

**Lazy Narrowing: Strong Completeness
and Eager Variable Elimination
(Extended Abstract)**

Satoshi Okui [†]
Aart Middeldorp ^{††}
Tetsuo Ida [‡]

University of Tsukuba
Tsukuba 305 Japan

December, 1994

ISE-TR-94-114

[†] Doctoral Program in Engineering, e-mail: okui@softlab.is.tsukuba.ac.jp

^{††} Institute of Information Sciences and Electronics, e-mail: ami@softlab.is.tsukuba.ac.jp

[‡] Institute of Information Sciences and Electronics, e-mail: ida@softlab.is.tsukuba.ac.jp

Lazy Narrowing: Strong Completeness and Eager Variable Elimination (Extended Abstract)

Satoshi Okui Aart Middeldorp Tetsuo Ida

Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305, Japan

ABSTRACT

Narrowing is an important method for solving unification problems in equational theories that are presented by confluent term rewriting systems. Because narrowing is a rather complicated operation, several authors studied calculi in which narrowing is replaced by more simple inference rules. This paper is concerned with one such calculus. Contrary to what has been stated in the literature, we show that the calculus lacks strong completeness, so selection functions to cut down the search space are not applicable. We prove completeness of the calculus and we establish an interesting connection between its strong completeness and the completeness of basic narrowing. We also address the eager variable elimination problem. It is known that many redundant derivations can be avoided if the variable elimination rule, one of the inference rules of our calculus, is given precedence over the other inference rules. We prove the completeness of a restricted variant of eager variable elimination in the case of orthogonal term rewriting systems.

1. Introduction

E -unification—solving equations modulo some equational theory E —is a fundamental technique in automated reasoning. Narrowing ([19, 4, 11]) is a general E -unification procedure for equational theories that are presented by confluent term rewriting systems (TRSs for short). Narrowing is the computational mechanism of many functional-logic programming languages (see Hanus [7] for a recent survey on the integration of functional and logic programming). It is well-known that narrowing is complete with respect to normalizable solutions. Completeness means that for every solution to a given equation, a more general solution can be found by narrowing. If we extend narrowing to goals consisting of several equations, we obtain *strong* completeness. This means that we don't lose completeness when we restrict applications of the narrowing rule to a single equation in each goal.

Since narrowing is not easily implemented, several authors studied calculi consisting of a small number of more elementary inference rules that simulate

narrowing (e.g. [15, 8, 9, 13, 20, 6]). In this paper we are concerned with a subset (actually the specialization to confluent TRSs) of the calculus TRANS proposed by Hölldobler [9]. We call this calculus *lazy narrowing calculus* (LNC for short). Because the purpose of LNC is to simulate narrowing by more elementary inference rules, it is natural to expect that LNC inherits strong completeness from narrowing, and indeed this is stated by Hölldobler (Corollary 7.3.9 in [9]). We show however that LNC lacks strong completeness.

An important improvement over narrowing is *basic* narrowing (Hullot [11]). In basic narrowing narrowing steps are never applied to (sub)terms introduced by previous narrowing substitutions, resulting in a significant reduction of the search space. In this paper we establish an interesting connection between LNC and basic narrowing: we show that LNC is strongly complete whenever basic narrowing is complete. The latter is known for complete (i.e., confluent and terminating) TRSs (Hullot [11]). Other sufficient conditions are right-linearity and orthogonality (Middeldorp and Hamoen [16]). So LNC is strongly complete for these three classes of TRSs. We prove completeness of LNC for the general case of confluent TRSs. In the literature completeness of LNC-like calculi is proved under the additional termination assumption. Without this assumption the completeness proof is significantly more involved.

It is known that LNC-like calculi generate many derivations which produce the same solutions (up to subsumption). Martelli *et al.* [15, 13] and Hölldobler [9], among others, pointed out that many of these redundant derivations can be avoided by giving the variable elimination rule, one of the inference rules of LNC-like calculi, precedence over the other inference rules. The problem whether this strategy is complete or not is called the *eager variable elimination* problem in [9, 20]. Martelli *et al.* stated in [13] that this is easily shown in the case of terminating (and confluent) TRSs, but Snyder questions the validity of this claim in his monograph [20] on *E*-unification. We address the eager variable elimination problem for non-terminating TRSs. We prove completeness of a slightly restricted version of eager variable elimination in the case of orthogonal TRSs. To this end we simplify and extend the main result of You [22] concerning the completeness of *outer* narrowing for orthogonal *constructor-based* TRSs.

The remainder of the paper is organized as follows. In a preliminary section we introduce narrowing and state the relevant completeness results. The narrowing calculus that we are interested in—LNC—is defined in Section 3. In that section we also show that LNC is not strongly complete. In Section 4 we establish the connection between the strong completeness of LNC and the completeness of basic narrowing. We prove the completeness of LNC for general confluent systems in Section 5. Section 6 is concerned with the eager variable elimination problem. In the final section we give suggestions for further research.

Due to lack of space, the present paper doesn't contain any proof details. All proofs can be found in the full version [17].

2. Preliminaries

We assume the reader's familiarity with the basic notions of term rewriting ([2, 12]). We use the position formalism to denote subterm occurrences. The set of positions in a term is denoted by $\mathcal{Pos}(t)$. If $p \in \mathcal{Pos}(t)$ then $t|_p$ denotes the subterm of t at position p and $t[s]_p$ denotes the term that is obtained from t by replacing the subterm at position p by the term s . The set $\mathcal{Pos}(t)$ is partitioned into $\mathcal{Pos}_V(t)$ and $\mathcal{Pos}_F(t)$ as follows: $\mathcal{Pos}_V(t) = \{p \in \mathcal{Pos}(t) \mid t|_p \in V\}$ and $\mathcal{Pos}_F(t) = \mathcal{Pos}(t) - \mathcal{Pos}_V(t)$. Positions are partially ordered by the prefix ordering \leq , i.e., $p \leq q$ if there exists a (necessarily unique) r such that $p \cdot r = q$. In that case we define $q \setminus p$ as the position r . We write $p < q$ if $p \leq q$ and $p \neq q$. If neither $p \leq q$ nor $q \leq p$, we write $p \perp q$. The domain $\mathcal{D}(\theta)$ of a substitution θ is the set $\{x \in V \mid x\theta \neq x\}$. As usual, this set is required to be finite. We denote the set $\bigcup_{x \in \mathcal{D}(\theta)} V(x\theta)$ by $\mathcal{I}(\theta)$. We distinguish a nullary function symbol **true** and a binary function symbol \doteq , written in infix notation. A term of the form $s \doteq t$, where neither s nor t contains any occurrences of \doteq and **true**, is called an *equation*. The term **true** is also viewed as an equation. The extension of a TRS \mathcal{R} with the rewrite rule $x \doteq x \rightarrow \text{true}$ is denoted by \mathcal{R}_+ . A *goal* is a sequence of equations. We use \top as a generic notation for goals containing only equations **true**. Let G be a goal and θ a substitution. If there exists a rewrite sequence $G\theta \rightarrow_{\mathcal{R}_+}^* \top$, we write $\mathcal{R} \vdash G\theta$ and we say that θ is an $(\mathcal{R}\text{-})$ solution of G . Narrowing is formulated as the following inference rule:

$$\frac{G_1, e, G_2}{(G_1, e[r]_p, G_2)\theta} \quad \begin{array}{l} \text{if there exist a fresh variant } l \rightarrow r \text{ of a rewrite rule} \\ \text{in } \mathcal{R}_+, \text{ a position } p \in \mathcal{Pos}_F(e), \text{ and a most general} \\ \text{unifier } \theta \text{ of } e|_p \text{ and } l. \end{array}$$

In the above situation we write $G_1, e, G_2 \rightsquigarrow_{\theta, p, l \rightarrow r, e} (G_1, e[r]_p, G_2)\theta$. This is called an *NC-step* (NC stands for narrowing calculus). Subscripts will be omitted when they are clear from the context or irrelevant. A sequence $G_1 \rightsquigarrow_{\theta_1} \dots \rightsquigarrow_{\theta_{n-1}} G_n$ of NC-steps is called an *NC-derivation* and abbreviated to $G_1 \rightsquigarrow_{\theta}^* G_n$ where $\theta = \theta_1 \dots \theta_{n-1}$. We use the symbol Π (and its derivatives) to denote NC-derivations. An NC-derivation which ends in \top is called an *NC-refutation*. The following completeness result is essentially due to Hullot [11].

THEOREM 2.1. *Let \mathcal{R} be a confluent TRS. If $\mathcal{R} \vdash G\theta$ and $\theta|_{\text{Var}(G)}$ is normalized then there exists an NC-refutation $G \rightsquigarrow_{\theta'}^* \top$ such that $\theta' \leq \theta|_{\text{Var}(G)}$. \square*

There are three sources of non-determinism in NC: the choice of the equation e , the choice of the subterm $e|_p$, and the choice of the rewrite rule $l \rightarrow r$. The last two choices are *don't know* non-deterministic, meaning that in general all possible choices have to be considered in order to guarantee completeness. The choice of the equations e is *don't care* non-deterministic, because of the *strong completeness* of NC. Strong completeness means completeness independent of selection functions. A *selection function* is mapping that assigns to every goal G different from \top an equation $e \in G$ different from **true**. An example of a selection function is S_{left} which always returns the leftmost equation different

from **true**. We say that an NC-derivation Π *respects* a selection function S if the selected equation in every step $G_1 \rightsquigarrow G_2$ of Π coincides with $S(G_1)$. Now strong completeness of NC is formulated as follows.

THEOREM 2.2. *Let \mathcal{R} be a confluent TRS and S a selection function. If $\mathcal{R} \vdash G\theta$ and $\theta|_{\text{Var}(G)}$ is normalized then there exists an NC-refutation $G \rightsquigarrow_{\theta}^* \top$ respecting S such that $\theta' \leq \theta|_{\text{Var}(G)}$. \square*

3. Lazy Narrowing Calculus

Calculi in which the narrowing inference rule is replaced by a small number of more primitive operations are comprehensively examined by Hölldobler in his thesis [9] and Snyder in his monograph [20]. The calculus that we investigate in this paper is the specialization of Hölldobler's calculus TRANS, which is defined for general equational systems and based on paramodulation, to (confluent) TRSs and narrowing.

DEFINITION 3.1. Let \mathcal{R} be a TRS. The *lazy narrowing calculus*, LNC for short, consists of the following five inference rules:

[on] *outermost narrowing*

$$\frac{G_1, f(s_1, \dots, s_n) \doteq t, G_2}{G_1, s_1 \doteq l_1, \dots, s_n \doteq l_n, r \doteq t, G_2}$$

if there exists a fresh variant $f(l_1, \dots, l_n) \rightarrow r$ of a rewrite rule in \mathcal{R} ,

[im] *imitation*

$$\frac{G_1, f(s_1, \dots, s_n) \doteq x, G_2}{(G_1, s_1 \doteq x_1, \dots, s_n \doteq x_n, G_2)\theta}$$

if $\theta = \{x \mapsto f(x_1, \dots, x_n)\}$ with x_1, \dots, x_n fresh variables,

[d] *decomposition*

$$\frac{G_1, f(s_1, \dots, s_n) \doteq f(t_1, \dots, t_n), G_2}{G_1, s_1 \doteq t_1, \dots, s_n \doteq t_n, G_2},$$

[v] *variable elimination*

$$\frac{G_1, x \doteq t, G_2}{(G_1, G_2)\theta}$$

if $x \notin \text{Var}(t)$ and $\theta = \{x \mapsto t\}$,

[t] *removal of trivial equations*

$$\frac{G_1, x \doteq x, G_2}{G_1, G_2}.$$

Here $s \doteq t$ stands for $s \doteq t$ or $t \doteq s$.

The variable elimination rule [v] is different from the one of Martelli *et al.* [15, 13] in that we don't keep the solved equation $x \doteq t$ around. The rules [v], [d], and [t] constitute the syntactic unification algorithm of Martelli and Montanari

[14]. We refer to these three rules as UC, which stands for *unification calculus*. Because syntactic unification is performed by UC, the rewrite rule $x \doteq x \rightarrow \text{true}$ is no longer used in LNC. As a consequence, we may assume that the symbol **true** doesn't occur in LNC-goals.

Contrary to usual narrowing, the outermost narrowing rule $[on]$ generates new *parameter-passing* equations $s_1 \doteq l_1, \dots, s_n \doteq l_n$ besides the *body* equation $r \doteq t$. These parameter-passing equations must eventually be solved in order to obtain a refutation, but we don't require that they are solved right away. That is the reason why we call the calculus *lazy*. We introduce some useful notations relating to the calculus LNC. If G and G' are the upper and lower goal in the inference rule $[\alpha]$ ($\alpha \in \{on, im, d, v, t\}$), we write $G \Rightarrow_{[\alpha]} G'$. This is called an LNC-step. The applied rewrite rule or substitution may be supplied as subscript, that is, we will write things like $G \Rightarrow_{[on], l \mapsto r} G'$ and $G \Rightarrow_{[im], \theta} G'$. LNC-derivations are defined as in the case of NC. An LNC-refutation is an LNC-derivation ending in the empty goal \square . UC-steps, derivations, and refutations are defined in exactly the same way.

Because the purpose of LNC is to simulate narrowing, it is natural to expect that LNC inherits strong completeness from NC. Indeed, Hölldobler [9, Corollary 7.3.9] states the strong completeness of LNC for confluent TRSs with respect to normalizable solutions. However, this does not hold.

COUNTEREXAMPLE 3.2. Consider the TRS $\mathcal{R} = \{f(x) \rightarrow g(h(x), x), g(x, x) \rightarrow a, b \rightarrow h(b)\}$ and the goal $G = f(b) \doteq a$. Confluence of \mathcal{R} can be proved by a routine induction argument on the structure of terms. The (normalized) empty substitution ε is a solution of G because $f(b) \doteq a \rightarrow_{\mathcal{R}} g(h(b), b) \doteq a \rightarrow_{\mathcal{R}} g(h(b), h(b)) \doteq a \rightarrow_{\mathcal{R}} a \doteq a \rightarrow_{\mathcal{R}_+} \text{true}$. Consider the selection function \mathcal{S}_{right} that selects the rightmost equation in every goal. There is essentially only one LNC-derivation issued from G respecting \mathcal{S}_{right} :

$$\begin{array}{ll}
f(b) \doteq a & \Rightarrow_{[on], f(x) \mapsto g(h(x), x)} b \doteq x, g(h(x), x) \doteq a \\
& \Rightarrow_{[on], g(x_1, x_1) \mapsto a} b \doteq x, h(x) \doteq x_1, x \doteq x_1, a \doteq a \\
& \Rightarrow_{[d]} b \doteq x, h(x) \doteq x_1, x \doteq x_1 \\
& \Rightarrow_{[v], \{x_1 \mapsto x\}} b \doteq x, h(x) \doteq x \\
& \Rightarrow_{[im], \{x \mapsto h(x_2)\}} b \doteq h(x_2), h(x_2) \doteq x_2 \\
& \Rightarrow_{[im], \{x_2 \mapsto h(x_3)\}} \dots
\end{array}$$

This is clearly not a refutation. (The alternative binding $\{x \mapsto x_1\}$ in the $\Rightarrow_{[v]}$ -step results in a renaming of the above LNC-derivation.) Hence LNC is *not* strongly complete.

This counterexample doesn't refute the completeness of LNC. The goal $f(b) \doteq a$ can be solved, for instance, by adopting the selection function \mathcal{S}_{left} .

In Section 5 we show that LNC is complete in the general case of confluent TRSs and normalized solutions. In the next section we present sufficient conditions for the strong completeness of LNC, which turns out to be a simpler than proving completeness.

4. Restoring Strong Completeness

In Middeldorp and Hamoen [16] it is shown that basic narrowing is not able to solve the goal $f(b) \doteq a$ with respect to the TRS \mathcal{R} of Counterexample 3.2. This suggests a connection between strong completeness of LNC and completeness of basic NC. In this section we prove that LNC is strongly complete whenever basic NC is complete.

The basis of our proof is the specialization of the transformation process used by Hölldobler in his proof of the (strong) completeness of TRANS. First we formalize the intuitively clear propagation of equations along NC-derivations.

DEFINITION 4.1. Let $G \rightsquigarrow_{\theta,p,l \rightarrow r} G'$ be an NC-step and e an equation in G . If e is the selected equation in this step, then e is narrowed into the equation $e[r]_p \theta$ in G' . In this case we say that $e[r]_p \theta$ is the *descendant* of e in G' . Otherwise, e is simply instantiated to the equation $e\theta$ in G' and we call $e\theta$ the descendant of e . The notion of descendant extends to NC-derivations in the obvious way.

Observe that in an NC-refutation $G \rightsquigarrow^* \top$ every equation $e \in G$ has exactly one descendant **true** in \top . We now introduce four transformation steps on NC-refutations. The first one corresponds to Proposition 7.3.4 in Hölldobler [9].

LEMMA 4.2. Let $\Pi: G_1, s \doteq t, G_2 \rightsquigarrow_{\theta}^* \top$ be an NC-refutation with the property that narrowing is applied to a descendant of $s \doteq t$ at position 1. Let V be a finite set of variables such that $\text{Var}(G_1, s \doteq t, G_2) \subseteq V$. If $l \rightarrow r$ is the applied rewrite rule in the first such step then there exists an NC-refutation $\phi_{[\text{on}]}(\Pi): G_1, s \doteq l, r \doteq t, G_2 \rightsquigarrow_{\theta'}^* \top$ such that $\theta = \theta' [V]$. \square

The second transformation step corresponds to Proposition 7.3.3 in [9].

LEMMA 4.3. Let e be the equation $f(s_1, \dots, s_n) \doteq f(t_1, \dots, t_n)$. Let $\Pi: G_1, e, G_2 \rightsquigarrow_{\theta}^* \top$ be an NC-refutation with the property that narrowing is never applied to a descendant of e at position 1 or 2. Let V be a finite set of variables that includes $\text{Var}(G_1, e, G_2)$. There exists an NC-refutation $\phi_{[\text{d}]}(\Pi): G_1, s_1 \doteq t_1, \dots, s_n \doteq t_n, G_2 \rightsquigarrow_{\theta'}^* \top$ such that $\theta = \theta' [V]$. \square

The third transformation step corresponds to Corollary 7.3.5 in Hölldobler [9]. This corollary is an immediate consequence of Hölldobler's lifting lemma for reflection, instantiation, and paramodulation (Lemma 6.2.6 in [9]). This easy proof does not work in our case since narrowing, unlike paramodulation, cannot be applied at variable positions. Nevertheless, we can repeat the proof of the lifting lemma (for NC) to obtain the validity of the third transformation step.

LEMMA 4.4. Let $\Pi: G \rightsquigarrow_{\theta}^* \top$ be an NC-refutation with the property that $x\theta = f(t_1, \dots, t_n)$ for some $x \in \text{Var}(G)$ and let V be a finite set of variables such that $\text{Var}(G) \subseteq V$. Let $\gamma = \{x \mapsto f(x_1, \dots, x_n)\}$ with $x_1, \dots, x_n \notin V$. There exists an NC-refutation $\phi_{[\text{im}]}(\Pi): G\gamma \rightsquigarrow_{\theta'}^* \top$ which employs the same rewrite rules at the same positions in the corresponding equations of the goals in Π such that $\gamma\theta' = \theta [V]$. \square

The fourth and final transformation step is presented in the following lemma.

LEMMA 4.5. *For every NC-refutation $\Pi: G_1, s \doteq t, G_2 \rightsquigarrow_{\theta_1, \epsilon} (G_1, \text{true}, G_2)\theta_1 \rightsquigarrow_{\theta_2}^* \top$ there exists an NC-refutation $\phi_{UC}(\Pi): (G_1, G_2)\theta_1 \rightsquigarrow_{\theta_2}^* \top$. \square*

The idea now is to repeatedly apply the above transformation steps to a given NC-refutation, connecting the initial goals of (some of) the resulting NC-refutations by LNC-steps, until we reach the empty goal. In order to guarantee termination of this process, we need a well-founded order on NC-refutations that is compatible with the four transformation steps.

DEFINITION 4.6. The *depth* $|t|$ of a term t is inductively defined as follows: $|t| = 1$ if t is a variable and $|t| = 1 + \max\{|t_1|, \dots, |t_n| \mid 1 \leq i \leq n\}$ if $t = f(t_1, \dots, t_n)$. The *complexity* $|\Pi|$ of an NC-refutation $\Pi: G \rightsquigarrow_{\theta}^* \top$ is defined as the triple (n, M, s) where n is the number of applications of narrowing in Π at non-root positions (so the number of steps that do not use the rewrite rule $x \doteq x \rightarrow \text{true}$), M is the multiset $\{|x_1\theta|, \dots, |x_m\theta| \mid \{x_1, \dots, x_m\}$ is the *multiset* of variables occurring in $G\}$, and s is the number of occurrences of symbols different from \doteq and true in G . We define a partial order \gg on NC-refutations as follows: $\Pi_1 \gg \Pi_2$ if $|\Pi_1| \text{ lex } (>, >_{mul}, >) |\Pi_2|$. Here $\text{lex } (>, >_{mul}, >)$ denotes the lexicographic product of $>$ (the standard order on \mathbb{N}), $>_{mul}$ (the multiset extension [3] of $>$), and $>$.

LEMMA 4.7. *The partial order \gg is a well-founded order on NC-refutations.* \square

Our complexity measure on NC-refutations is different from the one in Hölldobler [9, p. 188]. Since we are concerned with one-directional term rewriting and narrowing (as opposed to bi-directional equational reasoning and paramodulation in [9]), our simpler definition suffices. The next lemma states that \gg is compatible with the transformation steps defined above.

LEMMA 4.8. *Let Π be an NC-refutation. We have $\Pi \gg \phi_{[on]}(\Pi)$, $\phi_{[d]}(\Pi)$, $\phi_{[im]}(\Pi)$, $\phi_{UC}(\Pi)$ whenever the latter are defined.* \square

In the case of $\phi_{[on]}$ and $\phi_{[d]}$ this is easily verified. The validity of Lemma 4.8 for the transformation steps $\phi_{[im]}$ and ϕ_{UC} requires more effort. The following example illustrates how the above results are used to transform NC-refutations into LNC-refutations.

EXAMPLE 4.9. Consider the TRS $\mathcal{R} = \{f(g(y)) \rightarrow y\}$ and the NC-refutation

$$\Pi_1: g(f(x)) \doteq x \rightsquigarrow_{\{x \mapsto g(y)\}} g(y) \doteq g(y) \rightsquigarrow_{\epsilon} \text{true}.$$

In Π_1 the variable x is bound to $g(y)$, so the complexity of Π_1 is $(1, \{2, 2\}, 4)$. Transformation steps $\phi_{[on]}$, $\phi_{[d]}$, and ϕ_{UC} are not applicable to Π_1 . Hence we try $\phi_{[im]}$. This yields the NC-refutation

$$\Pi_2 = \phi_{[im]}(\Pi_1): g(f(g(x_1))) \doteq g(x_1) \rightsquigarrow_{\{x_1 \mapsto y\}} g(y) \doteq g(y) \rightsquigarrow_{\epsilon} \text{true}$$

which has complexity $(1, \{1, 1\}, 6)$. Next we apply $\phi_{[d]}$. This gives the NC-refutation

$$\Pi_3 = \phi_{[d]}(\Pi_2): f(g(x_1)) \doteq x_1 \rightsquigarrow_{\{x_1 \mapsto y\}} y \doteq y \rightsquigarrow_{\epsilon} \text{true}$$

with complexity $(1, \{1, 1\}, 4)$. Observe that the initial goal of Π_1 is transformed into the initial goal of Π_3 by the single LNC-step $g(f(x)) \doteq x \Rightarrow_{[im], \{x \mapsto g(x_1)\}} f(g(x_1)) \doteq x_1$. In Π_3 narrowing is applied to the initial equation at position 1. This calls for the transformation step $\phi_{[on]}$:

$$\Pi_4 = \phi_{[on]}(\Pi_3): f(g(x_1)) \doteq f(g(y)), y \doteq x_1 \rightsquigarrow_{\{x_1 \mapsto y\}} \text{true}, y \doteq y \rightsquigarrow_{\epsilon} \top.$$

NC-refutation Π_4 has complexity $(0, \{1, 1, 1, 1\}, 8)$. If we apply $\phi_{[d]}$ to Π_4 , we obtain the NC-refutation

$$\Pi_5 = \phi_{[d]}(\Pi_4): g(x_1) \doteq g(y), y \doteq x_1 \rightsquigarrow_{\{x_1 \mapsto y\}} \text{true}, y \doteq y \rightsquigarrow_{\epsilon} \top$$

with complexity $(0, \{1, 1, 1, 1\}, 6)$. The initial goals of Π_3 and Π_5 are connected by an $\Rightarrow_{[on]}$ -step: $f(g(x_1)) \doteq x_1 \Rightarrow_{[on]} g(x_1) \doteq g(y), y \doteq x_1$. In the first step of Π_5 narrowing is applied at the root position of the selected equation $g(x_1) \doteq g(y)$. Hence we use ϕ_{UC} , yielding the NC-refutation

$$\Pi_6 = \phi_{UC}(\Pi_5): y \doteq y \rightsquigarrow_{\epsilon} \text{true}$$

with complexity $(0, \{1, 1\}, 2)$. The initial goals of Π_5 and Π_6 are connected by the UC-derivation $g(x_1) \doteq g(y), y \doteq x_1 \Rightarrow_{[d]} x_1 \doteq y, y \doteq x_1 \Rightarrow_{[v], \{x_1 \mapsto y\}} y \doteq y$. Another application of ϕ_{UC} results in the empty NC-refutation

$$\Pi_7 = \phi_{UC}(\Pi_6): \square$$

which has complexity $(0, \emptyset, 0)$. Clearly $y \doteq y \Rightarrow_{[t]} \square$. Concatenating the various LNC-sequences yields an LNC-refutation $g(f(x)) \doteq x \Rightarrow_{\theta}^* \square$ whose substitution θ satisfies $x\theta = g(y)$.

Unfortunately, the simulation of NC by LNC illustrated above doesn't always work, as shown in the following example.

EXAMPLE 4.10. Consider the TRS $\mathcal{R} = \{f(x) \rightarrow x, a \rightarrow b, b \rightarrow g(b)\}$ and the NC-refutation $\Pi_{fail}: f(a) \doteq g(a) \rightsquigarrow f(a) \doteq g(b) \rightsquigarrow a \doteq g(b) \rightsquigarrow b \doteq g(b) \rightsquigarrow g(b) \doteq g(b) \rightsquigarrow \text{true}$. Because we apply narrowing at position 1 in the descendant $f(a) \doteq g(b)$ of the initial equation $f(a) \doteq g(a)$, using the rewrite rule $f(x) \rightarrow x$, we transform Π_{fail} using $\phi_{[on]}$ and $\phi_{[d]}$. This yields the NC-refutation $\phi_{[d]}(\phi_{[on]}(\Pi_{fail})): a \doteq x, x \doteq g(a) \rightsquigarrow a \doteq x, x \doteq g(b) \rightsquigarrow a \doteq g(b) \rightsquigarrow b \doteq g(b) \rightsquigarrow g(b) \doteq g(b) \rightsquigarrow \text{true}$. Observe that the initial goals of Π_{fail} and $\phi_{[d]}(\phi_{[on]}(\Pi_{fail}))$ are connected by $\Rightarrow_{[on]}$. Since in the refutation $\phi_{[d]}(\phi_{[on]}(\Pi_{fail}))$ narrowing is applied at position 1 in the descendant $a \doteq g(b)$ of the selected equation $x \doteq g(a)$ in the initial goal $a \doteq x, x \doteq g(a)$, we would like to use once more the transformation steps $\phi_{[on]}$ and $\phi_{[d]}$. This is however impossible since the subterm of $x \doteq g(a)$ at position 1 is a variable.

The reason why Π_{fail} cannot be transformed to an LNC-refutation by the transformation steps in this section is that in $\phi_{[d]}(\phi_{[on]}(\Pi_{fail}))$ narrowing is applied to a subterm introduced by a previous narrowing substitution. One might be tempted to think that this problem cannot occur if we restrict ourselves to normalized solutions. This is not true, however, because Π_{fail} computes the empty substitution ε , which is clearly normalized, but $\phi_{[d]}(\phi_{[on]}(\Pi_{fail}))$ computes the non-normalized solution $\{x \mapsto a\}$. So the transformation steps do not preserve *normalization* of the computed NC-solutions (restricted to the variables in the initial goal). However, it turns out that *basicness* ([11, 16]) is preserved. This is one of the two key observations to the connection between strong completeness of LNC and completeness of basic NC.

LEMMA 4.11. *Let Π be a basic NC-refutation. The NC-refutations $\phi_{[on]}(\Pi)$, $\phi_{[d]}(\Pi)$, $\phi_{[im]}(\Pi)$, and $\phi_{UC}(\Pi)$ are basic whenever they are defined. \square*

The other key observation is that for basic NC, strong completeness and completeness coincide.

LEMMA 4.12. *Let \mathcal{S} be a selection function. For every basic NC-refutation $\Pi: G \rightsquigarrow_{\theta}^* \top$ there exists a basic NC-refutation $\Pi_{\mathcal{S}}: G \rightsquigarrow_{\theta}^* \top$ respecting \mathcal{S} with the same complexity. \square*

We are now ready to present the main result of this section.

THEOREM 4.13. *Let \mathcal{R} be a TRS and $G \rightsquigarrow_{\theta}^* \top$ a basic NC-refutation. For every selection function \mathcal{S} there exists an LNC-refutation $G \Rightarrow_{\theta'}^* \square$ respecting \mathcal{S} such that $\theta' = \theta \upharpoonright \text{Var}(G)$. \square*

A related result for lazy paramodulation calculi is given by Moser [18]. He showed the completeness of his calculus \mathcal{T}_{BP} , a refined version of the calculus \mathcal{T} of Gallier and Snyder [5], by a reduction to the basic superposition calculus \mathcal{S} of [1]. Strong completeness (of \mathcal{T}_{BP}) follows because \mathcal{T}_{BP} satisfies the so-called “switching lemma”. Since from every \mathcal{T}_{BP} -refutation one easily extracts a \mathcal{T} -refutation respecting the same selection function, strong completeness of \mathcal{T} is an immediate consequence.

Basic narrowing is known to be complete (with respect to normalized solutions) for confluent and terminating TRSs (Hullot [11]), orthogonal TRSs under the additional assumption that $G\theta$ is normalizable (Middeldorp and Hamoen [16]), and confluent and right-linear TRSs ([16]). Hence for these three classes of TRSs we obtain the strong completeness (with respect to normalized solutions) of LNC as a corollary of Theorem 4.13.

The converse of Theorem 4.13 does not hold, as witnessed by the confluent TRS $\mathcal{R} = \{f(x) \rightarrow g(x, x), a \rightarrow b, g(a, b) \rightarrow c, g(b, b) \rightarrow f(a)\}$ from Middeldorp and Hamoen [16]. They show that the goal $f(a) \doteq c$ cannot be solved by basic narrowing. Straightforward calculations reveal that for any selection function \mathcal{S} there exists an LNC-refutation $f(a) \doteq c \Rightarrow^* \square$ respecting \mathcal{S} .

5. Completeness

In this section we show the completeness of LNC for confluent TRSs with respect to normalized solutions. Actually we show a stronger result: all normalized solutions are subsumed by substitutions produced by LNC-refutations that respect \mathcal{S}_{left} . Basic narrowing is of no help because of its incompleteness [16] for this general case. If we are able to define a class of NC-refutations respecting \mathcal{S}_{left} that (1) includes all NC-refutations respecting \mathcal{S}_{left} that produce normalized solutions, and (2) which is closed under the transformation steps $\phi_{[on]}$, $\phi_{[d]}$, $\phi_{[im]}$, and ϕ_{UC} , then completeness with respect to \mathcal{S}_{left} follows along the lines of the proof of Theorem 4.13. We didn't succeed in defining such a class, the main problem being the fact that an application of $\phi_{[on]}$ or $\phi_{[d]}$ to an NC-refutation that respects \mathcal{S}_{left} may result in an NC-refutation that doesn't respect \mathcal{S}_{left} . We found however a class of NC-refutations respecting \mathcal{S}_{left} that satisfies the first property and which is closed under $\phi_{[on]} \circ \phi_1$, $\phi_{[d]} \circ \phi_2$, $\phi_{[im]}$, and ϕ_{UC} . Here ϕ_1 and ϕ_2 are transformations that preprocess a given NC-refutation in such a way that a subsequent application of $\phi_{[on]}$ and $\phi_{[d]}$ results in an NC-refutation respecting \mathcal{S}_{left} . The following definition introduces our class of NC-refutations.

DEFINITION 5.1. An NC-refutation $\Pi: G \rightsquigarrow_{\theta}^* \top$ respecting \mathcal{S}_{left} is called *normal* if it satisfies the following property: if narrowing is applied to the left-hand side (right-hand side) of a descendant of an equation $s \doteq t$ in G then $\theta_2 \upharpoonright_{\text{var}(s\theta_1)}$ ($\theta_2 \upharpoonright_{\text{var}(t\theta_1)}$) is normalized. Here θ_1 and θ_2 are defined by writing Π as $G = G_1, s \doteq t, G_2 \rightsquigarrow_{\theta_1}^* \top, (s \doteq t, G_2)\theta_1 \rightsquigarrow_{\theta_2}^* \top$.

The following result states that the class of normal NC-refutations satisfies property (1) mentioned above.

LEMMA 5.2. *Every NC-refutation respecting \mathcal{S}_{left} that produces a normalized solution is normal.* \square

The converse of this lemma is not true. The next two lemmata introduce the transformations ϕ_1 and ϕ_2 . Both proofs rely on the same switching lemma.

LEMMA 5.3. *Let e be the equation $s \doteq t$. For every normal NC-refutation $\Pi: e, G \rightsquigarrow_{\theta_1}^* s' \doteq t', G\theta_1 \rightsquigarrow_{\theta_2, 1, l \mapsto r} (r \doteq t', G\theta_1)\theta_2 \rightsquigarrow_{\theta_3}^* \top$ with the property that narrowing is not applied to a descendant of e at position 1 in the subderivation that produces substitution θ_1 , there exists a normal NC-refutation $\phi_1(\Pi): e, G \rightsquigarrow_{\sigma_1}^* s'' \doteq t\sigma_1, G\sigma_1 \rightsquigarrow_{\sigma_2, 1, l \mapsto r} (r \doteq t\sigma_1, G\sigma_1)\sigma_2 \rightsquigarrow_{\sigma_3}^* \top$ with the same complexity such that $\sigma_1\sigma_2\sigma_3 = \theta_1\theta_2\theta_3$ and narrowing is neither applied at position 1 nor in the right-hand side of a descendant of e in the subderivation that produces the substitution σ_1 .* \square

LEMMA 5.4. *Let e be the equation $f(s_1, \dots, s_n) \doteq f(t_1, \dots, t_n)$. For every normal NC-refutation $\Pi: e, G \rightsquigarrow_{\theta_1}^* \text{true}, G\theta_1 \rightsquigarrow_{\theta_2}^* \top$ with the property that narrowing is never applied to a descendant of e at position 1 or 2, there exists a normal NC-refutation $\phi_2(\Pi): e, G \rightsquigarrow_{\theta_1}^* \text{true}, G\theta_1 \rightsquigarrow_{\theta_2}^* \top$ with the same complexity such that in the subderivation producing substitution θ_1 narrowing is applied to the subterms $s_1, \dots, s_n, t_1, \dots, t_n$ in the order $s_1, t_1, s_2, t_2, \dots, s_n, t_n$.* \square

The next result states that the transformation steps $\phi_{[on]} \circ \phi_1$, $\phi_{[d]} \circ \phi_2$, $\phi_{[im]}$, and ϕ_{UC} preserve normality.

LEMMA 5.5. *Let Π be a normal NC-refutation. The NC-refutations $\phi_{[on]}(\phi_1(\Pi))$, $\phi_{[d]}(\phi_2(\Pi))$, $\phi_{[im]}(\Pi)$, and $\phi_{UC}(\Pi)$ are normal whenever they are defined. \square*

EXAMPLE 5.6. Consider again the NC-refutation Π_{fail} of Example 4.10. This refutation is easily seen to be normal. An application of $\phi_{[on]}$ results in the NC-refutation $\phi_{[on]}(\Pi_{fail}): f(a) \doteq f(x), x \doteq g(a) \rightsquigarrow f(a) \doteq f(x), x \doteq g(b) \rightsquigarrow a \doteq g(b) \rightsquigarrow b \doteq g(b) \rightsquigarrow g(b) \doteq g(b) \rightsquigarrow \text{true}$, which doesn't respect S_{left} . If we first apply ϕ_1 we obtain the NC-refutation $\phi_1(\Pi_{fail}): f(a) \doteq g(a) \rightsquigarrow a \doteq g(a) \rightsquigarrow a \doteq g(b) \rightsquigarrow b \doteq g(b) \rightsquigarrow g(b) \doteq g(b) \rightsquigarrow \text{true}$. An application of $\phi_{[on]}$ to this normal NC-refutation yields $\phi_{[on]}(\phi_1(\Pi_{fail})): f(a) \doteq f(x), x \doteq g(a) \rightsquigarrow a \doteq g(a) \rightsquigarrow a \doteq g(b) \rightsquigarrow b \doteq g(b) \rightsquigarrow g(b) \doteq g(b) \rightsquigarrow \text{true}$. This NC-refutation is normal even though the produced substitution restricted to the variable in the initial goal is not normalized.

The following lemma is the counterpart of Lemma 4.12 for normal NC-refutations.

LEMMA 5.7. *For every NC-refutation $\Pi: G \rightsquigarrow_{\theta}^* \top$ with $\theta|_{\text{var}(G)}$ is normalized there exists a normal NC-refutation $\Pi_{S_{left}}: G \rightsquigarrow_{\theta}^* \top$ of the same complexity. \square*

Putting all pieces together, the following result can be proved along the lines of the proof of Theorem 4.13.

THEOREM 5.8. *Let \mathcal{R} be a TRS. For every NC-refutation $G \rightsquigarrow_{\theta}^* \top$ with the property that $\theta|_{\text{var}(G)}$ is normalized there exists an LNC-refutation $G \Rightarrow_{\theta}^* \top$ respecting S_{left} such that $\theta' = \theta|_{\text{var}(G)}$. \square*

COROLLARY 5.9. *Let \mathcal{R} be a confluent TRS. If $\mathcal{R} \vdash G\theta$ and $\theta|_{\text{var}(G)}$ is normalized then there exists an LNC-refutation $G \Rightarrow_{\theta'}^* \square$ respecting S_{left} such that $\theta' \leq \theta|_{\text{var}(G)}$. \square*

6. Eager Variable Elimination

LNC has three sources of non-determinism: the choice of the equation in the given goal, the choice of the inference rule, and the choice of the rewrite rule (in the case of $[on]$). In Section 4 we were concerned with the first kind of non-determinism. In this section we address the second kind of non-determinism. The non-deterministic application of the various inference rules to selected equations causes LNC to generate many redundant derivations. Consider for example the (orthogonal hence confluent) TRS $\{f(g(x)) \rightarrow a, b \rightarrow g(b)\}$. Figure 1 shows all LNC-refutations issued from the goal $f(b) \doteq a$ that respect the selection function S_{left} . There are infinitely many such refutations. Because the initial goal is ground, one of them suffices for completeness. At several places in the literature

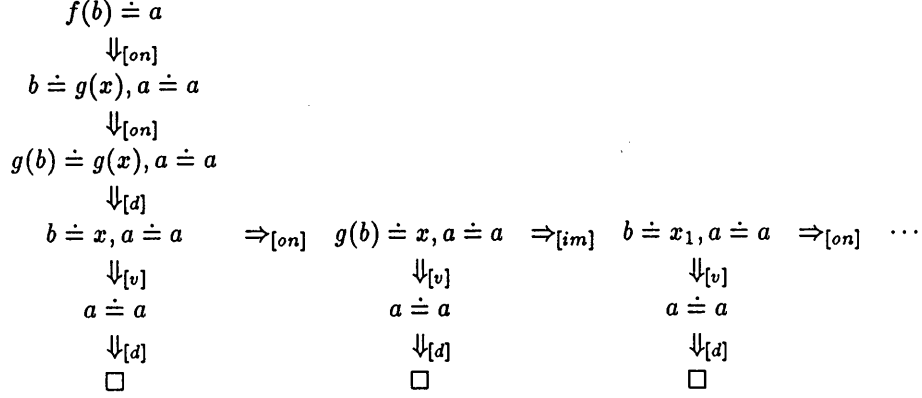


Fig. 1.

it is mentioned that this type of redundancy can be greatly reduced by applying the variable elimination rule $[v]$ prior to other applicable inference rules, although to the best of our knowledge there is no supporting proof of this so-called *eager variable elimination problem* for the general case of confluent systems.

In this section we show that a restricted version of the eager variable elimination strategy is complete with respect to \mathcal{S}_{left} for orthogonal TRSs. Before we can define our strategy, we need to extend the concept of descendant to LNC-derivations. Descendants of non-selected equations are defined as in Definition 4.1. The selected equation $f(s_1, \dots, s_n) \doteq t$ in the outermost narrowing rule $[on]$ has the body equation $r \doteq t$ as only (one-step) descendant. In the imitation rule $[im]$, all equations $s_i \theta \doteq x_i$ ($1 \leq i \leq n$) are descendants of the selected equation $f(s_1, \dots, s_n) \doteq x$. The selected equation $f(s_1, \dots, s_n) \doteq f(t_1, \dots, t_n)$ in the decomposition rule $[d]$ has all equations $s_1 \doteq t_1, \dots, s_n \doteq t_n$ as (one-step) descendants. Finally, the selected equations in $[v]$ and $[t]$ have no descendants.

DEFINITION 6.1. An equation of the form $x \doteq t$, with $x \notin \text{Var}(t)$, is called *solved*. An LNC-derivation Π is called *eager* if the variable elimination rule $[v]$ is applied to all selected solved equations that are descendants of a parameter-passing equation in Π .

Of the infinitely many LNC-refutations in Figure 1 only the leftmost one is eager since all others apply the outermost narrowing rule $[on]$ to the solved descendant $b \doteq x$ of the parameter-passing equation $b \doteq g(x)$ introduced in the first $\Rightarrow_{[on]}$ -step.

In this section we prove that eager LNC is complete with respect to \mathcal{S}_{left} for orthogonal TRSs (with respect to normalized solutions). The outline of our proof is as follows.

- (1) We define *outside-in* NC-derivations. These are the narrowing counterpart to the outside-in rewrite sequences of Huet and Lévy [10].
- (2) We show that the completeness of outside-in NC for orthogonal TRSs with respect to normalized solutions is an easy consequence of Huet and Lévy's

standardization theorem.

- (3) We show that the translation steps ϕ_1 , ϕ_2 , $\phi_{[on]}$, $\phi_{[d]}$, $\phi_{[im]}$, and ϕ_{UC} preserve the outside-in property.
- (4) We verify that the LNC-refutation obtained from an outside-in NC-refutation by means of the transformation described in the previous section is in fact eager.

Before defining outside-in NC-derivations, we introduce the concept of *NC-trace*. Let $\Pi : G \rightsquigarrow_{\theta} G'$ be an NC-step and e an equation in G different from **true**. Let e' be the (unique) descendant of e in G' . The construct $e \rightsquigarrow_{\theta} e'$ is called a one-step NC-trace. NC-traces are obtained by concatenating one-step NC-traces. An NC-trace $e_1 \rightsquigarrow_{\theta_1} \dots \rightsquigarrow_{\theta_{n-1}} e_n$ may be rendered as $e_1 \rightsquigarrow_{\theta}^* e_n$ where $\theta = \theta_1 \dots \theta_{n-1}$. For every such NC-trace π there is a corresponding rewrite sequence $\mathcal{R}(\pi) : e_1 \theta \rightarrow_{\mathcal{R}_+}^* e_n$. This rewrite sequence will be shorter than π if the latter contains one-step NC-traces of the form $e_i \rightsquigarrow_{\theta_i} e_i \theta_i$ —indicating that e_i was not selected in the underlying NC-step—which translate to identity at the rewrite level.

DEFINITION 6.2. Let \mathcal{R} be an orthogonal TRS. A rewrite sequence $e_1 \rightarrow_{p_1, l_1 \rightarrow r_1} \dots \rightarrow_{p_{n-1}, l_{n-1} \rightarrow r_{n-1}} e_n$ in \mathcal{R}_+ is called *outside-in* if the following condition is satisfied for all $1 \leq i < n - 1$: if there exists a j with $i < j < n$ such that $\epsilon < p_j < p_i$ then $p_i \setminus p_j \in \mathcal{Pos}_{\mathcal{F}}(l_j)$ for the least such j .

This definition is equivalent to the one given by Huet and Lévy in their seminal paper [10] on call-by-need computations in orthogonal TRSs. The following result is an immediate consequence of their standardization theorem (Theorem 3.19 in [10]).

THEOREM 6.3. *Let \mathcal{R} be an orthogonal TRS. For every rewrite sequence $e \rightarrow_{\mathcal{R}_+}^* \mathbf{true}$ there exists an outside-in rewrite sequence $e \rightarrow_{\mathcal{R}_+}^* \mathbf{true}$. \square*

DEFINITION 6.4. Let \mathcal{R} be an orthogonal TRS. An NC-derivation Π issued from a goal G is called *outside-in* if $\mathcal{R}(\pi)$ is outside-in for all traces π of the equations $e \in G$.

The following result is an easy consequence of Theorem 6.3 and the lifting lemma for NC.

THEOREM 6.5. *Let \mathcal{R} be an orthogonal TRS. For every NC-refutation $G \rightsquigarrow_{\theta}^* \top$ with $\theta|_{\text{Var}(G)}$ normalized there exists an outside-in NC-refutation $G \rightsquigarrow_{\theta'}^* \top$ such that $\theta' \leq \theta|_{\text{Var}(G)}$. \square*

The above theorem extends and simplifies the main result of You [22]: the completeness of *outer* narrowing for orthogonal *constructor-based* TRSs with respect to *constructor-based* solutions. One easily verifies that outer narrowing coincides with outside-in narrowing in the case of orthogonal constructor-based TRSs and that constructor-based substitutions are a special case of normalized substitutions. Hence You's completeness result (Theorem 3.13 in [22]) is a consequence of Theorem 6.5. Since You doesn't use the powerful standardization

theorem of Huet and Lévy, his completeness proof is (much) more complicated. Moreover, our result covers a larger class of TRSs.

A careful inspection of the transformation steps described in the preceding two sections reveals that ϕ_1 , ϕ_2 , $\phi_{[on]}$, $\phi_{[d]}$, $\phi_{[im]}$, and ϕ_{UC} preserve the outside-in property. Moreover, the LNC-refutation obtained from an outside-in NC-refutation can be shown to be eager, i.e., we obtain the following result.

THEOREM 6.6. *Let \mathcal{R} be an orthogonal TRS. For every outside-in NC-refutation $G \rightsquigarrow_{\theta}^* \top$ with $\theta|_{\text{Var}(G)}$ normalized there exists an eager LNC-refutation $G \Rightarrow_{\theta'}^* \top$ respecting \mathcal{S}_{left} such that $\theta' \leq \theta [Var(G)]$. \square*

The combination of the last two results yields the final result of this paper.

THEOREM 6.7. *Let \mathcal{R} be an orthogonal TRS. If $\mathcal{R} \vdash G\theta$ and $\theta|_{\text{Var}(G)}$ is normalized then there exists an eager LNC-refutation $G \Rightarrow_{\theta'}^* \top$ respecting \mathcal{S}_{left} such that $\theta' \leq \theta [Var(G)]$. \square*

7. Suggestions for Further Research

This paper leaves many questions unanswered. In the near future we would like to address the following two problems.

In Section 4 we have shown the strong completeness of LNC in the case of orthogonal TRSs, using the transformation steps $\phi_{[on]}$, $\phi_{[d]}$, $\phi_{[im]}$, and ϕ_{UC} . In Section 6 we showed the completeness of eager LNC with respect to \mathcal{S}_{left} for orthogonal TRSs, using the transformation steps $\phi_{[on]} \circ \phi_1$, $\phi_{[d]} \circ \phi_2$, $\phi_{[im]}$, and ϕ_{UC} . A natural question is whether these two results can be combined, i.e., is eager LNC strongly complete for orthogonal TRSs. Recall the NC-refutation Π_{fail} in Example 4.10. It is easy to see that Π_{fail} is outside-in. Nevertheless, $\phi_{[on]}$, together with $\phi_{[d]}$, results in failure. Hence the use of ϕ_1 and ϕ_2 is essential for the completeness of eager LNC. This suggests that it is not obvious whether or not eager LNC is strongly complete for orthogonal TRSs.

The orthogonality assumption in our proof of the completeness of eager LNC is essential since we make use of Huet and Lévy's standardization theorem. We didn't succeed in finding a non-orthogonal TRS for which eager LNC is not complete. Hence it is an open problem whether our restricted variable elimination strategy is complete for arbitrary confluent TRSs with respect to normalized solutions. A more general question is of course whether the variable elimination rule can always be eagerly applied, i.e., is the restriction to solved descendants of parameter-passing equations essential? In a recent paper Socher-Ambrosius [21] reports that the eager variable elimination problem has a positive solution in case of lazy paramodulation for arbitrary equational theories. It remains to be seen whether his techniques can be lifted to the present setting.

Acknowledgements. We thank an anonymous referee for drawing our attention to the work of Moser.

References

1. L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder, *Basic Paramodulation and Superposition*, Proc. 11th CADE, LNCS 607, pp. 462–476, 1992.
2. N. Dershowitz and J.-P. Jouannaud, *Rewrite Systems*, in: Handbook of Theoretical Computer Science, Vol. B, ed. J. van Leeuwen), North-Holland, pp. 243–320, 1990.
3. N. Dershowitz and Z. Manna, *Proving Termination with Multiset Orderings*, Communications of the ACM 22(8), pp. 465–476, 1979.
4. M. Fay, *First-Order Unification in Equational Theories*, Proc. 4th CADE, Austin, pp. 161–167, 1979.
5. J. Gallier and W. Snyder, *Complete Sets of Transformations for General E-Unification*, TCS 67, pp. 203–260, 1989.
6. M. Hanus, *Efficient Implementation of Narrowing and Rewriting*, Proc. PDK-91, LNAI 567, pp. 344–365, 1991.
7. M. Hanus, *The Integration of Functions into Logic Programming: From Theory to Practice*, JLP 19 & 20, pp. 583–628, 1994.
8. S. Hölldobler, *A Unification Algorithm for Confluent Theories*, Proc. 14th ICALP, LNCS 267, pp. 31–41, 1987.
9. S. Hölldobler, *Foundations of Equational Logic Programming*, LNAI 353, 1989.
10. G. Huet and J.-J. Lévy, *Computations in Orthogonal Rewriting Systems, I and II*, in: Computational Logic, Essays in Honor of Alan Robinson (eds. J.-L. Lassez and G. Plotkin), The MIT Press, pp. 396–443, 1991.
11. J.-M. Hullot, *Canonical Forms and Unification*, Proc. 5th CADE, LNCS 87, pp. 318–334, 1980.
12. J.W. Klop, *Term Rewriting Systems*, in: Handbook of Logic in Computer Science, Vol. II (eds. S. Abramsky, D. Gabbay, and T. Maibaum), Oxford University Press, pp. 1–116, 1992.
13. A. Martelli, C. Moiso, and G.F. Rossi, *Lazy Unification Algorithms for Canonical Rewrite Systems*, in: Resolution of Equations in Algebraic Structures, Vol. II (eds. H. Ait-Kaci and M. Nivat), Academic Press, pp. 245–274, 1989.
14. A. Martelli and U. Montanari, *An Efficient Unification Algorithm*, ACM TOPLAS 4(2), pp. 258–282, 1982.
15. A. Martelli, G.F. Rossi, and C. Moiso, *An Algorithm for Unification in Equational Theories*, Proc. 1986 Symposium on Logic Programming, pp. 180–186, 1986.
16. A. Middeldorp and E. Hamoen, *Completeness Results for Basic Narrowing*, AECC 5, pp. 213–253, 1994.
17. A. Middeldorp, S. Okui, and T. Ida, *Lazy Narrowing: Strong Completeness and Eager Variable Elimination* (in preparation).
18. M. Moser, *Improving Transformation Systems for General E-Unification*, Proc. 5th RTA, LNCS 690, pp. 92–105, 1993.
19. J.R. Slagle, *Automatic Theorem Proving in Theories with Simplifiers, Commutativity and Associativity*, Journal of the ACM 21, pp. 622–642, 1974.
20. W. Snyder, *A Proof Theory for General Unification*, Birkhäuser, 1991.
21. R. Socher-Ambrosius, *A Refined Version of General E-Unification*, Proc. 12th CADE, LNAI 814, pp. 665–677, 1994.
22. Y.H. You, *Enumerating Outer Narrowing Derivations for Constructor Based Term Rewriting Systems*, JSC 7, pp. 319–343, 1989.