceil_{\mathcal{V}(G)} = \sigma_3\sigma_1$ and that $|A'| < |A|$. ∎

Lemma D.2 Let $\mathcal{R}$ be an OTRS. If there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A : (G \triangleq) f(s_1, \ldots, s_n) \equiv t, E \overset{OINC}{\rightsquigarrow}_{\theta} \Box$$

then there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A' : s_1 = l_1, \ldots, s_n = l_n, r \equiv t, E \overset{OINC}{\rightsquigarrow}_{\theta'} \Box, \text{ such that } \theta = \theta'[\mathcal{V}(G)] \text{ and } |A'| < |A|.$$

Proof: The derivation A is written as follows.

$$\begin{array}{ll}
 & f(s_1, \ldots, s_n) \equiv t, E \\
\overset{\text{on}}{\rightsquigarrow} & f(s_1, \ldots, s_n) = c(x_1, \ldots, x_m), t = c(y_1, \ldots, y_m), (x_1 \equiv y_1 \wedge \ldots \wedge x_m \equiv y_m) = \text{true}, E \\
\overset{\text{on}}{\rightsquigarrow} & s_1 = l_1, \ldots, s_n = l_n, r = c(x_1, \ldots, x_m), t = c(y_1, \ldots, y_m), (x_1 \equiv y_1 \wedge \ldots \wedge x_m \equiv y_m) = \text{true} \\
\overset{OINC}{\rightsquigarrow}_{\tau} & \tau r = c(x_1, \ldots, x_m), \tau t = c(y_1, \ldots, y_m), (x_1 \equiv y_1 \wedge \ldots \wedge x_m \equiv y_m) = \text{true}, \tau E \\
\overset{OINC}{\rightsquigarrow}_{\sigma} & \Box
\end{array}$$

Since $s_1 = l_1, \ldots, s_n = l_n \overset{OINC}{\rightsquigarrow}_\tau \Box$, we have

$$\begin{array}{ll} & A' : s_1 = l_1, \ldots, s_n = l_n, r \equiv t, E \\ \overset{OINC}{\rightsquigarrow}_\tau & \tau r \equiv \tau t, \tau E \\ \overset{\text{on}}{\rightsquigarrow} & \tau r = c(x_1, \ldots, x_m), \tau t = c(y_1, \ldots, y_m), (x_1 \equiv y_1 \wedge \ldots \wedge x_m \equiv y_m) = \text{true}, \tau E \\ \overset{OINC}{\rightsquigarrow}_\sigma & \Box. \end{array}$$

We can easily check that $|A'| < |A|$. ∎

Note that we assume $m > 0$ in the first step of the derivation (30). The case $m = 0$ is similarly proved.

Lemma D.3 Let $\mathcal{R}$ be an OTRS. If there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A : c(s_1, \ldots, s_n) \equiv c(t_1, \ldots, t_n), E \overset{OINC}{\rightsquigarrow}_\theta \Box$$

then there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A' : s_1 \equiv t_1, \ldots, s_n \equiv t_n, E \overset{OINC}{\rightsquigarrow}_\theta \Box, \text{such that } |A'| < |A|.$$

Proof: We first consider the case $n > 0$.
The derivation A is written as follows.

$$\begin{array}{ll} (G \triangleq) & c(s_1, \ldots, s_n) \equiv c(t_1, \ldots, t_n), E \\ \overset{\text{on}}{\rightsquigarrow} & c(s_1, \ldots, s_n) = c(x_1, \ldots, x_n), c(t_1, \ldots, t_n) = c(y_1, \ldots, y_n), (x_1 \equiv y_1 \wedge \ldots \wedge x_n \equiv y_n) = \text{true} \\ \overset{OINC}{\rightsquigarrow}_{\theta_1} & (s_1 \equiv t_1 \wedge \ldots \wedge s_n \equiv t_n) = \text{true}, E \\ \overset{OINC}{\rightsquigarrow}_{\theta_2} & \Box. \end{array}$$

By Lemma D.1, we have

$$A' : s_1 \equiv t_1, \ldots, s_n \equiv t_n, E \overset{OINC}{\rightsquigarrow}_{\theta_2} \Box.$$

We can easily check that $\theta(\triangleq \theta_2\theta_1) = \theta_2[\mathcal{V}(G)]$ and that $|A'| < |A|$.
The case $n = 0$ is similarly proved. ∎

Lemma D.4 Let $\mathcal{R}$ be an OTRS. If there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A : (G \triangleq) c(s_1, \ldots, s_n) \equiv y, E \overset{OINC}{\rightsquigarrow}_\theta \Box$$

then there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A' : \sigma(s_1 \equiv y_1, \ldots, s_n \equiv y_n, E) \overset{OINC}{\rightsquigarrow}_\tau \Box \text{ where } \sigma = \{y \mapsto c(y_1, \ldots, y_n)\},$$

such that $\theta = \tau\sigma[\mathcal{V}(G)]$ and $|A'| < |A|$.

Proof: We first consider the case $n > 0$.
The derivation A is written as follows.

$$\begin{array}{ll} & c(s_1,\ldots,s_n) \equiv y, E \\ \overset{\text{on}}{\rightsquigarrow} & c(s_1,\ldots,s_n) = c(x_1,\ldots,x_n), y = c(y_1,\ldots,y_n), (x_1 \equiv y_1 \wedge \ldots \wedge x_n \equiv y_n) = \text{true}, E \\ \overset{OINC}{\rightsquigarrow}_{\eta} & y = c(y_1,\ldots,y_n), (s_1 \equiv y_1 \wedge \ldots \wedge s_n \equiv y_n) = \text{true}, E \\ \overset{\text{v2}}{\rightsquigarrow}_{\sigma} & \sigma(s_1 \equiv y_1 \wedge \ldots \wedge s_n \equiv y_n) = \text{true}, \sigma E \\ \overset{OINC}{\rightsquigarrow}_{\tau} & \Box \end{array}$$

By Lemma D.1, we have

$$A' : \sigma(s_1 \equiv y_1, \ldots, s_n \equiv y_n, E) \overset{OINC}{\rightsquigarrow}_{\tau} \Box.$$

We can easily check that $(\theta \triangleq)\, \tau\sigma\eta = \tau\sigma[\mathcal{V}(G)]$ and that $|A'| < |A|$.
The case $n = 0$ is similarly proved.∎

Lemma D.5 (Lifting Lemma for OINC-derivations) Let $\mathcal{R}$ be an OTRS. Suppose G is a proper goal and θ is a non-narrowable substitution such that $(\mathcal{D}\theta \cup \mathcal{V}(\mathcal{C}od\ \theta)) \cap \mathcal{V}(\text{rhs}(G)) = \emptyset$. If there exists an OINC-derivation

$$A : \theta G \overset{OINC}{\rightsquigarrow}_{\eta} \Box$$

then there exists an OINC-derivation

$$A' : G \overset{OINC}{\rightsquigarrow}_{\sigma} \Box, \text{ such that } \sigma \preceq \eta\theta[\mathcal{V}(G)]$$

and $|A'| \leq |A|$.

Proof: By the induction on the length n of the derivation A. The result holds trivially for $n = 0$. Suppose that the result holds for $n - 1$. We distinguish the following four cases by the first step of the derivation A.

(1) $G \triangleq s = x, E$
The derivation A is written as

$$A : (\theta G \triangleq)\, \theta s = x, \theta E \overset{\text{v1}}{\rightsquigarrow}_{\eta_1(=\{x \mapsto \theta s\})} \eta_1\theta E \overset{OINC}{\rightsquigarrow}_{\eta_2} \Box, \text{ where } \eta \triangleq \eta_2\eta_1[\mathcal{V}(\theta G)]. \quad (31)$$

On the other hand, we have

$$s = x, E \overset{\text{v1}}{\rightsquigarrow}_{\sigma_1(=\{x \mapsto s\})} \sigma_1 E.$$

Using the assumption on θ, we can easily see

$$\eta_1\theta = \theta\sigma_1. \quad (32)$$

By (31) and (32), we have

$$\theta\sigma_1 E \overset{OINC}{\rightsquigarrow}_{\eta_2} \Box.$$

By the assumption on θ and by the induction hypothesis, we have

$$B : \sigma_1 E \overset{OINC}{\rightsquigarrow}_{\sigma_2} \Box,$$

such that $\sigma_2 \preceq \eta_2\theta[\mathcal{V}(\sigma_1 E)]$ and $|B| \leq |A| - 1$.

We have $\sigma_2\sigma_1 \preceq \eta_2\theta\sigma_1 = \eta_2\eta_1\theta[\mathcal{V}(G)]$

. Therefore, we obtain an OINC-derivation

$$A' : s = x, E \overset{\mathrm{v1}}{\rightsquigarrow}_{\sigma_1} \sigma_1 E \overset{OINC}{\rightsquigarrow}_{\sigma_2} \Box,$$

such that $\sigma(\triangleq \sigma_2\sigma_1) \preceq \eta\theta[\mathcal{V}(G)]$ and $|A'| \leq |A|$.

(2) $G \triangleq x = t, E$ where $t \notin \mathcal{V}$ and $\theta x \equiv s$
The derivation A is written as

$$\begin{array}{lll} A : (\theta G \triangleq)\, \theta(x = t, E) & \triangleq & s = t, \theta E \\ & \overset{OINC}{\rightsquigarrow}_{\eta_1} & \eta_1\theta E \\ & \overset{OINC}{\rightsquigarrow}_{\eta_2} & \Box, \end{array} \tag{33}$$

where $\eta \triangleq \eta_2\eta_1[\mathcal{V}(G)]$.
Since s and t are non-narrowable, s and t are unifiable with the most general unifier η_1. This unification is realized by the combination of [d]-, [v1]- and [v2]-steps.
On the other hand, we have

$$x = t, E \overset{\mathrm{v2}}{\rightsquigarrow}_{\sigma_1(=\{x \mapsto t\})} \sigma_1 E. \tag{34}$$

We define a substitution τ as follows.

$$\tau z = \begin{cases} \eta_1 z & \text{if } z \in \mathcal{V}(t) \\ \eta_1\theta z & \text{otherwise.} \end{cases}$$

We know $\eta_1\theta = \tau\sigma_1$. From (33), we have an OINC-derivation

$$\tau\sigma_1 E \overset{OINC}{\rightsquigarrow}_{\eta_2} \Box.$$

Since $(\mathcal{D}\tau \cup \mathcal{V}(\mathcal{C}od\ \tau)) \cap \mathcal{V}(\mathrm{rhs}(\sigma_1 E)) = \emptyset$, by the induction hypothesis, there exists an OINC-derivation

$$B : \sigma_1 E \overset{OINC}{\rightsquigarrow}_{\sigma_2} \Box, \tag{35}$$

such that $\sigma_2 \preceq \eta_2\tau[\mathcal{V}(\sigma_1 E)]$ and $|B| \leq |A| - 1$. We have $\sigma \triangleq \sigma_2\sigma_1 \preceq \eta_2\tau\sigma_1[\mathcal{V}(G)]$, hence $\sigma \preceq \eta_2\eta_1\theta(\triangleq \eta\theta)[\mathcal{V}(G)]$.

Concatenating (34) with (35), we obtain an OINC-derivation

$$A' : x = t, E \overset{\mathrm{v2}}{\rightsquigarrow}_{\sigma_1} \sigma_1 E \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\sigma_2} \Box,$$

such that $\sigma \preceq \eta\theta[\mathcal{V}(G)]$ and $|A'| \leq |A|$.

(3) $G \triangleq f(s_1, \ldots, s_n) = t, E$ where $t \notin \mathcal{V}$ and the first step of the derivation A is [on]
By assumption, we have an OINC-derivation

$$\begin{array}{lll} A : (\theta G \triangleq)\, \theta(f(s_1, \ldots, s_n) = t, E) & \triangleq & f(\theta s_1, \ldots, \theta s_n) = t, \theta E \\ & \overset{\mathrm{on}}{\rightsquigarrow} & \theta s_1 = l_1, \ldots, \theta s_n = l_n, r = t, \theta E \\ & \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\eta_2} & \Box, \text{ and } \eta = \eta_2[\mathcal{V}(\theta G)]. \end{array}$$

In the above derivation, a rewrite rule $f(l_1, \ldots, l_n) \to r$ is employed in the [on]-step. By the assumption on θ and by the induction hypothesis, we have an OINC-derivation

$$B : (G' \triangleq) s_1 = l_1, \ldots, s_n = l_n, r = t, E \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\sigma_2} \Box,$$

such that $\sigma_2 \preceq \eta_2\theta[\mathcal{V}(G')]$ and $|B| \leq |A| - 1$. Hence, we obtain an OINC-derivation

$$A' : f(s_1, \ldots, s_n) = t, E \overset{\mathrm{on}}{\rightsquigarrow} s_1 = l_1, \ldots, s_n = l_n, r = t, E \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\sigma_2} \Box,$$

such that $\sigma(\triangleq \sigma_2) \preceq \eta\theta[\mathcal{V}(G)]$ and $|A'| \leq |A|$.

(4) $G \triangleq f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n), E$ where $t \notin \mathcal{V}$ and the first step of the derivation A is [d]
Similarly to the case (1). ∎

Lemma D.6 Let $\mathcal{R}$ be an OTRS. If there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A : (G \triangleq)\, x \equiv y, E \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\theta} \Box$$

then there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$A' : \eta E \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\sigma} \Box$$

where

$$\eta = \begin{cases} \emptyset & \text{if } x \equiv y \\ \{x \mapsto y\} & \text{otherwise} \end{cases}$$

such that $\sigma\eta \preceq \theta[\mathcal{V}(G)]$, and $|A'| < |A|$.

Proof: The proof for the case $\eta = \emptyset$ is trivial. Suppose $x \not\equiv y$. The derivation A is written as

$$A : x \equiv y, E \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\theta_1} \theta_1 E \overset{OINC}{\rightsquigarrow\!\!\!\!\rightsquigarrow}_{\theta_2} \Box,$$

where $\theta_1 = \{x \mapsto d, y \mapsto d\}$ for some ground data term d.

Let η be a substitution $\{x \mapsto y\}$ and τ a substitution $\{y \mapsto d\}$. Then, we have $\theta_1 = \tau\eta$. Hence, $\tau\eta E \overset{NC}{\rightsquigarrow}_{\theta_2} \Box$. By Lifting Lemma D.5 for OINC-derivations, we have

$$A' : \eta E \overset{NC}{\rightsquigarrow}_{\sigma} \Box, \text{ where } \sigma \preceq \theta_2\tau[\mathcal{V}(\eta E)]$$

and $|A'| \leq |A| - 1$. Furthermore, $\sigma\eta \preceq \theta_2\tau\eta = \theta_2\theta_1(\triangleq \theta)[\mathcal{V}(G)]$.∎

Theorem D.1 (Completeness of s-OINC) Let $\mathcal{R}$ be an OTRS, and G be a proper goal. If there exists an OINC-derivation with respect to $\mathcal{R}^{\equiv}$

$$G \overset{OINC}{\rightsquigarrow}_{\theta} \Box$$

then there exists an s-OINC-derivation with respect to $\mathcal{R}$

$$G \overset{s\text{-}OINC}{\rightsquigarrow}_{\sigma} \Box \text{ such that } \sigma \preceq \theta$$

Proof: By the induction on the length of the OINC-derivation. Use Lemmas D.2, D.3, D.4 and D.6. Note that we also need symmetric versions of Lemmas D.2 and D.4 to cope with the two inference rules of [ons] and of [ims], respectively. ∎