On Reducing the Search Space of Higher-Order Lazy Narrowing[⋆]

Mircea Marin[1], Tetsuo Ida[2], and Taro Suzuki[3]

[1] Institute RISC-Linz
Johannes Kepler University, A-4232 Linz, Austria
Mircea.Marin@risc.uni-linz.ac.at

[2] Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305-8573, Japan
ida@score.is.tsukuba.ac.jp

[3] School of Information Science,
JAIST Hokuriku, 923-1292, Japan
t_suzuki@jaist.ac.jp

Abstract. Higher-order lazy narrowing is a general method for solving E-unification problems in theories presented as sets of rewrite rules. In this paper we study the possibility of improving the search for normalized solutions of a higher-order lazy narrowing calculus LN. We introduce a new calculus, $\mathrm{LN_{ff}}$, obtained by extending LN and define an equation selection strategy $\mathcal{S}_n$ such that $\mathrm{LN_{ff}}$ with strategy $\mathcal{S}_n$ is complete. The main advantages of using $\mathrm{LN_{ff}}$ with strategy $\mathcal{S}_n$ instead of LN include the possibility of restricting the application of outermost narrowing at variable position, and the computation of more specific solutions because of additional inference rules for solving flex-flex equations. We also show that for orthogonal pattern rewrite systems we can adopt an eager variable elimination strategy that makes the calculus $\mathrm{LN_{ff}}$ with strategy $\mathcal{S}_n$ even more deterministic.

1 Introduction

Lazy narrowing is a method for solving E-unification problems in equational theories represented as sets of rewrite rules. It has been shown [2] that the lazy narrowing calculus forms a basis of functional logic programming. In recent years, various extensions of the lazy narrowing calculus to higher order equational theories have been proposed [3, 4, 14] in an attempt to define a suitable model for the design of an equational programming language. One such calculus is the calculus HLNC (Higher-order Lazy Narrowing Calculus) proposed by Suzuki, Nakagawa and Ida [14]. HLNC is based on the idea of combining the β-reduction

⋆ This work is partially supported by Grant-in-Aid for Scientific Research on Priority Areas "Research on the Principles for Constructing Software with Evolutionary Mechanisms", Grant-in-Aid for Scientific Research (B) 10480053, and Grant-in-Aid for Encouragement of Young Scientists 11780204, Ministry of Education, Science, Sports and Culture, Government of Japan.

of the lambda calculus and the first-order narrowing calculus LNC [6]. Based on HLNC, a programming system called CFLP (Constraint Functional Logic Programming) has been designed and implemented [5]. Independently, Prehofer studied higher-order lazy narrowing based on the higher-order rewrite system of Nipkow [9] and introduced the calculus LN [11].

Both calculi HLNC and LN are highly nondeterministic and they create a huge search space for solutions. In order to guarantee completeness, we must take into account all possible choices of (1) the equation in the current goal to be solved, (2) the inference rule of the calculus to be applied, and (3) the rewrite rule to be considered for outermost narrowing. For first-order lazy narrowing, research in reducing this non-determinism has brought important results [6, 7, 1] and gives an insight to how to eliminate some sources of the non-determinism in higher-order lazy narrowing.

In this paper we tackle the problem of reducing the non-determinism of computing substitutions that subsume all the normalized solutions of a given higher-order goal. Our main contribution in this paper is the following.

(a) We present a new higher-order lazy narrowing calculus LN_{ff} by extending LN.
(b) We introduce an equation selection strategy $\mathcal{S}_n$ that restricts the application of outermost narrowing at variable position and enables the application of the inference rules that can solve certain flex-flex equations.
(c) We prove that LN_{ff} with strategy $\mathcal{S}_n$ is complete (with respect to normalized solutions).
(d) We show that an eager variable elimination strategy makes our calculus even more deterministic for orthogonal pattern rewrite systems.

As a result we successfully reduce the search space of normalized solutions and compute more specific solutions than with LN.

The rest of this paper is structured as follows. In Sect. 2 we introduce some preliminary notions and notations. In Sect. 3 we recall some theoretical results about preunification and pattern unification. In Sect. 4 we introduce the unoriented higher-order lazy narrowing calculus LN and state the completeness result. In Sect. 5 we define our main calculus LN_{ff}. In Sect. 6 we define the equation strategy $\mathcal{S}_n$ and the class of normal LN_{ff}-refutations, and prove that all the normalized solutions of a goal are subsumed by substitutions computable with normal LN_{ff}-refutations. In Sect. 7 we extend our completeness result with an eager variable elimination strategy for solving parameter-passing equations. Finally we draw some conclusions and directions for further research.

2 Preliminaries

We employ the notation $\mathbf{a}_{m,n}$ for a sequence $a_m, a_{m+1}, \ldots, a_n$. We write $\mathbf{a}_n$ instead of $\mathbf{a}_{1,n}$. If the length of a sequence is irrelevant then we may omit the indices and write, e.g., $\mathbf{a}$ for an arbitrary (including the empty) sequence of a's. We sometimes denote an empty sequence by the symbol $\Box$.

A *term* is a simply typed lambda-term over a signature $\mathcal{F}$. We distinguish bound and free variables at the syntax level; we use uppercase letters X, Y, Z, H for free variables, lowercase letters x, y for bound variables, and letters l, r, s, t, u, v, w for terms if not stated otherwise. We extend this convention to sequences; For instance, $\mathbf{x}$ denotes a sequence of bound variables, whereas $\mathbf{s}_n$ denotes the sequence of terms $s_1, \ldots, s_n$. We denote by $\mathcal{FV}$ the set of free variables, and by $\mathcal{V}(t)$ the set of free variables occurring in a term t. A *flex term* is a term of the form $\lambda\mathbf{x}.X(\mathbf{s})$. A *rigid term* is a term which is not flex. A *pattern* is a term with the property that all its flex sub-terms are of the form $\lambda\mathbf{x}.X(\mathbf{y})$ with $\mathbf{y}$ distinct bound variables. We consider two terms s and t equal, notation $s = t$, if they are $\alpha\beta\eta$-equivalent. This notion of equality is extended to substitutions. In the sequel we represent terms in long $\beta\eta$-normal form.

Definition 1 (pattern rewrite system) A *pattern rewrite system* (PRS for short) is a set of rewrite rules of the form $f(\mathbf{l}) \to r$ with $f \in \mathcal{F}$, $f(\mathbf{l})$, r terms of the same base type, $\mathcal{V}(f(\mathbf{l})) \supseteq \mathcal{V}(r)$ and $f(\mathbf{l})$ a pattern.

Given a PRS $\mathcal{R}$, we denote by $\to$ the rewrite relation induced by $\mathcal{R}$. The relations $\to^*$, $\leftrightarrow^*$ and $\downarrow$ are defined as usual.

Definition 2 (equation) An *unoriented equation* is a pair $s \approx t$ of terms s and t of the same type. An *oriented equation* is a pair $s \rhd t$ of terms s and t of the same type. An *equation* is either an unoriented or an oriented equation. A *flex-flex equation* is an equation between flex terms. A *flex-rigid* equation is an equation between a flex and a rigid term. A *pattern equation* is an equation between patterns. A *goal* is a finite sequence of equations. A *flex-flex goal* is a goal consisting of flex-flex equations.

Let θ be a substitution. We define the *domain* of θ as $\mathcal{D}(\theta) \overset{\text{def}}{=} \{X \in \mathcal{FV} \mid X\theta \neq X\}$ and the *codomain* of θ as $\mathcal{I}(\theta) \overset{\text{def}}{=} \{X\theta \mid X \in \mathcal{D}(\theta)\}$. If θ_1, θ_2 are substitutions and V is a set of variables, we write $\theta_1 \leq \theta_2\ [V]$ if $\theta_1\delta \upharpoonright_V = \theta_2 \upharpoonright_V$ for some δ.

Definition 3 (unifier and solution) A substitution θ is a *unifier of two terms* s and t if $s\theta = t\theta$. A substitution θ is a *unifier of a goal* G if $s\theta = t\theta$ for every equation $s \approx t$ or $s \rhd t$ in G. θ is a solution of an equation $s \approx t$ if $s\theta \leftrightarrow^* t\theta$. θ is a *solution of an equation* $s \rhd t$ if $s\theta \to^* t\theta$. θ is a *solution of a goal* G if θ is a solution of all equations in G.

We will make use of the following important property of patterns. Given a PRS $\mathcal{R}$ and a substitution θ, we say that θ is *$\mathcal{R}$-normalized* if every term in $\mathcal{I}(\theta)$ is $\mathcal{R}$-normalized.

Lemma 1 If $\mathcal{R}$ is a PRS, X a free variable, $\mathbf{y}_m$ distinct bound variables and θ a substitution then $\lambda\mathbf{x}_k.X\theta(\mathbf{y}_m)$ is $\mathcal{R}$-normalized iff $X\theta$ is $\mathcal{R}$-normalized.

In the sequel, if not stated otherwise, $\mathcal{R}$ is a confluent PRS. We will often omit the prefix $\mathcal{R}$- when $\mathcal{R}$ is understood from the context. We denote by $\mathcal{R}_+$ the PRS $\mathcal{R}$ extended with the rules $\{X \approx X \to \mathtt{true}, X \rhd X \to \mathtt{true}\}$. $s \simeq t$ stands for either $s \approx t$ or $t \approx s$. We extend the notation of the binary relations between terms to componentwise relations between sequences of terms. For example, $\mathbf{s}_n \rhd \mathbf{t}_n$ stands for $s_1 \rhd t_1, \ldots, s_n \rhd t_n$. We denote sequences of equations by E, F and G, possibly subscripted.

3 Higher-Order Unification

We start our discussion with the general higher-order unification system PT.

The system PT. PT is a version of the preunification system proposed by Snyder and Gallier [12]. We omit the variable elimination rule in PT since it is not necessary for our completeness results. The inference rules for higher-order unification are:

$[\mathrm{del}]_\approx$ Deletion

$$\frac{E_1, t \approx t, E_2}{E_1, E_2}$$

$[\mathrm{dec}]_\approx$ Decomposition

$$\frac{E_1, \lambda\mathbf{x}.v(\mathbf{s}_n) \approx \lambda\mathbf{x}.v(\mathbf{t}_n), E_2}{E_1, \lambda\mathbf{x}.\mathbf{s}_n \approx \lambda\mathbf{x}.\mathbf{t}_n, E_2}$$

where $v \in \mathcal{F} \cup \{\mathbf{x}\}$.

$[\mathrm{i}]_\approx$ Imitation

$$\frac{E_1, \lambda\mathbf{x}.X(\mathbf{s}_n) \approx \lambda\mathbf{x}.f(\mathbf{t}_m), E_2}{E_1\delta, \lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta) \approx \lambda\mathbf{x}.\mathbf{t}_m\delta, E_2\delta} \quad \frac{E_1, \lambda\mathbf{x}.f(\mathbf{t}_m) \approx \lambda\mathbf{x}.X(\mathbf{s}_n), E_2}{E_1\delta, \lambda\mathbf{x}.\mathbf{t}_m\delta \approx \lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta), E_2\delta}$$

if $f \in \mathcal{F}$ where $\delta = \{X \mapsto \lambda\mathbf{x}_n.f(\mathbf{H}_m(\mathbf{x}_n))\}$.

$[\mathrm{p}]_\approx$ Projection

$$\frac{E_1, \lambda\mathbf{x}.X(\mathbf{s}_n) \approx \lambda\mathbf{x}.t, E_2}{E_1\delta, \lambda\mathbf{x}.(s_i\delta)(\mathbf{H}_p(\mathbf{s}_n\delta)) \approx \lambda\mathbf{x}.t\delta, E_2\delta} \quad \frac{E_1, \lambda\mathbf{x}.t \approx \lambda\mathbf{x}.X(\mathbf{s}_n), E_2}{E_1\delta, \lambda\mathbf{x}.t\delta \approx \lambda\mathbf{x}.(s_i\delta)(\mathbf{H}_p(\mathbf{s}_n\delta)), E_2\delta}$$

if $\lambda\mathbf{x}.t$ is rigid, where $\delta = \{X \mapsto \lambda\mathbf{x}_n.x_i(\mathbf{H}_p(\mathbf{x}_n))\}$.

In the inference rules $[\mathrm{i}]_\approx$ and $[\mathrm{p}]_\approx$, H_i are distinct fresh variables.

Notation. We write $G_1 \Rightarrow_{\alpha,\delta} G_2$ whenever $\dfrac{G_1}{G_2}$ is an instance of a PT inference rule α which computes δ. The label can be omitted if it is not relevant or it is clear from the context. We sometimes distinguish the selected equation in a PT-step by underlining it. The same conventions will be used later when we introduce other calculi.

The following completeness result is known for PT:

Theorem 1 (Completeness of PT) Let θ be a unifier of a goal G and $V \supseteq \mathcal{V}(G) \cup \mathcal{D}(\theta)$. There exist substitutions δ, θ' and a PT-derivation $G \Rightarrow_\delta^* F$ such that: (a) F is a flex-flex goal, and (b) $\delta\theta' = \theta$ $[V]$.

A proof of this theorem can be found in [11]. We note here that PT is strongly complete, i.e. Theorem 1 holds regardless of the order of selecting equations in the goal.

The system PU. It is well-known that two unifiable patterns have a unique (modulo variable renaming) most general unifier. PU is a transformation system for pattern unification. It consists of all the inference rules of the system PT and the following two inference rules:

$[\text{ffs}]_{\approx}$ Flex-flex same equation

$$\frac{E_1, \lambda\mathbf{x}.X(\mathbf{y}_m) \approx \lambda\mathbf{x}.X(\mathbf{y}'_m), E_2}{(E_1, E_2)\delta}$$

where $\delta = \{X \mapsto \lambda\mathbf{y}_m.H(\mathbf{z}_p)\}$ with $\{\mathbf{z}_p\} = \{y_i \mid 1 \leq i \leq m \text{ and } y_i = y'_i\}$.

$[\text{ffd}]_{\approx}$ Flex-flex different equation

$$\frac{E_1, \lambda\mathbf{x}.X(\mathbf{y}_m) \approx \lambda\mathbf{x}.Y(\mathbf{y}'_n), E_2}{(E_1, E_2)\delta}$$

where $\delta = \{X \mapsto \lambda\mathbf{x}_m.H(\mathbf{z}_p), Y \mapsto \lambda\mathbf{y}_n.H(\mathbf{z}_p)\}$ with $\{\mathbf{z}_p\} = \{\mathbf{y}_m\} \cap \{\mathbf{y}'_n\}$.

H is here a fresh variable, and $\mathbf{y}$, $\mathbf{y}'$ are sequences of distinct bound variables.

We denote by PU_d the system consisting of the inference rules of PU with $E_1 = \Box$. PU_d is of interest because of the following property (cf.[8]):

Theorem 2 (Completeness of PU_d) Let s, t be two unifiable patterns. Then there exists a PU_d-derivation of the form $s \approx t \Rightarrow^*_\theta \Box$ with θ a most general unifier of s and t.

4 Higher-Order Lazy Narrowing

Lazy narrowing is a goal-directed method for solving goals in equational theories presented by a confluent term rewriting system. In the first-order case a calculus called LNC [7] has been defined. LNC is sound and complete with respect to the leftmost equation selection strategy and several refinements have been proposed to reduce its non-determinism. The calculus LN introduced by Prehofer [11] is an approach for solving higher-order equations with respect to confluent PRSs. Since the calculus LN restricted to first-order terms has many similarities with LNC, one could expect that some of the deterministic refinements of LNC can be carried over to LN. Our starting point of investigation is the calculus LN defined below. It is a generalization of Prehofer's calculus LN in that we allow both unoriented and oriented equations in goals.

In order to handle oriented equations, we will introduce the following inference rules for oriented equations: $[\text{dec}]_{\rhd}$, $[\text{del}]_{\rhd}$, $[\text{i}]_{\rhd}$, $[\text{p}]_{\rhd}$, $[\text{ffs}]_{\rhd}$ and $[\text{ffd}]_{\rhd}$. They are distinguished from the corresponding rules of PU for unoriented equations by subscripting them with the equality symbol $\rhd$. Each new rule differs from the corresponding one only in that it treats oriented equations. For instance, the decomposition rule $[\text{dec}]_{\rhd}$ is the same as $[\text{dec}]_{\approx}$ except that all the occurrences of $\approx$ in the inference rule of $[\text{dec}]_{\approx}$ are replaced by $\rhd$.

The Calculus LN. LN consists of the inference rules $[\text{dec}]_{\approx}$, $[\text{del}]_{\approx}$, $[\text{i}]_{\approx}$, $[\text{p}]_{\approx}$, $[\text{dec}]_{\triangleright}$, $[\text{del}]_{\triangleright}$, $[\text{i}]_{\triangleright}$, $[\text{p}]_{\triangleright}$, plus the narrowing rules $[\text{of}]_{\simeq}$, $[\text{ov}]_{\simeq}$, $[\text{of}]_{\triangleright}$, $[\text{ov}]_{\triangleright}$ defined below:

$[\mathbf{of}]_{\simeq}$ outermost narrowing for unoriented equations

$$\frac{E_1, \lambda\mathbf{x}.f(\mathbf{s}_n) \simeq \lambda\mathbf{x}.t, E_2}{E_1, \lambda\mathbf{x}.\mathbf{s}_n \triangleright \lambda\mathbf{x}.\mathbf{l}_n, \lambda\mathbf{x}.r \approx \lambda\mathbf{x}.t, E_2}$$

$[\mathbf{ov}]_{\simeq}$ outermost narrowing at variable position for unoriented equations

$$\frac{E_1, \lambda\mathbf{x}.H(\mathbf{s}_n) \simeq \lambda\mathbf{x}.t, E_2}{E_1\delta, \lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta) \triangleright \lambda\mathbf{x}.\mathbf{l}_n, \lambda\mathbf{x}.r \approx \lambda\mathbf{x}.t\delta, E_2\delta}$$

if $\lambda\mathbf{x}.t$ is rigid, where $\delta = \{H \mapsto \lambda\mathbf{x}_n.f(\mathbf{H}_m(\mathbf{x}_n))\}$.

$[\mathbf{of}]_{\triangleright}$ outermost narrowing for oriented equations

$$\frac{E_1, \lambda\mathbf{x}.f(\mathbf{s}_n) \triangleright \lambda\mathbf{x}.t, E_2}{E_1, \lambda\mathbf{x}.\mathbf{s}_n \triangleright \lambda\mathbf{x}.\mathbf{l}_n, \lambda\mathbf{x}.r \triangleright \lambda\mathbf{x}.t, E_2}$$

$[\mathbf{ov}]_{\triangleright}$ outermost narrowing at variable position for oriented equations

$$\frac{E_1, \lambda\mathbf{x}.H(\mathbf{s}_n) \triangleright \lambda\mathbf{x}.t, E_2}{E_1\delta, \lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta) \triangleright \lambda\mathbf{x}.\mathbf{l}_n, \lambda\mathbf{x}.r \triangleright \lambda\mathbf{x}.t\delta, E_2\delta}$$

if $\lambda\mathbf{x}.t$ is rigid, where $\delta = \{H \mapsto \lambda\mathbf{x}_n.f(\mathbf{H}_m(\mathbf{x}_n))\}$.

In these inference rules $\mathbf{H}_m$ are distinct fresh variables and $f(\mathbf{l}_m) \to r$ is a fresh variant of an $\mathbf{x}$-lifted rule (see [11] for the definition of $\mathbf{x}$-lifting). We write [of] to denote $[\text{of}]_{\simeq}$ or $[\text{of}]_{\triangleright}$, and [ov] to denote $[\text{ov}]_{\simeq}$ or $[\text{ov}]_{\triangleright}$. [o] denotes [of] or [ov]. We use letter π to denote an LN-step and Π to denote an LN-derivation.

It can be easily verified that the calculus LN is sound, i.e. if $G_1 \Rightarrow_{\alpha,\delta} G_2$ and θ is a solution of G_2 then $\delta\theta$ is a solution of G_1.

In the sequel we use $\{\ldots\}$ to denote multisets and $>_{mul}$ for the multiset ordering on sets of non-negative integers. The expression $|e|$ may denote: (a) the length of e if e is a derivation, or (b) the size of e if e is an equation or a term.

The use of LN in solving higher-order goals is justified by the following completeness result:

Theorem 3 (Completeness of LN) Let $\mathcal{R}$ be a confluent PRS and G a goal with solution θ. Then there exists an LN-derivation $\Pi : G \Rightarrow^*_{\delta} F$ such that $\delta \leq \theta\ [\mathcal{V}(G)]$ and F is a flex-flex goal.

Proof. (Sketch) The proof of this theorem is along the following lines. We first define a suitable well-founded ordering on some structures that encode the fact that a substitution is a solution of a goal.

Definition 4 Let $G = \mathbf{e}_n$ be a goal. We define Repr(G) as the set of triples of the form $\langle G, \theta, \mathbf{R}_n \rangle$ with θ solution of G and $\mathbf{R}_n$ a sequence of reduction derivations of the form $R_j : e_j\theta \to^*_{\mathcal{R}_+} \texttt{true}$.

On such triples we define the following well-founded orderings:

– $\langle \mathbf{e}_n, \theta, \mathbf{R}_n \rangle >_A \langle \mathbf{e}'_m, \theta', \mathbf{R}'_m \rangle$ if $\{|R_1|, \ldots, |R_n|\} >_{mul} \{|R'_1|, \ldots, |R'_m|\}$,
– $\langle \mathbf{e}_n, \theta, \mathbf{R}_n \rangle >_B \langle \mathbf{e}'_m, \theta', \mathbf{R}'_m \rangle$ if $\{|t| \mid t \in \mathcal{I}(\theta\lceil_{\mathcal{V}(\mathbf{e}_n)})\} >_{mul} \{|t'| \mid t' \in \mathcal{I}(\theta'\lceil_{\mathcal{V}(\mathbf{e}'_m)})\}$,
– $\langle \mathbf{e}_n, \theta, \mathbf{R}_n \rangle >_C \langle \mathbf{e}'_m, \theta', \mathbf{R}'_m \rangle$ if $\{|e_1|, \ldots, |e_n|\} >_{mul} \{|e'_1|, \ldots, |e'_m|\}$,
– $\succ$ is the lexicographic combination of $>_A, >_B, >_C$.

Next we prove the following lemma, which is also used in the proof of Lemma 4:

Lemma 2 Let $G_0 = E_1, e, E_2$ be a goal with solution θ_0 and non-flex-flex equation e. Assume $V \supseteq \mathcal{V}(G_0) \cup \mathcal{D}(\theta_0)$. Then for any triple $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \in \mathrm{Repr}(G_0)$ there exists an LN-step $\pi : G_0 = E_1, \underline{e}, E_2 \Rightarrow_{\alpha,\delta} G_1$ and a triple $\langle G_1, \theta_1, \mathbf{R}^1 \rangle \in \mathrm{Repr}(G_1)$ such that: (a) $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \succ \langle G_1, \theta_1, \mathbf{R}^1 \rangle$, and (b) $\theta_0 = \delta\theta_1$ $[V]$.

We omit the proof of this lemma since it is similar to the proof of Theorem 6.1.1 in [11].
Finally, we note that repeated applications of Lemma 2 starting from a triple $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \in \mathrm{Repr}(G_0)$ produces the desired LN-derivation. $\square$

Remark 1 1. The substitution δ in Theorem 3 is a pattern substitution[1], since it is a composition of pattern substitutions.
2. LN is strongly complete, namely it does not depend on the order of selecting the non-flex-flex equations in the goal.

5 The Calculus LN$_{\mathbf{ff}}$

In this section we introduce our main calculus.

The calculus LN$_{\mathbf{ff}}$. LN$_{\mathrm{ff}}$ consists of the inference rules of LN and the rules $[\mathrm{ffs}]_\approx$, $[\mathrm{ffs}]_\triangleright$, $[\mathrm{ffd}]_\approx$, $[\mathrm{ffd}]_\triangleright$.

In the sequel we omit the subscripts $\approx$ and $\triangleright$ of inference rules $\alpha_\approx$ and $\alpha_\triangleright$ and write α when we treat the inference rules collectively.

We now obtain a more powerful calculus since all the rules for pattern unification are available. Unfortunately the calculus LN$_{\mathrm{ff}}$ is no longer strongly complete as we can see from the example below:

Example 1. Let $\mathcal{R} = \{f(X, a) \to b\}$ and the goal

$$G_0 = f(X, a) \triangleright Z, Z \approx Y, b \triangleright Y, Z \triangleright f(X, a)$$

with solution $\theta = \{Z \mapsto f(X, a), Y \mapsto b\}$. If we select the equation $Z \approx Y$ to solve the goal G_0 then the only applicable rule is $[\mathrm{ffd}]_\approx$. Hence we have the following derivation:

$$G_0 \Rightarrow_{[\mathrm{ffd}]_\approx, \{Z \mapsto H, Y \mapsto H\}} G_1 = f(X, a) \triangleright H, b \triangleright H, H \triangleright f(X, a)$$

It can be easily seen that the goal G_1 has no solution. Hence, LN$_{\mathrm{ff}}$ is not strongly complete. $\square$

[1] A substitution θ is a *pattern substitution* if every term in $\mathcal{I}(\theta)$ is a pattern.

Note that the previous example does not refute the strong completeness of LN because no rule of LN can act on the flex-flex equation $Z \approx Y$.

In the following we show that there exists an equation selection strategy $\mathcal{S}_n$ with respect to which the calculus $\mathrm{LN_{ff}}$ is complete. Actually we show a stronger result: by adopting the calculus $\mathrm{LN_{ff}}$ with strategy $\mathcal{S}_n$ we achieve two important desiderata:

1. We eliminate the nondeterminism due to the choice of the equation in a goal to be solved next,
2. We restrict the application of outermost narrowing at variable position; as a consequence, a smaller search space for solutions is created.

The steps of our investigation can be summarized as follows:

1. We first observe that if we know that the solution of a goal is normalized with respect to certain variables then we can apply the rules [ffs] and [ffd] to certain pattern equations and safely avoid the application of [ov] to certain flex-rigid or rigid-flex equations (Lemma 3 and Lemma 4).
2. We identify sufficient conditions which guarantee that the solution of a goal is normalized with respect to certain variables (Lemma 5).
3. Based on 2., we define a strategy $\mathcal{S}_n$. For the calculus $\mathrm{LN_{ff}}$ with the strategy $\mathcal{S}_n$ we identify the class of normal $\mathrm{LN_{ff}}$-refutations and prove that any normalized solution of a goal G is subsumed by a substitution which is computable with a normal $\mathrm{LN_{ff}}$-refutation that starts from G (Theorem 4).

Before starting to describe in detail the steps mentioned above, we note some similarities between the calculi LNC and $\mathrm{LN_{ff}}$ when we restrict ourselves to first-order terms:

1. It can be verified that $\mathrm{LN_{ff}}$ subsumes LNC with the rule [v] restricted to equations between variables.
2. A [v]-step in LNC can be simulated in $\mathrm{LN_{ff}}$ by a sequence of [i]-, [p]-,[ffs]-, and [ffd]-steps. Since LNC with leftmost equation selection strategy $\mathcal{S}_{\mathrm{left}}$ is complete [7], we conclude that the calculus $\mathrm{LN_{ff}}$ without [ov]-rules is complete if we adopt the strategy $\mathcal{S}_{\mathrm{left}}$.
3. Middeldorp *et al.* [7] conjecture that LNC is complete with respect to any strategy that never selects descendants of an equation created by an outermost narrowing step before all descendants of the corresponding parameter-passing equations created in that step have been selected. If this conjecture holds then we can replace $\mathcal{S}_{\mathrm{left}}$ with such a strategy and retain the completeness of $\mathrm{LN_{ff}}$ without [ov]-rules.

Note that the substitution θ in Example 1 is not normalized. We noticed that the normalization of substitutions restricted to the so called *critical variables* is crucial to restore completeness of $\mathrm{LN_{ff}}$.

Definition 5 (critical variable) The set $\mathcal{V}_c(e)$ of *critical variables* of an equation e is $\mathcal{V}(s) \cup \mathcal{V}(t)$ if $e = s \approx t$ and $\mathcal{V}(s)$ if $e = s \triangleright t$.

We first prove the following technical lemma.

Lemma 3 If $G_0 = E_1, \underline{e}, E_2 \Rightarrow_{[\mathrm{ff}],\delta_0} G_1$, and $\theta_0 \restriction_{\mathcal{V}_c(e)}$ is normalized then for any $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \in \mathrm{Repr}(G_0)$ there exists a solution θ_1 of G_1 and $\langle G_1, \theta_1, \mathbf{R}^1 \rangle \in \mathrm{Repr}(G_1)$ such that: (a) $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \succ \langle G_1, \theta_1, \mathbf{R}^1 \rangle$, and (b) $\theta_0 = \delta_0 \theta_1$.

Proof. Let $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \in \mathrm{Repr}(G_0)$. The proof is by case distinction on the shape of e.
(i) If $e = \lambda \mathbf{x}.X(\mathbf{y}_m) \approx \lambda \mathbf{x}.Y(\mathbf{y}'_n)$ then $\lambda \mathbf{x}.X(\mathbf{y}_m)\theta_0 \downarrow \lambda \mathbf{x}.Y(\mathbf{y}'_n)\theta_0$ because θ_0 is a solution of e. In this case $\mathcal{V}_c(e) = \{X, Y\}$ and therefore the terms $X\theta_0$ and $Y\theta_0$ are normalized. By Lemma 1 the terms $\lambda \mathbf{x}.X(\mathbf{y}_m)\theta_0$ and $\lambda \mathbf{x}.Y(\mathbf{y}'_n)\theta_0$ are also normalized. Hence the equality $\lambda \mathbf{x}.X(\mathbf{y}_m)\theta_0 = \lambda \mathbf{x}.Y(\mathbf{y}'_n)\theta_0$ holds, i.e. θ_0 is a unifier of $\lambda \mathbf{x}.X(\mathbf{y}_m)$, $\lambda \mathbf{x}.Y(\mathbf{y}'_n)$. Since δ_0 is a most general unifier of $\lambda \mathbf{x}.X(\mathbf{y}_m)$ and $\lambda \mathbf{x}.Y(\mathbf{y}'_n)$ there exists a solution θ_1 of G_1 such that $\theta_0 = \delta\theta_1$. The construction of $\mathbf{R}^1$ such that $\langle G_0, \theta_0, \mathbf{R}^0 \rangle =_A \langle G_1, \theta_1, \mathbf{R}^1 \rangle$ and $\langle G_0, \theta_0, \mathbf{R}^0 \rangle >_B \langle G_1, \theta_1, \mathbf{R}^1 \rangle$ is straightforward. Therefore, (a) holds as well.
(ii) The case when $e = \lambda \mathbf{x}.X(\mathbf{y}_m) \approx \lambda \mathbf{x}.X(\mathbf{y}'_m)$ can be proved in a similar way.
(iii) Let $e = \lambda \mathbf{x}.X(\mathbf{y}_m) \rhd \lambda \mathbf{x}.Y(\mathbf{y}'_n)$. Because θ_0 is a solution of $\lambda \mathbf{x}.X(\mathbf{y}_m) \rhd \lambda \mathbf{x}.Y(\mathbf{y}'_n)$, we have $\lambda \mathbf{x}.X(\mathbf{y}_m)\theta_0 \rightarrow^* \lambda \mathbf{x}.Y(\mathbf{y}'_n)\theta_0$. In this case we have $\mathcal{V}_c(e) = \{X\}$ and thus $X\theta_0$ is normalized. By Lemma 1 the term $\lambda \mathbf{x}.X(\mathbf{y}_m)\theta_0$ is also normalized. Therefore θ_0 is a unifier of the terms $\lambda \mathbf{x}.X(\mathbf{y}_m)\theta_0$ and $\lambda \mathbf{x}.Y(\mathbf{y}'_n)\theta_0$. Since δ_0 is a most general unifier of $\lambda \mathbf{x}.X(\mathbf{y}_m)$ and $\lambda \mathbf{x}.Y(\mathbf{y}'_n)$ there exists a solution θ_1 of G_1 such that $\theta_0 = \delta\theta_1$. The construction of $\mathbf{R}^1$ such that $\langle G_0, \theta_0, \mathbf{R}^0 \rangle =_A \langle G_1, \theta_1, \mathbf{R}^1 \rangle$ and $\langle G_0, \theta_0, \mathbf{R}^0 \rangle >_B \langle G_1, \theta_1, \mathbf{R}^1 \rangle$ is straightforward. Therefore, (a) holds as well.
(iv) The case when $e = \lambda \mathbf{x}.X(\mathbf{y}_m) \rhd \lambda \mathbf{x}.X(\mathbf{y}'_m)$ can be proved in a similar way.

We next investigate restrictions under which the rules $[\mathrm{ov}]_{\approx}$, $[\mathrm{ov}]_{\rhd}$ can be eliminated without losing the completeness of $\mathrm{LN_{ff}}$. The next example illustrates that in general we can not drop [ov] without loss of completeness.

Example 2. Consider the PRS $\mathcal{R} = \{f(g(X)) \rightarrow X\}$ and the goal $Z(g(a)) \approx a$. Obviously the normalized answer $\{Z \mapsto \lambda x.f(x)\}$ could not be computed by $\mathrm{LN_{ff}}$ if we dropped the [ov]-rule. □

Lemma 4 Let θ_0 be a solution of a goal $G_0 = E_1, e, E_2$ with $e = \lambda \mathbf{x}.X(\mathbf{y}) \simeq t$ or $e = \lambda \mathbf{x}.X(\mathbf{y}) \rhd t$, where t is a rigid term and $\lambda \mathbf{x}.X(\mathbf{y})$ is a pattern. Assume $V \supseteq \mathcal{V}(G_0) \cup \mathcal{D}(\theta_0)$. If $X\theta_0$ is normalized then for any $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \in \mathrm{Repr}(G_0)$ there exists an $\mathrm{LN_{ff}}$-step $\pi : G_0 = E_1, \underline{e}, E_2 \Rightarrow_{\alpha,\delta_0} G_1$ with $\alpha \neq [\mathrm{ov}]$ such that: (a) $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \succ \langle G_1, \theta_1, \mathbf{R}^1 \rangle$, and (b) $\theta_0 = \delta_0\theta_1$ $[V]$.

Proof. By Lemma 2, there exists an LN-step π which satisfies conditions (a) and (b). If $\alpha = [\mathrm{ov}]$ then the term $\lambda \mathbf{x}.X(\mathbf{y})\theta_0$ is reducible. This implies that $X\theta_0$ is reducible, which contradicts our hypothesis. Hence $\alpha \neq [\mathrm{ov}]$. □

6 Normal $\mathrm{LN_{ff}}$-refutations

We will use the results of Lemmata 3 and 4 in order to define a suitable equation selection strategy $\mathcal{S}_n$ with respect to which the calculus $\mathrm{LN_{ff}}$ is complete.

The success of defining such a strategy depends on the possibility to determine whether the solution of an equation is normalized with respect to certain variables. In the sequel we look for such normalization criteria.

The notions of *immediate linear descendant* (ILD for short) and of *immediate descendant* of the selected equation in an LN$_{\mathrm{ff}}$-step are defined as shown in the table below. In the table, the symbol $=^?$ stands for either $\approx$ or $\triangleright$ (but the same in the same row). The superscripts 1 and 2 on [i] and [p] distinguish the first and the second case of the corresponding inference rule.

rule	ILD	immediate descendant
[of]	$\lambda\mathbf{x}.r =^? \lambda\mathbf{x}.t$	$\lambda\mathbf{x}.\mathbf{s}_n \triangleright \lambda\mathbf{x}.\mathbf{l}_n, \lambda\mathbf{x}.r =^? \lambda\mathbf{x}.t$
[ov]	$\lambda\mathbf{x}.r =^? \lambda\mathbf{x}.t$	$\lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta) \triangleright \lambda\mathbf{x}.\mathbf{l}_n, \lambda\mathbf{x}.r =^? \lambda\mathbf{x}.t\delta$
[dec]	$\lambda\mathbf{x}.\mathbf{s}_n =^? \lambda\mathbf{x}.\mathbf{t}_n$	$\lambda\mathbf{x}.\mathbf{s}_n =^? \lambda\mathbf{x}.\mathbf{t}_n$
[i]^1	$\lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta) =^? \lambda\mathbf{x}.\mathbf{t}_m\delta$	$\lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta) =^? \lambda\mathbf{x}.\mathbf{t}_m\delta$
[i]^2	$\lambda\mathbf{x}.\mathbf{t}_m\delta =^? \lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta)$	$\lambda\mathbf{x}.\mathbf{t}_m\delta =^? \lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta)$
[p]^1	$\lambda\mathbf{x}.(s_i\delta)(\mathbf{H}_p(\mathbf{s}_n\delta)) =^? \lambda\mathbf{x}.v(\mathbf{t}_m\delta)$	$\lambda\mathbf{x}.(s_i\delta)(\mathbf{H}_p(\mathbf{s}_n\delta)) =^? \lambda\mathbf{x}.v(\mathbf{t}_m\delta)$
[p]^2	$\lambda\mathbf{x}.v(\mathbf{t}_m\delta) =^? \lambda\mathbf{x}.(s_i\delta)(\mathbf{H}_p(\mathbf{s}_n\delta))$	$\lambda\mathbf{x}.v(\mathbf{t}_m\delta) =^? \lambda\mathbf{x}.(s_i\delta)(\mathbf{H}_p(\mathbf{s}_n\delta))$
[del] [ffs], [ffd]	-	-

Descendants and ILDs of non-selected equations are defined as expected. The equations $\lambda\mathbf{x}.\mathbf{s}_n \triangleright \lambda\mathbf{x}.\mathbf{l}_n$ and $\lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta) \triangleright \lambda\mathbf{x}.\mathbf{l}_n$ created in an [of]-step and [ov]-step respectively, are called *parameter passing equations* created by that step.

The notion of *descendant* is obtained from that of immediate descendant by reflexivity and transitivity. The *ancestor* relation is defined as the inverse of the descendant relation.

Definition 6 (precursor) Let $\Pi : G_0 \Rightarrow^* E_1, e_1, E_2, e_2, E_3$ be an LN$_{\mathrm{ff}}$-derivation. e_1 is a *precursor* of e_2 in Π if there exists an equation e which is subjected to an [o]-step π in Π such that (a) e_1 is a descendant of a parameter passing-equation created by π, and (b) e_2 is a descendant of the ILD of e in π.

Given an LN$_{\mathrm{ff}}$-derivation $\Pi : G_0 \Rightarrow^N G_N = E_1, e, E_2$, we denote by $\mathrm{prec}_\Pi(e)$ the sub-sequence of equations of G_N that are precursors of e in Π.

Definition 7 (regular transformation) Let G_0, G_1 be goals with $\langle G_0, \theta_0, \mathbf{R}^0\rangle \in \mathrm{Repr}(G_0)$, $G_0 = E_1, e, E_2$ and $\langle G_1, \theta_1, \mathbf{R}^1\rangle \in \mathrm{Repr}(G_1)$. Assume $V \supseteq \mathcal{V}(G_0) \cup \mathcal{D}(\theta_0)$. A transformation step $\langle G_0, \theta_0, \mathbf{R}^0\rangle \Rightarrow \langle G_1, \theta_1, \mathbf{R}^1\rangle$ is *regular* if:

- e is a non-flex-flex equation and there exists an LN-step $\pi : E_1, \underline{e}, E_2 \Rightarrow_{\alpha,\delta} G_1$ such that the conditions (a) and (b) of Lemma 2 hold, or
- $\theta_0 \restriction_{\mathcal{V}_c(e)}$ is normalized, $G_0 = E_1, \underline{e}, E_2 \Rightarrow_{[\mathrm{ff}],\delta} G_1$ and the conditions (a) and (b) of Lemma 3 hold.

Lemma 5 Let G_0 be a goal with normalized solution θ_0 and $\langle G_0, \theta_0, \mathbf{R}^0\rangle \in \mathrm{Repr}(G_0)$. If $\langle G_0, \theta_0, \mathbf{R}^0\rangle \Rightarrow \ldots \Rightarrow \langle G_N, \theta_N, \mathbf{R}^N\rangle$ is a sequence of regular transformation steps and $\Pi : G_0 \Rightarrow_{\delta_0} G_1 \Rightarrow_{\delta_1} \cdots \Rightarrow_{\delta_{N-1}} G_N$ is the corresponding LN$_{\mathrm{ff}}$-derivation then for any $e \in G_N$ with $\mathrm{prec}_\Pi(e) = \Box$ we have that $\theta_N \restriction_{\mathcal{V}_c(e)}$ is normalized.

Proof. Let e_i be the ancestor of e in G_i and $\gamma_i = \delta_i\delta_{i+1}\ldots\delta_{N-1}$ $(0 \leq i \leq N)$. We prove a slightly stronger result: $\theta_N\lceil_{\mathcal{V}_c(e_i\gamma_i)}$ is normalized for any $0 \leq i \leq N$. It is easy to see that this implies the normalization of $\theta_N\lceil_{\mathcal{V}_c(e)}$.

We first introduce the notion of [o]-ancestor. We say that an ancestor e' of e is an [o]-*ancestor* of e if we have

$$\Pi : G \Rightarrow^* E_1, \underline{e'}, E_2 \Rightarrow_{[\mathrm{o}],\sigma} E_1\sigma, E_3, e'', E_2\sigma \Rightarrow^* E_1', e, E_2'.$$

(That is, an [o]-step is applied to e' and e descends from the ILD of e'.) We prove by induction on i $(0 \leq i \leq N)$ that $\theta_N \lceil_{\mathcal{V}_c(e_i\gamma_i)}$ is normalized. Let π_i be the i-th step of Π.

If $i = 0$ then $\theta_N \lceil_{\mathcal{V}_c(e_0\gamma_0)}$ is normalized because $\gamma_0\theta_N = \theta_0$ $[\mathcal{V}_c(e_0)]$ and θ_0 is normalized.

We next show that $\theta_N \lceil_{\mathcal{V}_c(e_{i+1}\gamma_{i+1})}$ is normalized if $\theta_N \lceil_{\mathcal{V}_c(e_i\gamma_i)}$ is normalized.

Suppose e_i is not an [o]-ancestor. We show $\mathcal{V}_c(e_{i+1}) \subseteq \mathcal{V}_c(e_i\delta_i)$ by the following case distinction.

(a) π_i is an [o]-step. Since e_i is not an [o]-ancestor, we have that e_{i+1} is a parameter-passing equation created by the i-th step of Π and therefore $\mathcal{V}_c(e_{i+1}) \subseteq \mathcal{V}_c(e_i\delta_i)$.

(b) π_i is not an [o]-step. If e_i is unoriented then $\mathcal{V}_c(e_{i+1}) = \mathcal{V}(e_{i+1}) \subseteq \mathcal{V}(e_i\delta_i) = \mathcal{V}_c(e_i\delta_i)$. If e_i is of the form $s \triangleright t$ then it can be shown by case distinction on π_i that $\mathcal{V}_c(e_{i+1}) \subseteq \mathcal{V}(s\delta) = \mathcal{V}_c(e_i\delta)$.

The induction hypothesis yields the normalization of $\theta_N \lceil_{\mathcal{V}_c(e_i\delta_i\gamma_{i+1})}$. Hence the above inclusion implies the normalization of $\theta_N \lceil_{\mathcal{V}_c(e_{i+1}\gamma_{i+1})}$.

Suppose e_i is an [o]-ancestor. Then $e_i = \lambda\mathbf{x}.h(\mathbf{s}_n) =^? \lambda\mathbf{x}.t$ or $e_i = \lambda\mathbf{x}.t \approx \lambda\mathbf{x}.h(\mathbf{s}_n)$ and Π is of the form

$$\begin{aligned} \Pi : G_0 &\Rightarrow^i_{\delta_0\ldots\delta_{i-1}} && G_i = E_1, \underline{e}, E_2 \\ &\Rightarrow_{[\mathrm{o}],f(\mathbf{l}_m)\to r,\delta_i} && G_{i+1} = E_1\delta_i, \lambda\mathbf{x}.\mathbf{w}_m \triangleright \lambda\mathbf{x}.\mathbf{l}_m, \lambda\mathbf{x}.r =^? \lambda\mathbf{x}.t\delta_i, E_2\delta_i \\ &\Rightarrow^*_{\delta_{i+1}\ldots\delta_{N-1}} && G_N = E_1', e, E_2'. \end{aligned}$$

Since $\mathrm{prec}_{\Pi}(e) = \Box$ we have

$$(\lambda\mathbf{x}.\mathbf{w}_m \triangleright \lambda\mathbf{x}.\mathbf{l}_m)\delta_{i+1}\ldots\delta_{N-1} \to^* \top.$$

We show that the following relation holds:

$$e\gamma_i \to^* (\lambda\mathbf{x}.r =^? \lambda\mathbf{x}.t\delta_i)\gamma_{i+1}. \tag{1}$$

If $h = f$ then we have $\delta_i = \varepsilon$, $n = m$ and $\lambda\mathbf{x}.\mathbf{w}_n = \lambda\mathbf{x}.\mathbf{s}_n$. Since $\delta_{i+1}\ldots\delta_{N-1}$ is a solution of $\lambda\mathbf{x}.\mathbf{s}_n \triangleright \lambda\mathbf{x}.\mathbf{l}_n$ we learn that $\lambda\mathbf{x}.s_j\gamma_{i+1} \to^* \mathbf{x}.l_j\gamma_{i+1}$ for $1 \leq j \leq n$ and therefore $\lambda\mathbf{x}.h(\mathbf{s}_n)\gamma_i \to^* \lambda\mathbf{x}.f(\mathbf{l}_n\gamma_{i+1}) \to \lambda\mathbf{x}.r\gamma_{i+1}$.

Otherwise $h \in \mathcal{FV}$ and in this case we have $\delta_i = \{h \mapsto \lambda\mathbf{x}_n.f(\mathbf{H}_m(\mathbf{x}_n))\}$ and $\lambda\mathbf{x}.\mathbf{w}_m = \lambda\mathbf{x}.\mathbf{H}_m(\mathbf{s}_n\delta_i)$. Because γ_{i+1} is a solution of $\lambda\mathbf{x}.\mathbf{w}_m \triangleright \lambda\mathbf{x}.\mathbf{l}_m$ we have $\lambda\mathbf{x}.h(\mathbf{s}_n)\gamma_i =_\beta \lambda\mathbf{x}.f(\mathbf{H}_m(\mathbf{s}_n\delta_i)\gamma_{i+1}) \to^* \lambda\mathbf{x}.f(\mathbf{l}_m\gamma_{i+1}) = \lambda\mathbf{x}.f(\mathbf{l}_m)\gamma_{i+1} \to \lambda\mathbf{x}.r\gamma_{i+1}$.

Thus in both situations we have $\lambda\mathbf{x}.h(\mathbf{s}_n)\gamma_i \to^* \lambda\mathbf{x}.r\gamma_{i+1}$ and hence (1) holds.

It follows that $\mathcal{V}_c(e_{i+1}\gamma_{i+1}) \subseteq \mathcal{V}_c(e_i\gamma_i)$. Since $\theta_N \restriction_{\mathcal{V}_c(e_i\gamma_i)}$ is normalized, the substitution $\theta_N \restriction_{\mathcal{V}_c(e_{i+1}\gamma_{i+1})}$ is normalized as well. $\Box$

We are ready now to define our equation selection strategy for LN$_{\text{ff}}$.

Definition 8 (strategy $\mathcal{S}_n$) An LN$_{\text{ff}}$-derivation Π respects the strategy $\mathcal{S}_n$ if for any subderivation $\Pi' : G_0 \Rightarrow^* G_m = E_1, \underline{e}, E_2$ of Π we have:

(c1) If [ffs] or [ffd] is applicable to e then $\text{prec}_{\Pi'}(e) = \Box$.
(c2) If e is of the form $\lambda\mathbf{x}.X(\mathbf{s}) \simeq t$ or $\lambda\mathbf{x}.X(\mathbf{s}) \triangleright t$, with t rigid and $\text{prec}_{\Pi'}(e) \neq \Box$ then all the selectable equations of G_m satisfy condition (c2).

Condition (c1) enables the selection of an equation e to which [ffs]- or [ffd]-rules are applicable only when e has no precursor. Condition (c2) enables the selection of equations to which [ov]-rules are applicable only when there is no other choice.

Definition 9 (normal LN$_{\text{ff}}$-refutation) An LN$_{\text{ff}}$-derivation $\Pi : G_0 \Rightarrow^* F$ is a *normal LN$_{ff}$-refutation* if

1. Π respects $\mathcal{S}_n$, and
2. F does not contain equations which are selectable with $\mathcal{S}_n$.

Completeness. We will prove that LN$_{\text{ff}}$ with strategy $\mathcal{S}_n$ is complete with respect to normalized solutions.

Theorem 4 For any normalized solution θ_0 of a goal G_0 such that $\mathcal{V}(G_0) \cup \mathcal{D}(\theta) \subseteq V$ there exists a normal LN$_{\text{ff}}$-refutation $\Pi : G_0 \Rightarrow^*_\delta F$ with $\delta \leq \theta\ [V]$.

Proof. Assume $\langle G_0, \theta_0, \mathbf{R}^0 \rangle \in \text{Repr}(G_0)$. Let

$$A : \langle G_0, \theta_0, \mathbf{R}^0 \rangle \Rrightarrow \langle G_1, \theta_1, \mathbf{R}^1 \rangle \Rrightarrow \ldots \Rrightarrow \langle G_N, \theta_N, \mathbf{R}^N \rangle$$

be a maximal sequence of transformation steps starting from $\langle G_1, \theta_1, \mathbf{R}^1 \rangle$ such that the corresponding LN$_{\text{ff}}$-step $\pi_i : G_i \Rightarrow G_{i+1}$ satisfies strategy $\mathcal{S}_n$. The existence of the sequence A is a consequence of the fact that $\Rrightarrow \subseteq \succ$ and $\succ$ is terminating. It suffices to show that the LN$_{\text{ff}}$-derivation $\pi_0 \cdots \pi_{N-1}$ is a normal LN$_{\text{ff}}$-refutation, which is obvious. $\Box$

7 Eager variable elimination

We address here the eager variable elimination problem for LN$_{\text{ff}}$ with respect to normal LN$_{\text{ff}}$-refutations. In the first-order case this problem is related to the possibility to apply the variable elimination rule prior to other applicable inference rules. In [6] it is shown that an eager variable elimination strategy for parameter-passing equations is complete for left-linear confluent TRSs.

The proof is mainly due to the standardization theorem for left-linear confluent TRSs, which roughly states that if a term s is reachable to a term t then an outside-in reduction derivation from s to t exists.

We will generalize the first-order eager variable elimination strategy to LN$_{\text{ff}}$ with the help of outside-in reduction derivations.

Definition 10 An LN$_{\text{ff}}$ refutation Π *respects the eager variable elimination strategy* if $[\mathrm{o}]_{\rhd}$ is never applied to rigid-flex equations of the form $\lambda\mathbf{x}.s \rhd \lambda\mathbf{x}.X(\mathbf{t})$ in Π.

We say that *LN$_{\text{ff}}$ with eager variable elimination strategy is complete for a class of PRSs* if for any goal G with a solution θ there exists an LN$_{\text{ff}}$ refutation $G \Rightarrow^*_{\sigma} \square$ that respects the eager variable elimination strategy with $\sigma \leq_{\mathcal{R}} \theta$, when $\mathcal{R}$ belongs to the class of PRSs.

The notion of outside-in reduction derivations for orthogonal PRSs is carried over from that of first order TRSs [13] except for the definition of anti-standard pairs stated below.

Definition 11 (outside-in reduction derivation for orthogonal PRSs) An $\mathcal{R}_+$-reduction derivation by an orthogonal PRS is called *outside-in* if every subderivation $e \rightarrow_p e_0 \rightarrow_{p_1} \cdots \rightarrow_{p_n} e_n \rightarrow_{q,l\rightarrow r} e'$ satisfies the following condition: if $p > q > \varepsilon$ and all p_i $(1 \leq i \leq n)$ are disjoint from p then p/q is above or disjoint from any free variable position in l. Here p/q is a position satisfying $p = q \cdot (p/q)$.

The only difference from the first-order case given in [6] is disregard for the bound variables below the free variables. The definition above states that the subterms headed by free variables in a higher-order pattern, called the *binding holes* after Oostrom [10], are regarded as mere variables.

In [10] Oostrom claimed that the following statement holds, which allows us to concentrate on only the outside-in reduction derivations.

Theorem 5 For any rewrite derivation $s \rightarrow^*_{\mathcal{R}} t$ by an orthogonal PRS $\mathcal{R}$, there exists an outside-in rewrite derivation from s to t. $\square$

We follow the same line of reasoning as in [7] to show that the eager variable elimination strategy for parameter-passing equations preserves completeness of LN$_{\text{ff}}$: first we introduce a property of reduction derivations which holds for any outside-in reduction derivation starting from a goal consisting of unoriented equations. Next we show that regular transformations preserve this property. This result motivates the possibility to inhibit the application of [o] to equations of the form $\lambda\mathbf{x}.s \rhd \lambda\mathbf{x}.X(\mathbf{y})$.

First we introduce a class of restricted outside-in reduction derivations.

Definition 12 Let $\mathcal{R}$ be an orthogonal PRS and $s \rhd t\theta \rightarrow^* \texttt{true}$ an outside-in $\mathcal{R}_+$-reduction derivation. Then we say the derivation has property $\mathcal{P}_{HO}$ if every reduction step in it satisfies the following condition: if a position $1 \cdot p$ is rewritten in the reduction step and later steps except the final step do not take place above $1 \cdot p$, then p is above or disjoint from any free variable position in t.

Let $\langle G, \theta, \mathbf{R}\rangle \in \mathrm{Repr}(G)$ such that every reduction derivation in R is outside-in. It is obvious that all the outside-in reduction derivations in $\mathbf{R}$ have property $\mathcal{P}_{HO}$ if the goal G consists only of unoriented equations. The following lemma establishes the preservation of the property $\mathcal{P}_{HO}$ during regular transformations.

Lemma 6 Let $\langle G, \theta, \mathbf{R}\rangle \in \mathrm{Repr}(G)$ and suppose $\langle G', \theta', \mathbf{R'}\rangle$ is obtained by a regular transformation from $\langle G, \theta, \mathbf{R}\rangle$. If $\mathbf{R}$ only consists of outside-in derivations with property $\mathcal{P}_{HO}$, then the same holds for $\mathbf{R}'$. □

The proof is done by an easy but tedious case analysis on the regular transformations.

Theorem 6 $\mathrm{LN}_{\mathrm{ff}}$ with eager variable elimination strategy is complete for orthogonal PRSs with respect to normalized solutions for goals consisting of unoriented equations.

Proof. Let $\langle G, \theta, \mathbf{R}\rangle \in \mathrm{Repr}(G)$, where G consists of unoriented equations, θ is a normalized substitution, and $\mathbf{R}$ contains an outside-in reduction derivation R. Note that R has no extended anti-standard pairs. For any $\langle G', \theta', \mathbf{R}'\rangle$ obtained by the repeated applications of regular transformations, from Lemma 6 we learn that if G' includes an equation e of the form $\lambda\mathbf{x}.s\theta' \triangleright \lambda\mathbf{x}.X\theta'(\mathbf{t}\theta')$ then the corresponding reduction derivation in $\mathbf{R}'$ should be $e \rightarrow_{\varepsilon}$ `true`; otherwise $\mathbf{R}'$ has an extended anti-standard pair. The regular transformation applied to the equation e never produces an [o]-step. □

Note that once we get outside-in reduction derivations, we no longer need the restriction on terms to patterns. The restriction, however, becomes crucial for further eager variable elimination. For instance, $[\mathrm{ov}]_{\triangleright}$ may be applied to a flex-rigid parameter-passing equation with non-pattern in the left-hand side. On the other hand, from Lemma 5 and Lemma 6 we infer that normal $\mathrm{LN}_{\mathrm{ff}}$-refutations never contain applications of $[\mathrm{ov}]_{\triangleright}$-steps to flex-rigid parameter-passing equations with pattern in the left-hand side provided we are interested only in normalized solutions and the precursors of the equation were completely solved. In practice, we expect that most terms occurring in $\mathrm{LN}_{\mathrm{ff}}$ derivations are patterns and hence $[\mathrm{ov}]_{\triangleright}$ is rarely employed.

The above remark assures that normal $\mathrm{LN}_{\mathrm{ff}}$ refutations that respect eager variable elimination strategy enjoy a generalization of the eager variable elimination strategy in the first order case. Recall that eager variable elimination is also applicable to parameter-passing equations of the form $X \triangleright t$ in the first order case [6]. Since X is obviously a pattern, we can prohibit the application of [ov] to this equation.

8 Conclusions and Further Research

We identified an equation selection strategy class with respect to which the calculus $\mathrm{LN}_{\mathrm{ff}}$ is complete. Note that $\mathcal{S}_n$ does not identify the equation which must be selected next but specifies a condition which must be satisfied by the selected equation. Therefore it is possible to define more specific equation selection strategies for $\mathrm{LN}_{\mathrm{ff}}$. Such a strategy is the one that always selects an equation which satisfies $\mathcal{S}_n$ and has small nondeterminism due to the selection of the applicable inference rule. Also, our result confirms the validity of the conjecture of Middeldorp which we mentioned in Sect. 5.

In Sect. 7 we proved that if we restrict ourselves to an orthogonal PRS then we can make the calculus $\mathrm{LN}_{\mathrm{ff}}$ even more deterministic by adopting an eager-variable elimination strategy.

We mention here the result of Prehofer [11] about the possibility to completely drop the [ov]-rules from LN if we restrict to convergent PRS. The proof is based on the existence of innermost derivations for such rewriting systems. However, the termination condition is very strong in practice.

References

1. M. Hamada, T. Ida. *Deterministic and Non-deterministic Lazy Conditional Narrowing and their Implementations.* Transactions of Information Processing Society of Japan, Vol. 39, No. 3, pp. 656–663, March 1998.
2. M. Hanus. *The Integration of Functions into Logic Programming: From Theory to Practice.* Journal of Logic Programming, 19&20:583-628, 1994.
3. K. Nakahara, A. Middeldorp, T. Ida. A Complete Narrowing Calculus for Higher-order Functional Logic Programming. In *Proceedings of the Seventh International Conference on Programming Languages: Implementations, Logics and Programs 95 (PLILP'95), LNCS* 982, 97-114, 1995.
4. M. Marin, A. Middeldorp, T. Ida, T. Yanagi. *LNCA: A Lazy Narrowing Calculus for Applicative Term Rewriting Systems.* Technical Report ISE-TR-99-158, University of Tsukuba, 1999.
5. M. Marin, T. Ida, W. Schreiner. CFLP: a Mathematica Implementation of a Distributed Constraint Solving System. *Third International Mathematica Symposium (IMS'99),* Hagenberg, Austria, August 23-25, 1999. Computational Mechanics Publications, WIT Press, Southampton, UK.
6. A. Middeldorp, S. Okui. A Deterministic Lazy Narrowing Calculus. *Journal of Symbolic Computation* 25(6), pp. 733-757, 1998.
7. A. Middeldorp, S. Okui, T. Ida. Lazy Narrowing: Strong Completeness and Eager Variable Elimination. *Theoretical Computer Science* 167(1,2), pp. 95-130, 1996.
8. T. Nipkow. Functional Unification of Higher-order Patterns. In *Proceedings of 8th IEEE Symposium on Logic in Computer Science*, pp. 64-74, 1993.
9. T. Nipkow, C. Prehofer. Higher-Order Rewriting and Equational Reasoning. In *Automated Deduction - A Basis for Applications.* Volume I. Kluwer, 1998, 399-430.
10. V. van Oostrom. Higher-order Families. In *International Conference on Rewriting Techniques and Applications '96, LNCS*, 1996.
11. C. Prehofer. *Solving Higher-Order Equations. From Logic to Programming.* Birkhäuser Boston, 1998.
12. W. Snyder, J. Gallier. Higher-order unification revisited: Complete sets of transformations. *Journal of Symbolic Computation*, 8:101-140, 1989.
13. T. Suzuki. Standardization Theorem Revisited. In *Proceedings of 5th International Conference, ALP'96, LNCS* 1139, pp.122-134, 1996.
14. T. Suzuki, K. Nakagawa, T. Ida. Higher-Order Lazy Narrowing Calculus: A Computation Model for a Higher-order Functional Logic Language. In *Proceedings of Sixth International Joint Conference, ALP '97 - HOA '97, LNCS* 1298, pp. 99–113, September 1997, Southampton.