Complete Selection Functions for a Lazy Conditional Narrowing Calculus

Aart Middeldorp

Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305-8573, Japan
ami@is.tsukuba.ac.jp

Taro Suzuki

Department of Computer Software
University of Aizu, Aizu-Wakamatsu 965-8580, Japan
taro@u-aizu.ac.jp

Mohamed Hamada

Department of Computer Software
University of Aizu, Aizu-Wakamatsu 965-8580, Japan
hamada@u-aizu.ac.jp

Abstract

In this paper we extend the lazy narrowing calculus LNC of Middeldorp, Okui, and Ida [26] to conditional rewrite systems. The resulting lazy conditional narrowing calculus LCNC is highly non-deterministic. We investigate for which classes of conditional rewrite systems the completeness of LCNC is ensured. In order to improve the efficiency of the calculus, we pay special attention to the removal of non-determinism due to the selection of equations in goals by fixing a selection strategy.

1 Introduction

Narrowing (Fay [7], Hullot [19]) was originally invented as a general method for solving unification problems in equational theories that are presented by confluent term rewriting systems (TRSs for short). More recently, narrowing was proposed as the computational mechanism of several functional-logic programming languages (Hanus [16]) and several new completeness results concerning the completeness of various narrowing strategies and calculi have been obtained in the past few years. Here completeness means that for every solution to a given goal a solution that is at least as general is computed by the narrowing strategy. Since narrowing is a complicated operation, numerous calculi consisting of a small number of more elementary inference rules that simulate narrowing have been proposed (e.g. [8, 18, 23, 30, 17, 26, 10, 29, 22, 11]).

Completeness issues for the lazy narrowing calculus LNC—which is based on the calculus TRANS of Hölldobler [18]—have been extensively studied in [26] and [25]. In [26] Middeldorp *et al.* prove that LNC is *strongly* complete whenever *basic* narrowing (Hullot [19]) is complete. Strong completeness means that the choice of the equation in goals can be made don't care non-deterministic, resulting in a huge reduction of the search space as well as easing implementations. For the completeness of basic narrowing several sufficient conditions are known, including termination. It is also shown in [26] that LNC is complete for arbitrary confluent TRSs and normalized solutions with respect to the selection function $\mathcal{S}_{\text{left}}$ that selects the leftmost equation in every goal. (For this general class of TRSs, LNC is not strongly complete [26, Counterexample 10].) Based on the latter result Middeldorp and Okui [25] present restrictions on the participating TRSs and solutions which guarantee that all non-determinism due to the choice of inference rules of LNC is removed. The resulting deterministic calculus LNC_d satisfies the optimality property that different derivations compute incomparable solutions for a class of TRSs that properly includes the class of TRSs for which a similar result was obtained by Antoy *et al.* in the setting of needed narrowing [1].

In this paper we extend LNC to deal with conditional TRSs (CTRSs for short). We present three main completeness results:

- LCNC with $\mathcal{S}_{\text{left}}$ is complete with respect to normalizable solutions for the class of confluent but not necessarily terminating conditional rewrite systems without so-called extra variables in the conditional parts of the rewrite rules.

- LCNC is strongly complete whenever basic conditional narrowing is complete. The latter is known for decreasing and confluent CTRSs without extra variables in the rewrite rules (Middeldorp and Hamoen [24]), for level-complete CTRSs with extra variables in the conditions only (Giovannetti and Moiso [9], Middeldorp and Hamoen [24]), and for terminating and shallow-confluent normal CTRSs with extra variables (Werner [33]).

- LCNC is complete for the class of terminating and level-confluent conditional rewrite systems without any restrictions on the distribution of variables in the rewrite rules. Unlike the previous two results, the proof of this last result does not provide any complete selection strategy. As a matter of fact, the selection strategy used in the proof is not effective in that it refers to the rewrite sequence that shows that the solution that we want to approximate with LCNC is actually a solution. It is an open question whether this result can be strengthened to completeness with respect to a fixed selection function or even to strong completeness.

The first two results generalize two of the three main results of [26] to the conditional case. The third result has no counterpart in the unconditional case. We stress that without a complete selection function, in implementations we need to backtrack over the choice of equations in goals in order to guarantee that all solutions are enumerated. This complicates implementations and, worse, leads to a dramatic increase in the search space, even more so since in conditional narrowing (whether presented as a single inference rule or in the form of a calculus like LCNC) the conditions of the applied rewrite rule are added to the current goal after every narrowing step.

The remainder of the paper is organized as follows. In the next section we recall some definitions pertaining to conditional rewriting and we present the calculus LCNC. Sections 3, 4, and 5 are devoted to the proofs of our three completeness results. We make some concluding remarks and list several open problems in Section 6. The Appendix contains the proofs of two technical lemmata in Section 4.

The results presented in this paper previously appeared in [14, 15, 12, 31].

2 Preliminaries

We assume familiarity with the basics of (conditional) term rewriting and narrowing. Surveys can be found in [2, 4, 21, 24]. We just recall some basic definitions in order to fix our notation and terminology.

A conditional term rewriting system (CTRS) over a signature $\mathcal{F}$ is a set $\mathcal{R}$ of (conditional) rewrite rules of the form $l \to r \Leftarrow c$ where the conditional part c is a (possibly empty) sequence $s_1 \approx t_1, \ldots, s_n \approx t_n$ of equations. All terms $l, r, s_1, \ldots, s_n, t_1, \ldots, t_n$ must belong to $\mathcal{T}(\mathcal{F}, \mathcal{V})$ and we require that l is not a variable. Here $\mathcal{V}$ denotes a countably infinite set of variables. Following [24], CTRSs are classified according to the distribution of variables in rewrite rules. A 1-CTRS contains no extra variables (i.e., $\mathcal{V}\mathrm{ar}(r, c) \subseteq \mathcal{V}\mathrm{ar}(l)$ for all rewrite rules $l \to r \Leftarrow c$), a 2-CTRS may contain extra variables in the conditions only ($\mathcal{V}\mathrm{ar}(r) \subseteq \mathcal{V}\mathrm{ar}(l)$ for all rewrite rules $l \to r \Leftarrow c$), and a 3-CTRS may also have extra variables in the right-hand sides provided these occur in the corresponding conditions ($\mathcal{V}\mathrm{ar}(r) \subseteq \mathcal{V}\mathrm{ar}(l, c)$ for all rewrite rules $l \to r \Leftarrow c$). Extra variables enable a more natural style of writing specifications of programs. For instance, using extra variables we can easily write the following specification of the efficient computation of Fibonacci numbers:

$$\begin{array}{rcl} \mathsf{0} + y & \to & y \\ \mathsf{s}(x) + y & \to & \mathsf{s}(x + y) \\ \mathsf{fib}(\mathsf{0}) & \to & \langle \mathsf{0}, \mathsf{s}(\mathsf{0}) \rangle \\ \mathsf{fib}(\mathsf{s}(x)) & \to & \langle z, y + z \rangle \;\Leftarrow\; \mathsf{fib}(x) \approx \langle y, z \rangle \end{array}$$

However, the presence of extra variables comes with a price. For instance, completeness results for narrowing that hold for arbitrary confluent TRSs and 1-CTRSs typically do not carry over to 2-CTRSs and 3-CTRSs without requiring some kind of termination assumption.

We assume that every CTRS contains the rewrite rule $x \approx x \to \mathtt{true}$. Here $\approx$ and $\mathtt{true}$ are function symbols that do not occur in the other rewrite rules. These symbols may only occur at the root position of terms. Let $\mathcal{R}$ be a CTRS. We inductively define unconditional TRSs $\mathcal{R}_n$ for $n \geqslant 0$ as follows:

$$\begin{array}{lcl} \mathcal{R}_0 & = & \{\, x \approx x \to \mathtt{true} \,\}, \\ \mathcal{R}_{n+1} & = & \{\, l\theta \to r\theta \mid l \to r \Leftarrow c \in \mathcal{R} \text{ and } c\theta \to^*_{\mathcal{R}_n} \top \,\}. \end{array}$$

Here $\top$ stands for any sequence of $\mathtt{true}$s. We define $s \to_{\mathcal{R}} t$ if and only if there exists an $n \geqslant 0$ such that $s \to_{\mathcal{R}_n} t$. We abbreviate $\to_{\mathcal{R}_n}$ to $\to_n$ and

$\rightarrow_{\mathcal{R}}$ to $\rightarrow$ if the CTRS $\mathcal{R}$ can be inferred from the context. Our CTRS are known as *join* CTRSs in the term rewriting literature.

A CTRS $\mathcal{R}$ is level-confluent (Giovannetti and Moiso [9]) if every $\mathcal{R}_n$ is confluent, i.e., ${}_n^*\!\!\leftarrow \cdot \rightarrow_n^* \subseteq \rightarrow_n^* \cdot {}_n^*\!\!\leftarrow$ for all $n \geqslant 0$, and shallow-confluent if ${}_m^*\!\!\leftarrow \cdot \rightarrow_n^* \subseteq \rightarrow_n^* \cdot {}_m^*\!\!\leftarrow$ for all $m, n \geqslant 0$. Shallow-confluent CTRSs are level-confluent but the reverse is not true. A CTRS $\mathcal{R}$ is level-terminating if every $\mathcal{R}_n$ is terminating. Level-termination is a weaker ([24]) property than termination. A level-complete CTRS $\mathcal{R}$ is both level-confluent and level-terminating. A CTRS $\mathcal{R}$ over a signature $\mathcal{F}$ is decreasing (Dershowitz *et al.* [6]) if there exists a well-founded order $\succ$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ with the following three properties: $\succ$ contains $\rightarrow_{\mathcal{R}}$, $\succ$ has the subterm property (i.e., $\rhd \subseteq \succ$ where $s \rhd t$ if and only if t is proper subterm of s), and $l\theta \succ s\theta, t\theta$ for every rewrite rule $l \rightarrow r \Leftarrow c$ of $\mathcal{R}$, every equation $s \approx t$ in c, and every substitution θ. Note that according to this definition 2-CTRSs and 3-CTRSs are never decreasing. Decreasing CTRSs are terminating and, when there are finitely many rewrite rules, have a decidable rewrite relation. Sufficient syntactic conditions for level-confluence of 3-CTRSs are presented in Suzuki *et al.* [32].

An equation is a term of the form $s \approx t$. The constant `true` is also viewed as an equation. A goal is a sequence of equations. A substitution θ is a ($\mathcal{R}$-)solution of a goal G if $s\theta \leftrightarrow_{\mathcal{R}}^* t\theta$ for every equation $s \approx t$ in G. This is equivalent to validity of the equations in $G\theta$ in all models of the underlying conditional equational system of $\mathcal{R}$ (Kaplan [20]) and for confluent $\mathcal{R}$ to $G\theta \rightarrow_{\mathcal{R}}^* \top$. We abbreviate the latter to $\mathcal{R} \vdash G\theta$. A *normalized* solution satisfies the additional property that variables are mapped to normal forms with respect to $\mathcal{R}$.

For a substitution θ and a set of variables W, we denote $(W \setminus \mathcal{D}(\theta)) \cup \mathcal{I}(\theta\restriction_W)$ by $\mathcal{V}\mathrm{ar}_W(\theta)$. Here $\mathcal{D}(\theta) = \{x \in \mathcal{V} \mid \theta(x) \neq x\}$ denotes the domain of θ, which is always assumed to be finite, and $\mathcal{I}(\theta\restriction_W) = \bigcup_{x \in \mathcal{D}(\theta) \cap W} \mathcal{V}\mathrm{ar}(x\theta)$ the set of variables introduced by the restriction of θ to W.

The *lazy conditional narrowing calculus* LCNC consists of the following five inference rules:

[o] *outermost narrowing*

$$\frac{G', f(s_1, \ldots, s_n) \simeq t, G''}{G', s_1 \approx l_1, \ldots, s_n \approx l_n, r \approx t, c, G''}$$

if there exists a fresh variant $f(l_1, \ldots, l_n) \rightarrow r \Leftarrow c$ of a rewrite rule in

$\mathcal{R}$,

[i] *imitation*

$$\frac{G', f(s_1, \ldots, s_n) \simeq x, G''}{(G', s_1 \approx x_1, \ldots, s_n \approx x_n, G'')\theta}$$

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

[d] *decomposition*

$$\frac{G', f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n), G''}{G', s_1 \approx t_1, \ldots, s_n \approx t_n, G''},$$

[v] *variable elimination*

$$\frac{G', s \simeq x, G''}{(G', G'')\theta}$$

if $x \notin \mathcal{V}\mathrm{ar}(s)$ and $\theta = \{x \mapsto s\}$,

[t] *removal of trivial equations*

$$\frac{G', x \approx x, G''}{G', G''}.$$

In the rules [o], [i], and [v], $s \simeq t$ stands for $s \approx t$ or $t \approx s$. (Since the inference rules never produce the equation `true`, we assume that LCNC deals only with goals that do not contain `true`.) Note that the outermost narrowing rule is applicable as soon as the root symbol of one side s of an equation equals the root symbol of the left-hand side l of a rewrite rule. The *parameter-passing* equations $s_1 \approx l_1, \ldots, s_n \approx l_n$ code the problem of unifying s and l. Further note that unlike higher-order narrowing calculi (e.g. [29, 22]) we do not permit outermost narrowing at variable positions. This makes the task of proving completeness (much) more challenging but results in a much smaller search space.

The only difference between LCNC and the calculus LNC of [26] is in the outermost narrowing rule: In LCNC we add the conditional part of the applied rewrite rule to the new goal.

If G and G' are the upper and lower goal in the inference rule $[\alpha]$ ($\alpha \in \{\mathrm{o}, \mathrm{i}, \mathrm{d}, \mathrm{v}, \mathrm{t}\}$), we write $G \Rightarrow_{[\alpha]} G'$. This is called an LCNC-step. The applied rewrite rule or substitution may be supplied as subscript, that is, we write things like $G \Rightarrow_{[\mathrm{o}],\, l \to r \Leftarrow c} G'$ and $G \Rightarrow_{[\mathrm{i}],\, \theta} G'$. A finite LCNC-derivation $G_1 \Rightarrow_{\theta_1} \cdots \Rightarrow_{\theta_{n-1}} G_n$ may be abbreviated to $G_1 \Rightarrow^*_{\theta} G_n$ where θ is the composition $\theta_1 \cdots \theta_{n-1}$ of the substitutions $\theta_1, \ldots, \theta_{n-1}$ computed along its steps. An LCNC-refutation is an LCNC-derivation ending in the empty goal $\square$.

Example 2.1 *Consider the CTRS $\mathcal{R}$ for computing Fibonacci numbers from the introduction. The goal* $\mathsf{fib}(x) \approx \langle x, x\rangle$ *admits the solution* $\{x \mapsto \mathsf{s}(\mathsf{0})\}$ *because of the following rewrite sequence:*

$$\mathsf{fib}(\mathsf{s}(\mathsf{0})) \;\to\; \langle \mathsf{s}(\mathsf{0}), \mathsf{0} + \mathsf{s}(\mathsf{0})\rangle \;\to\; \langle \mathsf{s}(\mathsf{0}), \mathsf{s}(\mathsf{0})\rangle$$

In the first step the rewrite rule $\mathsf{fib}(\mathsf{s}(x)) \to \langle z, y + z\rangle \Leftarrow \mathsf{fib}(x) \approx \langle y, z\rangle$ *is applied with substitution* $\{x \mapsto \mathsf{0}, y \mapsto \mathsf{0}, z \mapsto \mathsf{s}(\mathsf{0})\}$*; the instantiated condition is satisfied because of the rewrite rule* $\mathsf{fib}(\mathsf{0}) \to \langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle$*. The following* LCNC*-derivation ends in the unsolvable goal* $\mathsf{s}(\mathsf{0}) \approx \mathsf{0}$*:*

$$\begin{array}{l}
\underline{\mathsf{fib}(x) \approx \langle x, x\rangle} \\
\quad \Downarrow_{[\mathrm{o}],\ \mathsf{fib}(\mathsf{0}) \to \langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle} \\
\underline{x \approx \mathsf{0}},\ \langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle \approx \langle x, x\rangle \\
\quad \Downarrow_{[\mathrm{v}],\ \{x \mapsto \mathsf{0}\}} \\
\underline{\langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle \approx \langle \mathsf{0}, \mathsf{0}\rangle} \\
\quad \Downarrow_{[\mathrm{d}]} \\
\underline{\mathsf{0} \approx \mathsf{0}},\ \mathsf{s}(\mathsf{0}) \approx \mathsf{0} \\
\quad \Downarrow_{[\mathrm{d}]} \\
\mathsf{s}(\mathsf{0}) \approx \mathsf{0}
\end{array}$$

The underlined equations are selected in each step. Note that none of the inference rules of LCNC *are applicable to* $\mathsf{s}(\mathsf{0}) \approx \mathsf{0}$*. In the first step of the above derivation the rewrite rule* $\mathsf{fib}(\mathsf{0}) \to \langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle$ *is chosen. If we choose the rule* $\mathsf{fib}(\mathsf{s}(x)) \to \langle z, y + z\rangle \Leftarrow \mathsf{fib}(x) \approx \langle y, z\rangle$ *instead,* LCNC *is able to solve the goal* $\mathsf{fib}(x) \approx \langle x, x\rangle$*:*

$$\begin{array}{l}
\underline{\mathsf{fib}(x) \approx \langle x, x\rangle} \\
\quad \Downarrow_{[\mathrm{o}],\ \mathsf{fib}(\mathsf{s}(x_1)) \to \langle z_1, y_1 + z_1\rangle \Leftarrow \mathsf{fib}(x_1) \approx \langle y_1, z_1\rangle}
\end{array}$$

$$x \approx \mathsf{s}(x_1),\ \underline{\langle z_1, y_1 + z_1\rangle \approx \langle x, x\rangle},\ \mathsf{fib}(x_1) \approx \langle y_1, z_1\rangle$$

$$\Downarrow_{[\mathrm{d}]}$$

$$x \approx \mathsf{s}(x_1),\ \underline{z_1 \approx x},\ y_1 + z_1 \approx x,\ \mathsf{fib}(x_1) \approx \langle y_1, z_1\rangle$$

$$\Downarrow_{[\mathrm{v}]},\ \{z_1 \mapsto x\}$$

$$x \approx \mathsf{s}(x_1),\ y_1 + x \approx x,\ \underline{\mathsf{fib}(x_1) \approx \langle y_1, x\rangle}$$

$$\Downarrow_{[\mathrm{o}]},\ \mathsf{fib}(\mathsf{0}) \to \langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle$$

$$x \approx \mathsf{s}(x_1),\ y_1 + x \approx x,\ \underline{x_1 \approx \mathsf{0}},\ \langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle \approx \langle y_1, x\rangle$$

$$\Downarrow_{[\mathrm{v}]},\ \{x_1 \mapsto \mathsf{0}\}$$

$$x \approx \mathsf{s}(\mathsf{0}),\ y_1 + x \approx x,\ \underline{\langle \mathsf{0}, \mathsf{s}(\mathsf{0})\rangle \approx \langle y_1, x\rangle}$$

$$\Downarrow_{[\mathrm{d}]}$$

$$x \approx \mathsf{s}(\mathsf{0}),\ y_1 + x \approx x,\ \underline{\mathsf{0} \approx y_1},\ \mathsf{s}(\mathsf{0}) \approx x$$

$$\Downarrow_{[\mathrm{v}]},\ \{y_1 \mapsto \mathsf{0}\}$$

$$\underline{x \approx \mathsf{s}(\mathsf{0})},\ \mathsf{0} + x \approx x,\ \mathsf{s}(\mathsf{0}) \approx x$$

$$\Downarrow_{[\mathrm{v}]},\ \{x \mapsto \mathsf{s}(\mathsf{0})\}$$

$$\mathsf{0} + \mathsf{s}(\mathsf{0}) \approx \mathsf{s}(\mathsf{0}),\ \underline{\mathsf{s}(\mathsf{0}) \approx \mathsf{s}(\mathsf{0})}$$

$$\Downarrow_{[\mathrm{d}]}$$

$$\mathsf{0} + \mathsf{s}(\mathsf{0}) \approx \mathsf{s}(\mathsf{0}),\ \underline{\mathsf{0} \approx \mathsf{0}}$$

$$\Downarrow_{[\mathrm{d}]}$$

$$\underline{\mathsf{0} + \mathsf{s}(\mathsf{0}) \approx \mathsf{s}(\mathsf{0})}$$

$$\Downarrow_{[\mathrm{o}]},\ \mathsf{0} + y_2 \to y_2$$

$$\mathsf{0} \approx \mathsf{0},\ \mathsf{s}(\mathsf{0}) \approx y_2,\ \underline{y_2 \approx \mathsf{s}(\mathsf{0})}$$

$$\Downarrow_{[\mathrm{v}]},\ \{y_2 \mapsto \mathsf{s}(\mathsf{0})\}$$

$$\underline{\mathsf{0} \approx \mathsf{0}},\ \mathsf{s}(\mathsf{0}) \approx \mathsf{s}(\mathsf{0})$$

$$\Downarrow_{[\mathrm{d}]}$$

$$\underline{\mathsf{s}(\mathsf{0}) \approx \mathsf{s}(\mathsf{0})}$$

$$\Downarrow_{[\mathrm{d}]}$$

$$\underline{\mathsf{0} \approx \mathsf{0}}$$

$$\Downarrow_{[\mathrm{d}]}$$

$$\Box$$

The solution computed by this refutation is obtained by composing the substitutions $\{z_1 \mapsto x\}$, $\{x_1 \mapsto \mathsf{0}\}$, $\{y_1 \mapsto \mathsf{0}\}$, $\{x \mapsto \mathsf{s}(\mathsf{0})\}$, $\{y_2 \mapsto \mathsf{s}(\mathsf{0})\}$ *employed in the* $\Rightarrow_{[\mathrm{v}]}$*-steps, and restricting the resulting substitution to the variable* x *in the initial goal, which yields* $\{x \mapsto \mathsf{s}(\mathsf{0})\}$.

The following lemma states the soundness of LCNC. The routine induction proof is omitted.

Lemma 2.2 *Let* $\mathcal{R}$ *be a CTRS and* G *a goal. If* $G \Rightarrow^*_\theta \square$ *then* $\theta\restriction_{\mathcal{V}\mathrm{ar}(G)}$ *is an* $\mathcal{R}$*-solution of* G. $\square$

3 Leftmost Selection

This section contains our first main result, the completeness of LCNC for arbitrary confluent 1-CTRSs with respect to normalized solutions and the leftmost selection function $\mathcal{S}_{\mathrm{left}}$. So we assume throughout this section that the sequence G' of equations to the left of the selected equation in the inference rules of LCNC is empty.

In Middeldorp *et al.* [26] the same result is proved for *unconditional* TRSs by means of a complicated inductive transformation process that operates on narrowing sequences. In the proof presented in this section we use conditional rewrite sequences instead. The advantage of rewriting is that rewrite steps applied to different parts of a goal or equation can be swapped at will, which greatly facilitates a proof of completeness with respect to a particular selection strategy. In the proof below we use the variant of conditional rewriting in which the list of instantiated conditions of the applied rewrite rule is explicitly added to the goal after every rewrite step. Formally, we use the relation $\rightarrowtail$ defined as follows: $G \rightarrowtail G'$ if $G = G_1, e, G_2$, $e \rightarrow e'$ by applying the conditional rewrite rule $l \rightarrow r \Leftarrow c$ with substitution θ (so $e' = e[r\sigma]_p$ for some position p in e and $\mathcal{R} \vdash c\sigma$), and $G' = G_1, e', c\sigma, G_2$. It is well-known ([3, 24]) that $\mathcal{R} \vdash G$ if and only if $G \rightarrowtail^* \top$. We assume without loss of generality that in a rewrite proof $G \rightarrowtail^* \top$ always the leftmost equation different from `true` is selected.

Below we define a couple of basic transformations on rewrite proofs $\Pi\colon G\theta \rightarrowtail^* \top$. In order to make the completeness proof work, we need to keep track of a number of variables along the transformation process. Since these variables cannot be inferred from the current rewrite proof Π, together with G and θ, we need to enrich rewrite proofs. This is the reason why we consider quadruples, called *states*, of the form $\langle G, \theta, \Pi, X\rangle$ where G is a goal, θ a solution of G, $\Pi\colon G\theta \rightarrowtail^* \top$ a rewrite proof of $G\theta$, and X a finite set of variables associated to Π. Variables in X are said to be *of interest* and variables in G but not in X are called *intermediate*. In order to avoid confusion, we occasionally write X-intermediate or even $\underline{\Pi}$-intermediate (when comparing different states with the same X).

We require that the properties defined below are satisfied.

Definition 3.1 *A state* $\underline{\Pi} = \langle G, \theta, \Pi, X\rangle$ *is called* normal *if* $\theta\lceil_X$ *is normalized. We say that* $\underline{\Pi}$ *satisfies the* variable condition *if for every equation* $s \approx t$ *in* $G = G_1, s \approx t, G_2$ *the following three conditions hold:*

VC1 *all intermediate variables in* s *occur in* G_1 *or all intermediate variables in* t *occur in* G_1,

VC2 *if* $s\theta$ *is rewritten in* Π *then all intermediate variables in* s *occur in* G_1,

VC3 *if* $t\theta$ *is rewritten in* Π *then all intermediate variables in* t *occur in* G_1.

A normal state with the variable condition is called admissible.

Example 3.2 *Consider the confluent 1-CTRS consisting of the rewrite rules*

$$\begin{array}{rcl} \mathsf{f}(x,y) & \rightarrow & \mathsf{g}(y) \quad \Leftarrow \quad x \approx \mathsf{a} \\ \mathsf{a} & \rightarrow & \mathsf{b} \\ \mathsf{g}(\mathsf{b}) & \rightarrow & \mathsf{b} \end{array}$$

and the goal $G = \mathsf{f}(x,y) \approx \mathsf{g}(y), \mathsf{g}(\mathsf{g}(y)) \approx \mathsf{a}$. *We determine* θ, Π, *and* X *such that* $\underline{\Pi} = \langle G, \theta, \Pi, X\rangle$ *is admissible. First of all, since the variable* y *occurs in both sides of the leftmost equation* $\mathsf{f}(x,y) \approx \mathsf{g}(y)$, *it must be a variable of interest for otherwise* **VC1** *is violated. So* $y \in X$ *and hence, to satisfy normality,* $\theta(y)$ *should be a normal form. The only normal form that satisfies the second equation of* G *is* b, *with associated rewrite proof*

$$\mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{a} \;\rightarrowtail\; \mathsf{g}(\mathsf{b}) \approx \mathsf{a} \;\rightarrowtail\; \mathsf{b} \approx \mathsf{a} \;\rightarrowtail\; \mathsf{b} \approx \mathsf{b} \;\rightarrowtail\; \mathtt{true}.$$

Since $\mathsf{g}(\mathsf{g}(y)) \approx \mathsf{a}$ *does not contain intermediate variables, it satisfies* **VC1**, **VC2**, *and* **VC3**. *Likewise, since* $\mathsf{g}(y)$ *lacks intermediate variables,* $\mathsf{f}(x,y) \approx \mathsf{g}(y)$ *satisfies* **VC3**. *The only way to solve this equation is by applying the first rewrite rule to its (instantiated) left-hand side. Hence, to satisfy* **VC2**,

x cannot be an intermediate variable. Consequently, $x \in X$ and thus to conclude that $\underline{\Pi}$ is admissible it only remains to show that we can substitute a normal form for x such that the equation $\mathsf{f}(x, \mathsf{b}) \approx \mathsf{g}(\mathsf{b})$ is solvable. It is easy to see that we should again take the normal form b, with associated rewrite proof

$$\begin{aligned} \mathsf{f}(\mathsf{b},\mathsf{b}) \approx \mathsf{g}(\mathsf{b}) \;&\rightarrowtail\; \mathsf{g}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b}), \mathsf{b} \approx \mathsf{a} \;\rightarrowtail\; \mathtt{true}, \mathsf{b} \approx \mathsf{a} \\ &\rightarrowtail\; \mathtt{true}, \mathsf{b} \approx \mathsf{b} \;\rightarrowtail\; \top. \end{aligned}$$

An example of a goal without admissible states is $\mathsf{f}(x) \approx x$ with respect to the TRS consisting of the rule $\mathsf{a} \to \mathsf{f}(\mathsf{a})$. Note that $\mathsf{f}(x) \approx x$ has no normalized solutions.

Lemma 3.3 *Let $\underline{\Pi} = \langle G, \theta, \Pi, X \rangle$ be an admissible state with $G = s \approx t, H$.*

1. *If $s\theta$ is rewritten in Π then s is not a variable and does not include intermediate variables.*

2. *If $t\theta$ is rewritten in Π then t is not a variable and does not include intermediate variables.*

Proof We prove the first statement. Suppose the left-hand side of $s\theta \approx t\theta$ is rewritten in Π. By **VC2** all intermediate variables in s should occur in the equations to the left of $s \approx t$ in G. Since $s \approx t$ is the leftmost equation, there cannot be intermediate variables in s. Hence, if s is a variable then it must be a variable of interest and thus $s\theta$ is a normal form because $\underline{\Pi}$ is normal. This however contradicts the assumption that $s\theta$ is rewritten in Π. The second statement is proved in exactly the same way (with **VC3** instead of **VC2**). $\square$

In the following transformation lemmata $\underline{\Pi}$ denotes an admissible state $\langle G, \theta, \Pi, X \rangle$ such that $G = s \approx t, H$ and, in Lemmata 3.4, 3.5, and 3.7, W denotes a finite set of variables such that $\mathcal{V}\mathrm{ar}(G) \subseteq W$. Recall our earlier assumption that Π respects $\mathcal{S}_{\mathrm{left}}$. In particular, in the first step of Π the equation $s \approx t$ is selected.

Lemma 3.4 *Let $s = f(s_1, \ldots, s_n)$ and suppose that a reduct of $s\theta \approx t\theta$ in Π is rewritten at position 1. If $l \to r \Leftarrow c$ is the employed rewrite rule in the first such step then there exists an admissible state $\phi_{[\mathrm{o}]}(\underline{\Pi}) = \langle G', \theta', \Pi', X \rangle$ with $G' = s \approx l, r \approx t, c, H$ such that $\theta' = \theta \; [W]$.*

Proof The given rewrite proof Π is of the form

$$G\theta \rightarrowtail^* l\tau \approx t', C, H\theta \rightarrowtail r\tau \approx t', c\tau, C, H\theta \rightarrowtail^* \top.$$

Here C are the instantiated conditions of the rewrite rules applied in the rewrite sequence from $s\theta \approx t\theta$ to $l\tau \approx t'$. Without loss of generality we assume that $\mathcal{V}\text{ar}(l \to r \Leftarrow c) \cap (X \cup W) = \varnothing$ and $\mathcal{D}(\tau) = \mathcal{V}\text{ar}(l \to r \Leftarrow c)$. Hence the substitution $\theta' = \theta \cup \tau$ is well-defined. Since $\mathcal{D}(\tau) \cap W = \varnothing$, $\theta' = \theta\ [W]$. We have $G'\theta' = s\theta \approx l\tau, r\tau \approx t\theta, c\tau, H\theta$. The first part of Π can be transformed into

$$\begin{aligned} G'\theta' &\rightarrowtail^* l\tau \approx l\tau, C_1, r\tau \approx t\theta, c\tau, H\theta \rightarrowtail^* l\tau \approx l\tau, C_1, r\tau \approx t', C_2, c\tau, H\theta \\ &\rightarrowtail \texttt{true}, C_1, r\tau \approx t', C_2, c\tau, H\theta. \end{aligned}$$

Here C_1, C_2 and C consist of the same equations in possibly different order. Hence by rearranging the steps in the remaining part of Π we obtain

$$\texttt{true}, C_1, r\tau \approx t', C_2, c\tau, H\theta \rightarrowtail^* \top.$$

Concatenating these two derivations yields the rewrite proof Π'. It remains to show that the state $\phi_{[\mathrm{o}]}(\underline{\Pi}) = \langle G', \theta', \Pi', X\rangle$ is admissible. Since $\theta'\upharpoonright_X = \theta\upharpoonright_X \cup \tau\upharpoonright_X = \theta\upharpoonright_X$, $\phi_{[\mathrm{o}]}(\underline{\Pi})$ inherits normality from $\underline{\Pi}$. For the variable condition we need some more effort. Below we make tacit use of the observation that a variable $x \in \mathcal{V}\text{ar}(s \approx t, H)$ is $\underline{\Pi}$-intermediate in G if and only if x is $\phi_{[\mathrm{o}]}(\underline{\Pi})$-intermediate in G'. First consider the equation $s \approx l$. Since $s\theta'$ is rewritten in Π', we obtain from Lemma 3.3(1) that s does not contain intermediate variables. Hence **VC1** and **VC2** are trivially satisfied. By construction, the right-hand side of $s\theta' \approx l\theta'$ is never rewritten in Π'. Hence **VC3** holds vacuously. Next consider the equations in $r \approx t, c$. Because we deal with CTRSs without extra variables, all variables in r and c occur in l and hence the three variable conditions are true for the equations in c and $r \approx t$ satisfies **VC1** and **VC2**. Suppose the right-hand side of $r\theta' \approx t\theta'$ is rewritten in Π'. By construction of Π', this is only possible if the right-hand side of $s\theta \approx t\theta$ is rewritten in Π. According to Lemma 3.3(2) t does not contain intermediate variables and thus the equation $r \approx t$ in G' satisfies **VC3**. Finally, consider an equation $s' \approx t'$ in $H = H_1, s' \approx t', H_2$. Let V_1 be the set of intermediate variables in s' and V_2 the set of intermediate variables in t'. Since $\underline{\Pi}$ is admissible, $V_1 \subseteq \mathcal{V}\text{ar}(s \approx t, H_1)$ or $V_2 \subseteq \mathcal{V}\text{ar}(s \approx t, H_1)$. Hence also $V_1 \subseteq \mathcal{V}\text{ar}(s \approx l, r \approx t, H_1)$ or $V_2 \subseteq \mathcal{V}\text{ar}(s \approx l, r \approx t, H_1)$. This proves **VC1**. The proof of **VC2** is just as easy: If $s'\theta'$ is rewritten in Π' then $s'\theta$

is rewritten in Π and thus all intermediate variables in s' occur in $s \approx t, H_1$ and therefore also in $s \approx l, r \approx t, H_1$. Finally, **VC3** is proved in exactly the same way. $\square$

Note that the above proof breaks down if we admit extra variables in the conditional rewrite rules. Further note that the proof remains valid if we would put the equation $r \approx t$ after the conditions c in G'.

Lemma 3.5 *Let $s = f(s_1, \ldots, s_n)$ and $t \in \mathcal{V}$. If $\mathrm{root}(t\theta) = f$ then there exists an admissible state $\phi_{[\mathrm{i}]}(\underline{\Pi}) = \langle G', \theta', \Pi, X' \rangle$ with $G' = G\sigma_1$ such that $\sigma_1\theta' = \theta\ [W]$. Here $\sigma_1 = \{t \mapsto f(x_1, \ldots, x_n)\}$ with $x_1, \ldots, x_n \notin W$.*

Proof Write $t\theta = f(t_1, \ldots, t_n)$ and define $\theta' = \theta \cup \{x_i \mapsto t_i \mid 1 \leqslant i \leqslant n\}$. One easily verifies that $\sigma_1\theta' = \theta\ [W]$. We have $G'\theta' = G\sigma_1\theta' = G\theta$ and thus Π is a rewrite proof of $G'\theta'$. We define $X' = \cup_{x \in X} \mathcal{V}\mathrm{ar}(x\sigma_1)$. Equivalently,

$$X' = \begin{cases} (X \setminus \{t\}) \cup \{x_1, \ldots, x_n\} & \text{if } t \in X, \\ X & \text{otherwise.} \end{cases}$$

It remains to show that $\phi_{[\mathrm{i}]}(\underline{\Pi})$ is admissible. First we show that $\theta'\restriction_{X'}$ is normalized. We consider two cases. If $t \in X$ then $\theta'\restriction_{X'} = \theta\restriction_{X\setminus\{t\}} \cup \{x_i \mapsto t_i \mid 1 \leqslant i \leqslant n\}$. The substitution $\theta\restriction_{X\setminus\{t\}}$ is normalized because $\theta\restriction_X$ is normalized. Furthermore, since every t_i is a subterm of $t\theta$ and $t\theta$ is a normal form because $\underline{\Pi}$ is normal and $t \in X$, $\{x_i \mapsto t_i \mid 1 \leqslant i \leqslant n\}$ is normalized as well. If $t \notin X$ then $\theta'\restriction_{X'} = \theta\restriction_X$ is normalized by assumption. Next we show that every equation in G' satisfies the variable condition. Again we consider two cases.

1. Suppose that $t \in X$. Then $x_1, \ldots, x_n$ belong to X' and thus the right-hand side $t\sigma_1 = f(x_1, \ldots, x_n)$ of the leftmost equation $s\sigma_1 \approx t\sigma_1$ in G' does not contain X'-intermediate variables. Hence $s\sigma_1 \approx t\sigma_1$ satisfies **VC1** and **VC3**. Suppose $s\sigma_1\theta'$ is rewritten in Π. Then, as $\underline{\Pi}$ is admissible, s does not contain X-intermediate variables. We have to show that $s\sigma_1$ does not contain X'-intermediate variables. Since the variables $x_1, \ldots, x_n$ are of interest, it follows that every X'-intermediate variable in $s\sigma_1$ is X-intermediate in s. Therefore $s\sigma_1 \approx t\sigma_1$ satisfies **VC3**.

   Next consider an equation $s'\sigma_1 \approx t'\sigma_1$ in $H\sigma_1 = (H_1, s' \approx t', H_2)\sigma_1$. Let V_1 be the set of X'-intermediate variables in $s'\sigma_1$ and V_2 the set of

X'-intermediate variables in $t'\sigma_1$. Since the variables $x_1, \ldots, x_n$ are not X'-intermediate and t is not X-intermediate, V_1 (V_2) coincides with the set of X-intermediate variables in s' (t'). Since $\underline{\Pi}$ is admissible, $V_1 \subseteq \mathcal{V}\mathrm{ar}(s \approx t, H_1)$ or $V_2 \subseteq \mathcal{V}\mathrm{ar}(s \approx t, H_1)$. Because $t \notin V_1 \cup V_2$, $V_1 \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$ or $V_2 \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$. This concludes the proof of **VC1**. Next we prove **VC2**. If $s'\sigma_1\theta' = s'\theta$ is rewritten in Π then $V_1 \subseteq \mathcal{V}\mathrm{ar}(s \approx t, H_1)$ and as $t \notin V_1$ also $V_1 \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$. The proof of **VC3** is just as easy.

2. Suppose that $t \notin X$. Then t is $\underline{\Pi}$-intermediate and $x_1, \ldots, x_n$ are $\phi_{[\mathrm{i}]}(\underline{\Pi})$-intermediate. First consider the equation $s\sigma_1 \approx t\sigma_1$. Since t is intermediate, s cannot contain intermediate variables. In particular, t does not occur in s and therefore $s\sigma_1 = s$. So $s\sigma_1 \approx t\sigma_1$ satisfies **VC1** and **VC2**. Since t is a variable, $t\sigma_1\theta' = t\theta$ cannot be rewritten in Π as a consequence of Lemma 3.3(2) and hence **VC3** is satisfied too.

   Next consider an equation $s'\sigma_1 \approx t'\sigma_1$ in $H\sigma_1 = (H_1, s' \approx t', H_2)\sigma_1$. Let V_1' (V_2') be the set of $\phi_{[\mathrm{i}]}(\underline{\Pi})$-intermediate variables in $s'\sigma_1$ ($t'\sigma_1$) and let V_1 (V_2) be the set of $\underline{\Pi}$-intermediate variables in s' (t'). We have

$$V_1' = \begin{cases} (V_1 \setminus \{t\}) \cup \{x_1, \ldots, x_n\} & \text{if } t \in \mathcal{V}\mathrm{ar}(s'), \\ V_1 & \text{otherwise,} \end{cases}$$

   and

$$V_2' = \begin{cases} (V_2 \setminus \{t\}) \cup \{x_1, \ldots, x_n\} & \text{if } t \in \mathcal{V}\mathrm{ar}(t'), \\ V_2 & \text{otherwise.} \end{cases}$$

   Because $\underline{\Pi}$ is admissible, $V_1 \subseteq \mathcal{V}\mathrm{ar}(s \approx t, H_1)$ or $V_2 \subseteq \mathcal{V}\mathrm{ar}(s \approx t, H_1)$. We consider the former. To conclude **VC1** it is sufficient to show that $V_1' \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$. We distinguish two cases. If $t \in \mathcal{V}\mathrm{ar}(s')$ then $t \in V_1$ and thus $x_1, \ldots, x_n \in \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$. Since we also have the inclusion $V_1 \setminus \{t\} \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$, $V_1' \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$ holds. If $t \notin \mathcal{V}\mathrm{ar}(s')$ then $t \notin V_1$ and thus $V_1' = V_1 \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$. This proves **VC1**. For **VC2** we reason as follows. Suppose $s'\sigma_1\theta' = s'\theta$ is rewritten in Π. This implies that $V_1 \subseteq \mathcal{V}\mathrm{ar}(s \approx t, H_1)$ and hence $V_1' \subseteq \mathcal{V}\mathrm{ar}((s \approx t, H_1)\sigma_1)$ by using similar arguments as before. The proof of **VC3** is again very similar.

□

Lemma 3.6 *Let $s = f(s_1,\dots,s_n)$, $t = f(t_1,\dots,t_n)$, and suppose that no reduct of $s\theta \approx t\theta$ in Π is rewritten at position 1 or 2. There exists an admissible state $\phi_{[\mathrm{d}]}(\underline{\Pi}) = \langle G', \theta, \Pi', X\rangle$ with $G' = s_1 \approx t_1, \dots, s_n \approx t_n, H$.*

Proof The given rewrite proof Π is of the form

$$G\theta \;\rightarrowtail^*\; f(u_1,\dots,u_n) \approx f(u_1,\dots,u_n), C, H\theta \;\rightarrowtail\; \texttt{true}, C, H\theta \;\rightarrowtail^*\; \top.$$

Here C are the instantiated conditions of the rewrite rules applied in the rewrite sequence from $s\theta \approx t\theta$ to $f(u_1,\dots,u_n) \approx f(u_1,\dots,u_n)$. The first part of Π can be transformed into

$$\begin{aligned} G'\theta' \;&\rightarrowtail^*\; u_1 \approx u_1, C_1, \dots, u_n \approx u_n, C_n, H\theta \\ &\rightarrowtail^*\; \texttt{true}, C_1, \dots, \texttt{true}, C_n, H\theta. \end{aligned}$$

Here $C_1,\dots,C_n$ and C consist of the same equations in possibly different order. Hence by rearranging the steps in the latter part of Π we obtain

$$\texttt{true}, C_1, \dots, \texttt{true}, C_n, H\theta \;\rightarrowtail^*\; \top.$$

Concatenating these two derivations yields the rewrite proof Π' of $G'\theta$. It remains to show that the state $\phi_{[\mathrm{d}]}(\underline{\Pi}) = \langle G', \theta, \Pi', X\rangle$ is admissible. Since $\underline{\Pi}$ has the same θ and X, normality is obvious. Because $\underline{\Pi}$ satisfies condition **VC1**, s or t does not contain intermediate variables. Hence there are no intermediate variables in $s_1,\dots,s_n$ or in $t_1,\dots,t_n$. Consequently, the equations $s_1 \approx t_1, \dots, s_n \approx t_n$ in G' satisfy **VC1**. The conditions **VC2** and **VC3** are also easily verified. For instance, suppose that $t_i\theta$ is rewritten in Π'. Then, by construction of Π', $t\theta$ is rewritten in Π. According to Lemma 3.3, t does not contain intermediate variables and since t_i is a subterm of t, t_i also lacks intermediate variables. By using similar arguments one easily verifies that the equations in H satisfy the three conditions. $\Box$

Lemma 3.7 *Let $t \in \mathcal{V}$, $s \neq t$, and suppose that in the first step of Π $s\theta \approx t\theta$ is rewritten at the root position. There exists an admissible state $\phi_{[\mathrm{v}]}(\underline{\Pi}) = \langle G', \theta, \Pi', X'\rangle$ with $G' = H\sigma_1$ such that $\sigma_1\theta = \theta\ [W]$. Here $\sigma_1 = \{t \mapsto s\}$.*

Proof Since $s\theta \approx t\theta$ is rewritten to $\texttt{true}$ by the rule $x \approx x \to \texttt{true}$, we must have $s\theta = t\theta$. Hence $t\sigma_1\theta = s\theta = t\theta$. For variables y different from t we have $y\sigma_1\theta = y\theta$. Hence $\sigma_1\theta = \theta\ [W]$. Since $\mathcal{V}\mathrm{ar}(H) \subseteq W$, $G'\theta = H\sigma_1\theta = H\theta$

and thus from the tail of the rewrite proof $\Pi\colon G\theta \rightarrowtail \texttt{true}, H\theta \rightarrowtail^* \top$ we can extract a rewrite proof Π' of $G'\theta$. We define $X' = \cup_{x \in X} \mathcal{V}\mathrm{ar}(x\sigma_1)$. Clearly

$$X' = \begin{cases} (X \setminus \{t\}) \cup \mathcal{V}\mathrm{ar}(s) & \text{if } t \in X, \\ X & \text{otherwise.} \end{cases}$$

It remains to show that $\phi_{[\mathrm{v}]}(\underline{\Pi})$ is admissible. First we show that $\theta\restriction_{X'}$ is normalized. We consider two cases. If $t \in X$ then $\theta\restriction_{X'} = \theta\restriction_{X \setminus \{t\} \cup \mathcal{V}\mathrm{ar}(s)}$. The substitution $\theta\restriction_{X \setminus \{t\}}$ is normalized because $\theta\restriction_X$ is normalized. If $x \in \mathcal{V}\mathrm{ar}(s)$ then $x\theta$ is a subterm of $s\theta = t\theta$. Since $t\theta$ is a normal form by the normality of $\underline{\Pi}$, so is $x\theta$. Hence $\theta\restriction_{\mathcal{V}\mathrm{ar}(s)}$ is normalized as well. If $t \notin X$ then $\theta\restriction_{X'} = \theta\restriction_X$ is normalized by assumption. Next we show that every equation in G' satisfies the variable condition. Let $s'\sigma_1 \approx t'\sigma_1$ be an equation in $H\sigma_1 = (H_1, s' \approx t', H_2)\sigma_1$. Let V_1' (V_2') be the set of X'-intermediate variables in $s'\sigma_1$ ($t'\sigma_1$) and let V_1 (V_2) be the set of intermediate variables in s' (t'). We consider two cases.

1. Suppose that $t \in X$. Then $\mathcal{V}\mathrm{ar}(s) \subseteq X'$. So the variables in $\mathcal{V}\mathrm{ar}(s)$ are not X'-intermediate and t is not X-intermediate. It follows that $V_1' = V_1$ and $V_2' = V_2$. Since $\underline{\Pi}$ is admissible, $V_1 \subseteq \mathcal{V}\mathrm{ar}(H_1)$ or $V_2 \subseteq \mathcal{V}\mathrm{ar}(H_1)$. Because $t \notin V_1 \cup V_2$, $V_1' \subseteq \mathcal{V}\mathrm{ar}(H_1\sigma_1)$ or $V_2' \subseteq \mathcal{V}\mathrm{ar}(H_1\sigma_1)$. This concludes the proof of **VC1**. The proofs of **VC2** and **VC3** also easily follow from the identities $V_1' = V_1$ and $V_2' = V_2$ and the fact that $t \notin V_1 \cup V_2$.

2. Suppose that $t \notin X$. Then t is $\underline{\Pi}$-intermediate and all variables in $\mathcal{V}\mathrm{ar}(s)$ are $\phi_{[\mathrm{v}]}(\underline{\Pi})$-intermediate. We have

$$V_1' = \begin{cases} (V_1 \setminus \{t\}) \cup \mathcal{V}\mathrm{ar}(s) & \text{if } t \in \mathcal{V}\mathrm{ar}(s'), \\ V_1 & \text{otherwise,} \end{cases}$$

   and

$$V_2' = \begin{cases} (V_2 \setminus \{t\}) \cup \mathcal{V}\mathrm{ar}(s) & \text{if } t \in \mathcal{V}\mathrm{ar}(t'), \\ V_2 & \text{otherwise.} \end{cases}$$

   Because $\underline{\Pi}$ is admissible, $V_1 \subseteq \mathcal{V}\mathrm{ar}(H_1)$ or $V_2 \subseteq \mathcal{V}\mathrm{ar}(H_1)$. We consider the latter. To conclude **VC1** it is therefore sufficient to show that $V_2' \subseteq \mathcal{V}\mathrm{ar}(H_1\sigma_1)$. We distinguish two cases. If $t \in \mathcal{V}\mathrm{ar}(t')$ then $t \in V_2$ and thus $\mathcal{V}\mathrm{ar}(s) \subseteq \mathcal{V}\mathrm{ar}(H_1\sigma_1)$. Since the inclusion $V_2 \setminus \{t\} \subseteq \mathcal{V}\mathrm{ar}(H_1\sigma_1)$

also holds, $V_2' \subseteq \mathcal{V}\mathrm{ar}(H_1\sigma_1)$ as desired. If $t \notin \mathcal{V}\mathrm{ar}(t')$ then $t \notin V_2$ and thus $V_2' = V_2 \subseteq \mathcal{V}\mathrm{ar}(H_1\sigma_1)$. This completes the proof of **VC1**. The proofs of **VC2** and **VC3** are based on similar arguments and omitted.

$\square$

Lemma 3.8 *Let $t \in \mathcal{V}$, $s = t$, and suppose that in the first step of Π $s\theta \approx t\theta$ is rewritten at the root position. There exists an admissible state $\phi_{[t]}(\underline{\Pi}) = \langle G', \theta, \Pi', X\rangle$ with $G' = H$.*

Proof The given rewrite proof Π has the form $G\theta \rightarrowtail \texttt{true}, H\theta \rightarrowtail^* \top$. From the tail of Π we extract a rewrite proof Π' of $G'\theta = H\theta$. It is easy to show that $\phi_{[t]}(\underline{\Pi})$ is admissible. $\square$

Lemma 3.9 *There exists an admissible state $\phi_{\text{swap}}(\underline{\Pi}) = \langle G', \theta, \Pi', X\rangle$ with $G' = t \approx s, H$.*

Proof The given rewrite proof Π: $(s \approx t, H)\theta \rightarrowtail^* \top$ is transformed into a rewrite proof Π' of $(t \approx s, H)\theta$ by simply swapping the two sides of every reduct of $s\theta \approx t\theta$. This clearly does not affect normality and since the variable condition is symmetric with respect the two sides of an equation it follows that $\phi_{\text{swap}}(\underline{\Pi})$ is admissible. $\square$

We want to stress that swapping different equations (as opposed to the two sides of a single equation as in the preceding lemma) does *not* preserve the variable condition. This makes a lot of sense, since if it would preserve the variable condition then we could prove strong completeness of LCNC but from [26] we already know that the LNC is not strongly complete (for the class of confluent TRSs with respect to normalized solutions).

In the proof of the main theorem below, we use induction on admissible states with respect to the well-founded order defined below. This order is essentially the same as the one used in the completeness proofs of [26].

Definition 3.10 *The complexity $|\underline{\Pi}|$ of a state $\underline{\Pi} = \langle G, \theta, \Pi, X\rangle$ is defined as the triple consisting of (1) the number of rewrite steps in Π at non-root positions, (2) the multiset $|\mathcal{MV}\mathrm{ar}(G)\theta|$, and (3) the number of occurrences of symbols different from $\approx$ and* `true` *in G. Here $\mathcal{MV}\mathrm{ar}(G)$ denotes the* *multiset of variable occurrences in G, and for any multiset $M = \{t_1, \ldots, t_n\}$ of terms, $M\theta$ and $|M|$ denote the multisets $\{t_1\theta, \ldots, t_n\theta\}$ and $\{|t_1|, \ldots, |t_n|\}$, respectively. The well-founded order $\gg$ on states is defined as follows: $\underline{\Pi_1} \gg \underline{\Pi_2}$ if $|\underline{\Pi_1}|$ $\mathsf{lex}(>, >_{\mathsf{mul}}, >)$ $|\underline{\Pi_2}|$. Here $>$ denotes the standard order on natural numbers and $>_{\mathsf{mul}}$ denotes the multiset extension of $>$, i.e., $M >_{\mathsf{mul}} N$ for finite multisets M, N if and only if there exist multisets X and Y such that $\varnothing \neq X \subseteq M$, $N = (M - X) \uplus Y$, and for every $y \in Y$ there exists an $x \in X$ with $x \succ y$; with $-$ and $\uplus$ denoting multiset difference and sum. Furthermore, $\mathsf{lex}(>, >_{\mathsf{mul}}, >)$ denotes the lexicographic product of $>$, $>_{\mathsf{mul}}$, and $>$.*

From [5] we know that $>_{\mathsf{mul}}$ inherits well-foundedness from $>$. Consequently, the lexicographic product of $>$, $>_{\mathsf{mul}}$, and $>$ is a well-founded order and hence $\gg$ is a well-founded order on states.

Lemma 3.11 *Let $\underline{\Pi}$ be a state and $\alpha \in \{\mathrm{o}, \mathrm{i}, \mathrm{d}, \mathrm{v}, \mathrm{t}\}$. We have $\underline{\Pi} \gg \phi_{[\alpha]}(\underline{\Pi})$ whenever the latter is defined. Moreover, $|\underline{\Pi}| = |\phi_{\mathrm{swap}}(\underline{\Pi})|$.*

Proof Basically the same as the proof of Lemma 20 in [26]. For $\alpha = \mathrm{o}$ we observe a decrease in the first component of $|\underline{\Pi}|$. Here it is essential that we work with $\rightarrowtail$ instead of the ordinary rewrite relation $\rightarrow$; in this way steps that take place in the conditional part of the applied rewrite rule are already accounted for in $|\underline{\Pi}|$. For $\alpha \in \{\mathrm{i}, \mathrm{d}, \mathrm{v}, \mathrm{t}\}$ the number of rewrite steps at non-root positions remains the same. For $\alpha \in \{\mathrm{i}, \mathrm{v}, \mathrm{t}\}$ the second component of $|\underline{\Pi}|$ decreases. For $\alpha = \mathrm{d}$ the second component remains the same while the third component of $|\underline{\Pi}|$ decreases. $\square$

Theorem 3.12 *Let $\mathcal{R}$ be a confluent CTRS without extra variables and G a goal. For every normalized solution θ of G there exists an* LCNC*-refutation $G \Rightarrow^*_\sigma \square$ respecting $\mathcal{S}_{\mathrm{left}}$ such that $\sigma \leqslant \theta$ $[\mathcal{V}\mathrm{ar}(G)]$.*

Proof Because $\mathcal{R}$ is confluent, $G\theta$ admits a rewrite proof Π. Consider the state $\underline{\Pi} = \langle G, \theta, \Pi, X\rangle$ with $X = \mathcal{V}\mathrm{ar}(G)$. By assumption $\theta\restriction_X$ is normalized. Since all variables of G are of interest, G does not contain intermediate variables and hence the variable condition is trivially satisfied. Therefore $\underline{\Pi}$ is admissible. We use induction on the complexity of $\underline{\Pi}$. In order to make the induction work we prove $\sigma \leqslant \theta$ $[W]$ for a finite set of variables W that includes $\mathcal{V}\mathrm{ar}(G)$. The base case is trivial since G must be the empty goal (and thus we can take $\sigma = \varepsilon$, the empty substitution). For the induction step we proceed as follows. We prove the existence of an LCNC-step $\Psi_1 \colon G \Rightarrow_{\sigma_1} G'$ that respects $\mathcal{S}_{\text{left}}$ and an admissible state $\underline{\Pi'} = \langle G', \theta', \Pi', X' \rangle$ such that $\sigma_1\theta' = \theta\ [W]$. Let $G = s \approx t, H$. We distinguish the following cases, depending on what happens to $s\theta \approx t\theta$ in Π.

1. Suppose no reduct of $s\theta \approx t\theta$ is rewritten at position 1 or 2. We distinguish five further cases.

    (a) Suppose $s, t \notin \mathcal{V}$. We may write $s = f(s_1, \ldots, s_n)$ and $t = f(t_1, \ldots, t_n)$. From Lemma 3.6 we obtain an admissible state $\phi_{[\mathrm{d}]}(\underline{\Pi}) = \langle G', \theta', \Pi', X' \rangle$ with $G' = s_1 \approx t_1, \ldots, s_n \approx t_n, H$, $\theta' = \theta$, and $X' = X$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{d}]} G'$. Take $\sigma_1 = \varepsilon$ and $\underline{\Pi'} = \phi_{[\mathrm{d}]}(\underline{\Pi})$.

    (b) Suppose $t \in \mathcal{V}$ and $s = t$. According to Lemma 3.3 no $s\theta$ and $t\theta$ are not rewritten and hence in the first step of Π $s\theta \approx t\theta$ is rewritten at the root position. Hence Lemma 3.8 is applicable, yielding an admissible state $\phi_{[\mathrm{t}]}(\underline{\Pi}) = \langle G', \theta', \Pi', X' \rangle$ with $G' = H$, $\theta' = \theta$, and $X' = X$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{t}]} G'$. Take $\sigma_1 = \varepsilon$ and $\underline{\Pi'} = \phi_{[\mathrm{t}]}(\underline{\Pi})$.

    (c) Suppose $t \in \mathcal{V}$, $s \neq t$, and a reduct of $s \approx t$ is rewritten at a non-root position. From Lemma 3.3 we infer that s is not a variable and moreover that $t\theta$ is not rewritten in Π. Hence we may write $s = f(s_1, \ldots, s_n)$ and we have $\mathrm{root}(t\theta) = f$. Consequently, Lemma 3.5 is applicable, yielding an admissible state $\phi_{[\mathrm{i}]}(\underline{\Pi}) = \langle G'', \theta'', \Pi'', X'' \rangle$ with $G'' = G\sigma_1$, $\Pi'' = \Pi$, and $\sigma_1\theta'' = \theta\ [W]$ for the substitution $\sigma = \{t \mapsto f(x_1, \ldots, x_n)\}$. We have $G'' = (f(s_1, \ldots, s_n) \approx f(x_1, \ldots, x_n), H)\sigma_1$. By assumption no reduct of $s\sigma\theta'' \approx t\sigma\theta''$ is rewritten at position 1 or 2. Hence we can apply Lemma 3.6. This yields an admissible state $\phi_{[\mathrm{d}]}(\phi_{[\mathrm{i}]}(\underline{\Pi})) = \langle G', \theta', \Pi', X' \rangle$ with $G' = (s_1 \approx x_1, \ldots, s_n \approx x_n, H)\sigma_1$, $\theta' = \theta''$, and $X' = X''$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{i}], \sigma_1} G'$ and $\sigma_1\theta' = \sigma_1\theta'' = \theta\ [W]$. Take $\underline{\Pi'} = \phi_{[\mathrm{d}]}(\phi_{[\mathrm{i}]}(\underline{\Pi}))$.

    (d) Suppose $t \in \mathcal{V}$, $s \neq t$, and the first rewrite step takes place at the root position of $s \approx t$. Lemma 3.7 yields an admissible state $\phi_{[\mathrm{v}]}(\underline{\Pi}) = \langle G', \theta', \Pi', X' \rangle$ with $G' = G\sigma$, $\Pi' = \Pi$, and $\sigma_1\theta' = \theta\ [W]$ for the substitution $\sigma_1 = \{t \mapsto s\}$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{v}], \sigma_1} G'$. Take $\underline{\Pi'} = \phi_{[\mathrm{v}]}(\underline{\Pi})$.

(e) In the remaining case we have $t \notin \mathcal{V}$ and $s \in \mathcal{V}$. This case reduces to case 1(c) or 1(d) by an appeal to Lemma 3.9.

2. Suppose a reduct of $s\theta \approx t\theta$ is rewritten at position 1. Let $l = f(l_1, \dots, l_n) \to r \Leftarrow c$ be the employed rewrite rule the first time this happens. From Lemma 3.4 we obtain an admissible state $\phi_{[o]}(\underline{\Pi}) = \langle G'', \theta'', \Pi'', X'' \rangle$ with $G'' = s \approx l, r \approx t, c, H$, $X'' = X$, and $\theta'' = \theta\ [W]$. According to Lemma 3.3, s cannot be a variable. Hence we may write $s = f(s_1, \dots, s_n)$. Let $G' = s_1 \approx l_1, \dots, s_n \approx l_n, r \approx t, c, H$. We have $\Psi_1 \colon G \Rightarrow_{[o]} G''$. Note that Lemma 3.6 is applicable to $\phi_{[o]}(\underline{\Pi})$ since by construction no reduct of $s\theta'' \approx l\theta''$ is rewritten at position 1 and 2. This results in an admissible state $\phi_{[d]}(\phi_{[o]}(\underline{\Pi})) = \langle G', \theta', \Pi', X' \rangle$ with $\theta' = \theta''$ and $X' = X$. Clearly $\theta' = \theta\ [W]$. Take $\sigma_1 = \varepsilon$ and $\underline{\Pi'} = \phi_{[d]}(\phi_{[o]}(\underline{\Pi}))$.

3. Suppose a reduct of $s\theta \approx t\theta$ is rewritten at position 2. This case reduces to the previous one by an appeal to Lemma 3.9.

In all cases we obtain $\underline{\Pi'}$ from $\underline{\Pi}$ by applying one or two transformation steps $\phi_{[o]}$, $\phi_{[i]}$, $\phi_{[d]}$, $\phi_{[v]}$, $\phi_{[t]}$ together with an additional application of ϕ_{swap} in case 1(e) and 3. According to Lemma 3.11 $\underline{\Pi'}$ has smaller complexity than $\underline{\Pi}$. Let $W' = \mathcal{V}\text{ar}_W(\sigma_1) \cup \mathcal{V}\text{ar}(G')$. We have $\mathcal{V}\text{ar}(G') \subseteq W'$ and thus we can apply the induction hypothesis to $\underline{\Pi'}$. This yields an LCNC-refutation $\Psi' \colon G' \Rightarrow^*_{\sigma'} \square$ respecting $\mathcal{S}_{\text{left}}$ such that $\sigma' \leqslant \theta'\ [W']$. Define $\sigma = \sigma_1\sigma'$. From $\sigma_1\theta' = \theta\ [W]$, $\sigma' \leqslant \theta'\ [W']$, and $\mathcal{V}\text{ar}_W(\sigma_1) \subseteq W'$ we infer that $\sigma \leqslant \theta\ [W]$. Concatenating the LCNC-step Ψ_1 and the LCNC-refutation Ψ' yields the desired LCNC-refutation Ψ. □

An immediate corollary of Theorem 3.12 is the completeness of LNC for confluent TRSs and normalized solutions with respect to leftmost selection. We remark that the proof in [26] of this result is considerably more complicated than the proof given above.

4 Strong Completeness

In this section we prove that LCNC is strongly complete whenever basic conditional narrowing is complete. We obtain this result by an inductive transformation process that operates on basic conditional narrowing sequences. To this end we extend the proof of Middeldorp, Okui, and Ida [26] who obtained this result for unconditional TRSs. The structure of the proof is exactly the same as the one in [26, Section 4]. Below we state the relevant lemmata and we present complete proof details of some of the lemmata. The proofs of the other lemmata are straightforward modifications of the corresponding ones in [26]. But first we recall a number of definitions pertaining to (basic) conditional narrowing.

The conditional narrowing calculus CNC consists of the following inference rule:

$$\frac{G', e, G''}{(G', e[r]_p, c, G'')\theta}$$

if there exist a fresh variant $l \to r \Leftarrow c$ of a rewrite rule in $\mathcal{R}$, a non-variable position p in e, and a most general unifier θ of $e_{|p}$ and l.

In the above situation we write $G', e, G'' \rightsquigarrow_\theta (G', e[r]_p, c, G'')\theta$. For a CNC-derivation $\Pi\colon G \rightsquigarrow_\theta^* G'$, $\Pi\theta$ denotes the corresponding rewrite sequence $G\theta \to^* G'$. Conditional narrowing is sound: If $G \rightsquigarrow_\theta^* \top$ then $\theta\restriction_{\mathcal{V}\mathrm{ar}(G)}$ is a solution of G.

A position constraint for a goal G is a mapping that assigns to every equation $e \in G$ a subset of $\mathcal{P}\mathrm{os}_\mathcal{F}(e)$. The position constraint that assigns to every $e \in G$ the set $\mathcal{P}\mathrm{os}_\mathcal{F}(e)$ is denoted by $\bar{G}$. A CNC-derivation Π:

$$G_1 \rightsquigarrow_{\theta_1, e_1, p_1, l_1 \to r_1 \Leftarrow c_1} \cdots \rightsquigarrow_{\theta_{n-1}, e_{n-1}, p_{n-1}, l_{n-1} \to r_{n-1} \Leftarrow c_{n-1}} G_n$$

is based on a position constraint B_1 for G_1 if $p_i \in B_i(e_i)$ for $1 \leqslant i \leqslant n-1$. Here the position constraints $B_2, \ldots, B_{n-1}$ for the goals $G_2, \ldots, G_{n-1}$ are inductively defined by

$$B_{i+1}(e) = \begin{cases} B_i(e') & \text{if } e' \in G_i \setminus \{e_i\} \\ \mathcal{B}(B_i(e_i), p_i, r_i) & \text{if } e' = e_i[r_i]_{p_i} \\ \mathcal{P}\mathrm{os}_\mathcal{F}(e') & \text{if } e' \in c_i \end{cases}$$

for all $1 \leqslant i < n-1$ and $e = e'\theta_i \in G_{i+1}$, with $\mathcal{B}(B_i(e_i), p_i, r_i)$ abbreviating the set of positions

$$(B_i(e_i) \setminus \{q \in B_i(e_i) \mid q \geqslant p_i\}) \cup \{p_i q \in \mathcal{P}\mathrm{os}_\mathcal{F}(e) \mid q \in \mathcal{P}\mathrm{os}_\mathcal{F}(r_i)\}.$$

We say that Π is basic if it is based on $\bar{G}$. So in a basic CNC-derivation no steps take place at subterms introduced by previous narrowing substitutions. Basic CNC has a much smaller search space than CNC. We refer to Middeldorp and Hamoen [24] for a survey of completeness results for CNC and basic CNC.

Definition 4.1 *Let $G \rightsquigarrow_{\theta, p, l \to r \Leftarrow c} G_1$ be a* CNC*-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_1. In this case we say that $e[r]_p\theta$ is the descendant of e in G_1. Otherwise, e is simply instantiated to the equation $e\theta$ in G_1 and we call $e\theta$ the descendant of e. The notion of descendant extends to* CNC*-derivations in the obvious way.*

Observe that in a CNC-refutation $G \rightsquigarrow^* \top$ every equation $e \in G$ has exactly one descendant `true` in $\top$, but, unlike the unconditional case, not every `true` in $\top$ descends from an equation in G.

Lemma 4.2 (12) *Let δ be a variable renaming. For every* CNC*-refutation $\Pi\colon G \rightsquigarrow^+_\theta \top$ there exists a* CNC*-refutation $\varphi_\delta(\Pi)\colon G\delta \rightsquigarrow^+_{\delta^{-1}\theta} \top$.* $\square$

The number between parentheses refers to the corresponding statement in [26].

In the following five lemmata Π denotes a CNC-refutation $G \rightsquigarrow^+_\theta \top$ with $G = G', s \approx t, G''$ such that $s \approx t$ is selected in the first step of Π and W denotes a finite set of variables that includes all variables in the initial goal G of Π. The first transformation lemma depends on the applied rewrite rule. So in the proof we have to take its conditional part into account.

Lemma 4.3 (13) *Suppose narrowing is applied to a descendant of $s \approx t$ in Π at position 1. If $l \to r \Leftarrow c$ is the applied rewrite rule in the first such step then there exists a* CNC*-refutation $\varphi_{[\mathrm{o}]}(\Pi)\colon G', s \approx l, r \approx t, c, G'' \rightsquigarrow^*_{\theta_1} \top$ such that $\theta_1 = \theta$ $[W]$.*

Proof Write $l = f(l_1, \ldots, l_n)$. The given refutation Π is of the form

$$\begin{aligned} G \ &\rightsquigarrow^*_{\tau_1} \ G_1', f(u_1, \ldots, u_n) \approx t', G_1'' \ \rightsquigarrow_{\tau_2, 1, l \to r \Leftarrow c} \ (G_1', r \approx t', c, G_1'')\tau_2 \\ &\rightsquigarrow^*_{\tau_3} \ \top \end{aligned}$$

with $\tau_1\tau_2\tau_3 = \theta$. Write $G_1'' = C, G_2''$ such that $s \approx t \rightsquigarrow^* f(u_1, \ldots, u_n) \approx t', C$ and $G'' \rightsquigarrow^* G_2''$. So C consists of all descendants of the (instantiated) equations that appear in the conditional parts of the applied rewrite rules in the derivation from $s \approx t$ to $f(u_1, \ldots, u_n) \approx t'$. The first part of Π can be transformed into

$$G', s \approx l, r \approx t, c, G'' \ \rightsquigarrow^*_{\tau_1} \ G_1', f(u_1, \ldots, u_n) \approx l, C_1, r \approx t', C_2, c, G_2''$$

where $C_1, C_2 = C$. The last part of Π can be rearranged into

$$(G_1', C_1, r \approx t', C_2, c, G_2'')\tau_2 \;\rightsquigarrow^*_{\tau_3}\; \top.$$

Let x be a fresh variable (so $x \notin W$) and define the substitution υ_2 as the (disjoint) union of τ_2 and $\{x \mapsto l\tau_2\}$. Because υ_2 is a most general unifier of $f(u_1, \ldots, u_n) \approx l$ and $x \approx x$, the CNC-derivation Π can be transformed into the refutation $\varphi_{[\mathrm{o}]}(\Pi)$:

$$\begin{array}{rcl} G', s \approx l, r \approx t, c, G'' & \rightsquigarrow^*_{\tau_1} & G_1', f(u_1, \ldots, u_n) \approx l, C_1, r \approx t', C_2, c, G_2'' \\ & \rightsquigarrow_{\upsilon_2} & (G_1', \mathtt{true}, C_1, r \approx t', C_2, c, G_2'')\upsilon_2 \\ & = & (G_1', \mathtt{true}, C_1, r \approx t', C_2, c, G_2'')\tau_2 \\ & \rightsquigarrow^*_{\tau_3} & \top. \end{array}$$

Let $\theta_1 = \tau_1\upsilon_2\tau_3$. We have $\theta_1 = \theta \cup \{x \mapsto l\tau_2\tau_3\}$ and because $x \notin W$ we obtain $\theta_1 = \theta\ [W]$. $\square$

For the tedious proof of the next lemma we refer to the appendix.

Lemma 4.4 (14) *Let $s = f(s_1, \ldots, s_n)$ and $t \in \mathcal{V}$. If $\mathrm{root}(t\theta) = f$ then there exists a* CNC*-refutation $\varphi_{[\mathrm{i}]}(\Pi)\colon G\sigma_1 \rightsquigarrow^*_{\theta_1} \top$ such that Π subsumes $\varphi_{[\mathrm{i}]}(\Pi)$, $\Pi\theta = \varphi_{[\mathrm{i}]}(\Pi)\theta_1$, and $\sigma_1\theta_1 = \theta\ [W]$. Here $\sigma_1 = \{t \mapsto f(x_1, \ldots, x_n)\}$ with $x_1, \ldots, x_n \notin W$.* $\square$

Lemma 4.5 (15) *Let $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$, and suppose that narrowing is never applied to a descendant of $s \approx t$ in Π at position 1 or 2. There exists a* CNC*-refutation $\varphi_{[\mathrm{d}]}(\Pi)\colon G', s_1 \approx t_1, \ldots, s_n \approx t_n, G'' \rightsquigarrow^*_{\theta_1} \top$ such that $\theta_1 \leqslant \theta\ [W]$.*

Proof The given refutation Π must be of the form

$$G \;\rightsquigarrow^*_{\tau_1}\; G_1', s' \approx t', G_1'' \;\rightsquigarrow_{\tau_2,\epsilon}\; (G_1', \mathtt{true}, G_1'')\tau_2 \;\rightsquigarrow^*_{\tau_3}\; \top$$

with $s' = f(s_1', \ldots, s_n')$, $t' = f(t_1', \ldots, t_n')$, and $\tau_1\tau_2\tau_3 = \theta$. Write $G_1'' = C, G_2''$ such that $s \approx t \rightsquigarrow^* \mathtt{true}, C$ (and thus $G'' \rightsquigarrow^* G_2''$). The first part of Π can be transformed into Π_1:

$$G', s_1 \approx t_1, \ldots, s_n \approx t_n, G'' \;\rightsquigarrow^*_{\tau_1}\; G_1', s_1' \approx t_1', C_1, \ldots, s_n' \approx t_n', C_n, G_2''.$$

Here $C_1, \ldots, C_n$ and C consists of the same equations in possibly different order. Consider the step from $G_1', s' \approx t', G_1''$ to $(G_1', \mathtt{true}, G_1'')\tau_2$. Let $x \approx$

$x \rightarrow \texttt{true}$ be the employed rewrite rule, so τ_2 is a most general unifier of $x \approx x$ and $s' \approx t'$. There clearly exists a rewrite sequence

$$\begin{aligned}&(G_1', s_1' \approx t_1', C_1, \ldots, s_n' \approx t_n', C_n, G_2'')\tau_2\\ &\quad \rightarrow_\epsilon^* \ (G_1', \texttt{true}, C_1, \ldots, \texttt{true}, C_n, G_2'')\tau_2.\end{aligned}$$

Since all steps in this rewrite sequence take place at root positions using the unconditional rewrite rule $x \approx x \rightarrow \texttt{true}$, lifting can be applied, resulting in a CNC-derivation Π_2:

$$\begin{aligned}&G_1', s_1' \approx t_1', C_1, \ldots, s_n' \approx t_n', C_n, G_2''\\ &\quad \rightsquigarrow_{\upsilon_2,\epsilon}^* \ (G_1', \texttt{true}, C_1, \ldots, \texttt{true}, C_n, G_2'')\upsilon_2\end{aligned}$$

such that $\upsilon_2 \leqslant \tau_2$ $[W \cup \mathcal{I}(\tau_1)]$. We distinguish two cases.

1. Suppose $G_1', G_1'' = \square$. In this case $\tau_3 = \varepsilon$. We simply define $\varphi_{[\mathrm{d}]}(\Pi) = \Pi_1; \Pi_2$. Note that $(G_1', \texttt{true}, C_1, \ldots, \texttt{true}, C_n, G_2'')\upsilon_2 = \top$. Let $\theta_1 = \tau_1\upsilon_2$. From $\upsilon_2 \leqslant \tau_2$ $[W \cup \mathcal{I}(\tau_1)]$ we infer that $\theta_1 \leqslant \tau_1\tau_2 = \theta$ $[W]$.

2. The case $G_1', G_1'' \neq \square$ is more involved. First observe that υ_2 is a unifier of s' and t'. Using the fact that τ_2 is a most general unifier of $s' \approx t'$ and $x \approx x$, it is not difficult to show that $\tau_2 \leqslant \upsilon_2$ $[\mathcal{V} \setminus \{x\}]$. Since $x \notin W \cup \mathcal{I}(\tau_1)$ we have in particular $\tau_2 \leqslant \upsilon_2$ $[W \cup \mathcal{I}(\tau_1)]$. It follows that there exists a variable renaming δ such that $\upsilon_2 = \tau_2\delta$ $[W \cup \mathcal{I}(\tau_1)]$. Clearly $\mathcal{V}\mathrm{ar}(G_1', G_1'') \subseteq W \cup \mathcal{I}(\tau_1)$. The last part of Π can be transformed (by changing the number of occurrences of $\texttt{true}$ as well as the order of the equations in each goal) into

$$\Pi_3\colon (G_1', \texttt{true}, C_1, \ldots, \texttt{true}, C_n, G_2'')\tau_2 \ \rightsquigarrow_{\tau_3}^+ \ \top.$$

An application of Lemma 4.2 results in the CNC-refutation

$$\varphi_\delta(\Pi_3)\colon (G_1', \texttt{true}, C_1, \ldots, \texttt{true}, C_n, G_2'')\upsilon_2 \ \rightsquigarrow_{\delta^{-1}\tau_3}^+ \ \top.$$

Define $\varphi_{[\mathrm{d}]}(\Pi) = \Pi_1; \Pi_2; \varphi_\delta(\Pi_3)$. Let $\theta_1 = \tau_1\upsilon_2\delta^{-1}\tau_3$. We have $\theta_1 = \tau_1\tau_2\tau_3 = \theta$ $[W]$.

□

The proofs of the two remaining transformation lemmata are exactly the same as the unconditional ones.

Lemma 4.6 (16) *Let $t \in \mathcal{V}$, $t \notin \mathcal{V}\mathrm{ar}(s)$, and suppose that the first step of* Π *takes place at the root position. There exists a* CNC*-refutation* $\varphi_{[\mathrm{v}]}(\Pi)\colon (G', G'')\sigma_1 \rightsquigarrow^*_{\theta_1} \top$ *with* $\sigma_1 = \{t \mapsto s\}$ *such that* $\sigma_1\theta_1 \leqslant \theta\ [W]$. □

Lemma 4.7 (17) *Let* $t \in \mathcal{V}$, $s = t$, *and suppose that the first step of* Π *takes place at the root position. There exists a* CNC*-refutation* $\varphi_{[\mathrm{t}]}(\Pi)\colon G', G'' \rightsquigarrow^*_{\theta_1} \top$ *such that* $\theta_1 \leqslant \theta\ [W]$. □

Definition 4.8 *The complexity* $|\Pi|$ *of a* CNC*-refutation* $\Pi\colon G \rightsquigarrow^*_\theta \top$ *and the well-founded order* $\gg$ *on* CNC*-refutations are defined as in Definition 3.10. The only difference is that in part (1) of the complexity measure we count the number of* narrowing *steps at non-root positions in* Π.

Lemma 4.9 (20) *Let* Π *be a* CNC*-refutation and* $\alpha \in \{\mathrm{o}, \mathrm{i}, \mathrm{d}, \mathrm{v}, \mathrm{t}\}$. *We have* $\Pi \gg \varphi_{[\alpha]}(\Pi)$ *whenever* $\varphi_{[\alpha]}(\Pi)$ *is defined.* □

Lemma 4.10 (21) *For every* CNC*-refutation* $\Pi\colon G', s \approx t, G'' \rightsquigarrow^*_\theta \top$ *there exists a* CNC*-refutation* $\varphi_{\mathrm{swap}}(\Pi)\colon G', t \approx s, G'' \rightsquigarrow^*_\theta \top$ *with the same complexity.* □

Lemma 4.11 (24) *Let* Π *be a basic* CNC*-refutation and* $\alpha \in \{\mathrm{o}, \mathrm{i}, \mathrm{d}, \mathrm{v}, \mathrm{t}\}$. *The* CNC*-refutation* $\varphi_{[\alpha]}(\Pi)$ *is basic whenever it is defined.* □

Lemma 4.12 is the key lemma. It can be diagrammatically depicted as follows:

$$\begin{array}{llllll} \forall & \text{basic} & \Pi & : & G & \rightsquigarrow^+_\theta \ \top \\ \exists & & \Psi_1 & : & \Downarrow_{\sigma_1} & \\ \exists & \text{basic} & \Pi_1 & : & G_1 & \rightsquigarrow^*_{\theta_1} \ \top \end{array} \quad \text{such that} \quad \left\{ \begin{array}{l} \sigma_1\theta_1 \leqslant \theta\ [W] \\ \Pi \gg \Pi_1 \end{array} \right.$$

Although the proof is very similar to that of Lemma 25 in [26], we present the complete proof in order to show how the previous lemmata are used.

Lemma 4.12 (25) *For every basic* CNC*-refutation* $\Pi\colon G \rightsquigarrow^+_\theta \top$ *there exist an* LCNC*-step* $\Psi_1\colon G \Rightarrow_{\sigma_1} G_1$ *and a basic* CNC*-refutation* $\Pi_1\colon G_1 \rightsquigarrow^*_{\theta_1} \top$ *such that* $\sigma_1\theta_1 \leqslant \theta\ [W]$, $\Pi \gg \Pi_1$, *and the equation selected in the first step of* Π *is selected in* Ψ_1.

Proof We distinguish the following cases, depending on what happens to the selected equation $e = s \approx t$ in the first step of Π. Let $G = G', e, G''$.

1. Suppose narrowing is never applied to a descendant of $s \approx t$ at position 1 or 2. We distinguish four further cases.

   (a) Suppose $s, t \notin \mathcal{V}$. We may write $s = f(s_1, \dots, s_n)$ and $t = f(t_1, \dots, t_n)$. Let $G_1 = G', s_1 \approx t_1, \dots, s_n \approx t_n, G''$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{d}]} G_1$. Lemma 4.5 yields a CNC-refutation $\varphi_{[\mathrm{d}]}(\Pi) \colon G_1 \rightsquigarrow^*_{\theta_1} \top$ such that $\theta_1 \leqslant \theta\ [W]$. Take $\sigma_1 = \varepsilon$.

   (b) Suppose $t \in \mathcal{V}$ and $s = t$. In this case the first step of Π_1 must take place at the root of e. Let $G_1 = G', G''$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{t}]} G_1$. Lemma 4.7 yields a CNC-refutation $\varphi_{[\mathrm{t}]}(\Pi) \colon G_1 \rightsquigarrow^*_{\theta_1} \top$ such that $\theta_1 \leqslant \theta\ [W]$. Take $\sigma_1 = \varepsilon$.

   (c) Suppose $t \in \mathcal{V}$ and $s \neq t$. We distinguish two further cases, depending on what happens to e in the first step of Π.

      i. Suppose narrowing is applied to e at the root position. Let $\sigma_1 = \{t \mapsto s\}$ and $G_1 = (G', G'')\sigma_1$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{v}], \sigma_1} G_1$. Lemma 4.6 yields a CNC-refutation $\varphi_{[\mathrm{v}]}(\Pi) \colon G_1 \rightsquigarrow^*_{\theta_1} \top$ such that $\sigma_1\theta_1 \leqslant \theta\ [W]$.

      ii. Suppose narrowing is not applied to e at the root position. This implies that $s \notin \mathcal{V}$. Hence we may write $s = f(s_1, \dots, s_n)$. Let $\sigma_1 = \{t \mapsto f(x_1, \dots, x_n)\}$, $G_1 = (G', s_1 \approx x_1, \dots, s_n \approx x_n, G'')\sigma_1$, and $G_2 = G\sigma_1$. Here $x_1, \dots, x_n$ are fresh variables. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{i}], \sigma_1} G_1$. From Lemma 4.4 we obtain a CNC-refutation $\Pi_2 = \varphi_{[\mathrm{i}]}(\Pi) \colon G_2 \rightsquigarrow^*_{\theta_2} \top$ such that $\sigma_1\theta_2 = \theta\ [W]$. Let $W' = W \cup \{x_1, \dots, x_n\}$. Clearly $\mathcal{V}\mathrm{ar}(G_2) \subseteq W'$. An application of Lemma 4.5 to Π_2 results in a CNC-refutation $\Pi_1 = \varphi_{[\mathrm{d}]}(\Pi_2) \colon G_1 \rightsquigarrow^*_{\theta_1} \top$ such that $\theta_1 \leqslant \theta_2\ [W']$. Using the inclusion $\mathcal{V}\mathrm{ar}_W(\sigma_1) \subseteq W'$ we obtain $\sigma_1\theta_1 \leqslant \sigma_1\theta_2 = \theta\ [W]$.

   (d) In the remaining case we have $t \notin \mathcal{V}$ and $s \in \mathcal{V}$. This case reduces to case 1(c) by an appeal to Lemma 4.10.

2. Suppose narrowing is applied to a descendant of e at position 1. Let $l = f(l_1, \dots, l_n) \to r \Leftarrow c$ be the used rewrite rule the first time this happens. Because Π is basic, s cannot be a variable, for otherwise narrowing would be applied to a subterm introduced by previous narrowing substitutions. Hence we may write $s = f(s_1, \dots, s_n)$. Let $G_1 = G', s_1 \approx l_1, \dots, s_n \approx l_n, r \approx t, c, G''$ and $G_2 = G', s \approx l, r \approx t, c, G''$. We have $\Psi_1 \colon G \Rightarrow_{[\mathrm{o}]} G_1$. From Lemma 4.3 we obtain a

CNC-refutation $\Pi_2 = \varphi_{[o]}(\Pi)\colon G_2 \rightsquigarrow^*_{\theta_2} \top$ such that $\theta_2 = \theta\ [W]$. Let $W' = W \cup \mathcal{V}\mathrm{ar}(l \to r \Leftarrow c)$. Clearly $\mathcal{V}\mathrm{ar}(G_2) \subseteq W'$. An application of Lemma 4.5 to Π_2 results in a CNC-refutation $\Pi_1 = \varphi_{[d]}(\Pi_2)\colon G_1 \rightsquigarrow^*_{\theta_1} \top$ such that $\theta_1 \leqslant \theta_2\ [W']$. Using $W \subseteq W'$ we obtain $\theta_1 \leqslant \theta\ [W]$. Take $\sigma_1 = \varepsilon$.

3. Suppose narrowing is applied to a descendant of e at position 2. This case reduces to the previous one by an appeal to Lemma 4.10.

In all cases we obtain Π_1 from Π by applying one or two transformation steps $\varphi_{[o]}$, $\varphi_{[i]}$, $\varphi_{[d]}$, $\varphi_{[v]}$, $\varphi_{[t]}$ together with an additional application of φ_{swap} in case 1(d) and (3). According to Lemma 4.11 Π_1 is basic. According to Lemmata 4.9 and 4.10 Π_1 has smaller complexity than Π. □

The proof of the following switching lemma can be found in the appendix.

Lemma 4.13 (26) *Let G_1 be a goal containing distinct equations e_1 and e_2. For every* CNC*-derivation $G_1 \rightsquigarrow_{\tau_1, e_1, p_1, l_1 \to r_1 \Leftarrow c_1} G_2 \rightsquigarrow_{\tau_2, e_2\tau_1, p_2, l_2 \to r_2 \Leftarrow c_2} G_3$ with $p_2 \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(e_2)$ there exists a* CNC*-derivation $G_1 \rightsquigarrow_{\upsilon_2, e_2, p_2, l_2 \to r_2 \Leftarrow c_2} H_2 \rightsquigarrow_{\upsilon_1, e_1\upsilon_2, p_1, l_1 \to r_1 \Leftarrow c_1} H_3$ such that $G_3 = H_3$ and $\tau_1\tau_2 = \upsilon_2\upsilon_1$.* □

Lemma 4.14 (27) *Let $\mathcal{S}$ be an arbitrary selection function. For every basic* CNC*-refutation $\Pi\colon G \rightsquigarrow^*_\theta \top$ there exists a basic* CNC*-refutation $\varphi_{\mathcal{S}}(\Pi)\colon G \rightsquigarrow^*_\theta \top$ respecting $\mathcal{S}$ with the same complexity.* □

Theorem 4.15 (28) *Let $\mathcal{R}$ be an arbitrary CTRS and $\Pi\colon G \rightsquigarrow^*_\theta \top$ a basic* CNC*-refutation. For every selection function $\mathcal{S}$ there exists an* LCNC*-refutation $\Psi\colon G \Rightarrow^*_\sigma \square$ respecting $\mathcal{S}$ such that $\sigma \leqslant \theta\ [\mathcal{V}\mathrm{ar}(G)]$.*

Proof We use well-founded induction on the complexity of the given basic CNC-refutation Π. In order to make the induction work we prove $\sigma \leqslant \theta\ [W]$ for a finite set of variables W that includes $\mathcal{V}\mathrm{ar}(G)$ instead of $\sigma \leqslant \theta\ [\mathcal{V}\mathrm{ar}(G)]$. The base case is trivial: G must be the empty goal. For the induction step we proceed as follows. First we use Lemma 4.14 to transform Π into a basic CNC-refutation $\varphi_{\mathcal{S}}(\Pi)\colon G \rightsquigarrow^+_\theta \top$ respecting $\mathcal{S}$ with equal complexity. According to Lemma 4.12 there exist an LCNC-step $\Psi_1\colon G \Rightarrow_{\sigma_1} G_1$ respecting $\mathcal{S}$ and a basic CNC-refutation $\Pi_1\colon G_1 \rightsquigarrow^*_{\theta_1} \top$ such that $\sigma_1\theta_1 \leqslant \theta\ [W]$ and $\varphi_{\mathcal{S}}(\Pi) \gg \Pi_1$. Let $W' = \mathcal{V}\mathrm{ar}_W(\sigma_1) \cup \mathcal{V}\mathrm{ar}(G_1)$. Clearly W' is a finite set of variables that includes $\mathcal{V}\mathrm{ar}(G_1)$. The induction hypothesis yields an LCNC-refutation $\Psi'\colon G_1 \Rightarrow^*_{\sigma'} \square$ respecting $\mathcal{S}$ such that $\sigma' \leqslant \theta_1\ [W']$. Now define

$\sigma = \sigma_1\sigma'$. From $\sigma_1\theta_1 \leqslant \theta\ [W]$, $\sigma' \leqslant \theta_1\ [W']$, and $\mathcal{V}\mathrm{ar}_W(\sigma_1) \subseteq W'$, we infer that $\sigma \leqslant \theta\ [W]$ and thus also $\sigma \leqslant \theta\ [\mathcal{V}\mathrm{ar}(G)]$. Concatenating the LCNC-step Ψ_1 and the LCNC-refutation Ψ' yields the desired LCNC-refutation Ψ. $\square$

Since basic conditional narrowing is known to be complete for decreasing and confluent CTRSs ([24]), level-complete 2-CTRSs ([9, 24]), and terminating and shallow-confluent normal 3-CTRSs ([33, Satz 9.7]), we obtain as corollary the strong completeness of LCNC for these three classes of CTRSs.

Let us illustrate the above proof on a small example.

Example 4.16 *Consider the CTRS $\mathcal{R}$ consisting of the following rewrite rules:*

$$\begin{array}{rcl} \mathsf{f}(\mathsf{g}(x)) & \to & x \ \Leftarrow\ x \approx \mathsf{b} \\ \mathsf{a} & \to & \mathsf{b} \end{array}$$

This CTRS is easily seen to be decreasing (take e.g. the recursive path order with precedence $\mathsf{f} > \mathsf{b}$ *and* $\mathsf{a} > \mathsf{b}$*) and confluent. Consider the goal $G = \mathsf{g}(\mathsf{f}(x)) \approx x$. The substitution $\theta = \{x \mapsto \mathsf{g}(\mathsf{a})\}$ is a solution of G because we have the rewrite sequence*

$$G\theta = \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{a}) \ \to\ \mathsf{g}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a}) \ \to\ \mathtt{true}.$$

Basic conditional narrowing computes the solution $\{x \mapsto \mathsf{g}(\mathsf{b})\}$ which is different from θ but equal to θ modulo (the equational theory induced by) $\mathcal{R}$:

$$\begin{array}{l} \mathsf{g}(\underline{\mathsf{f}(x)}) \approx x \ \rightsquigarrow_{\{x \mapsto \mathsf{g}(x_1)\}}\ \underline{\mathsf{g}(x_1) \approx \mathsf{g}(x_1)},\ x_1 \approx \mathsf{b} \ \rightsquigarrow\ \mathtt{true},\ \underline{x_1 \approx \mathsf{b}} \\ \qquad\qquad \rightsquigarrow_{\{x_1 \mapsto \mathsf{b}\}}\ \top \end{array}$$

Starting from this basic narrowing refutation, the above proof yields the following LCNC*-refutation; the numbers on the right refer to the various cases* *in the proof:*

$$\begin{array}{cll}
\mathsf{g}(\mathsf{f}(x)) \approx x & & \\
\Downarrow_{[\mathrm{i}]},\ \{x \mapsto \mathsf{g}(x_1)\} & & \textit{1(c)ii} \\
\mathsf{f}(\mathsf{g}(x_1)) \approx x_1 & & \\
\Downarrow_{[\mathrm{o}]},\ \mathsf{f}(\mathsf{g}(x_2)) \to x_2 \Leftarrow x_2 \approx \mathsf{b} & & \textit{2} \\
\mathsf{g}(x_1) \approx \mathsf{g}(x_2),\ x_2 \approx x_1,\ x_2 \approx \mathsf{b} & & \\
\Downarrow_{[\mathrm{d}]} & & \textit{1(a)} \\
x_1 \approx x_2,\ x_2 \approx x_1,\ x_2 \approx \mathsf{b} & & \\
\Downarrow_{[\mathrm{v}]},\ \{x_1 \mapsto x_2\} & & \textit{1(c)i} \\
x_2 \approx x_2,\ x_2 \approx \mathsf{b} & & \\
\Downarrow_{[\mathrm{t}]} & & \textit{1(b)} \\
x_2 \approx \mathsf{b} & & \\
\Downarrow_{[\mathrm{v}]},\ \{x_2 \mapsto \mathsf{b}\} & & \textit{1(c)i} \\
\square & &
\end{array}$$

In every step of this refutation the leftmost equation is selected.

The following variant of a well-known example due to Giovannetti and Moiso [9] shows that LCNC is not (strongly) complete for terminating and confluent 2-CTRSs, even under the additional normality restriction.

Example 4.17 *Consider the 2-CTRS consisting of the following rewrite rules:*

$$\begin{array}{rcl}
\mathsf{a} & \to & \mathsf{b} \\
\mathsf{a} & \to & \mathsf{c} \\
\mathsf{b} & \to & \mathsf{c} \ \Leftarrow\ \mathsf{f}(x,\mathsf{b}) \approx \mathsf{d},\ \mathsf{f}(x,\mathsf{c}) \approx \mathsf{d} \\
\mathsf{f}(\mathsf{b},\mathsf{b}) & \to & \mathsf{d} \\
\mathsf{f}(\mathsf{c},\mathsf{c}) & \to & \mathsf{d}
\end{array}$$

Confluence and termination of $\mathcal{R}$ are easily shown. The goal $\mathsf{b} \approx \mathsf{c}$ *admits the empty substitution as solution but one easily shows that* LCNC *cannot solve the goal* $\mathsf{b} \approx \mathsf{c}$.

It is interesting to note that the previous first-order example refutes the completeness of Prehofer's conditional lazy narrowing calculus CLN for weakly normalizing and confluent higher-order normal CTRSs [27, Theorem 3.2].[1]

5 Level-Complete CTRSs

In this section we present our third and final completeness result, for the class of level-complete conditional (3-)CTRSs. Note that this class is not covered by the results of the preceding section because of the incompleteness of basic conditional narrowing (see Example 5.4 below).

Definition 5.1 *Let $\mathcal{R}$ be a CTRS and G a goal with solution θ. The level* $\mathsf{level}(e)$ *of an equation $e \in G\theta$ is the smallest n such that $e \rightarrow_n^* \mathtt{true}$. We say that the rewrite sequence $G\theta \rightarrow_{\mathcal{R}}^* \top$ is* level-minimal *if the depths of its rewrite steps do not exceed the level of the originating equations. Clearly, every rewrite sequence $G\theta \rightarrow_{\mathcal{R}}^* \top$ can be transformed into a level-minimal one (which may be longer than the given sequence).*

We use induction with respect to the well-founded order on rewrite sequences defined below.

Definition 5.2 *Let $\mathcal{R}$ be a level-terminating CTRS. For every $n \geqslant 0$, let $\mathcal{G}_n = \{G \mid \mathcal{R}_n \vdash G\}$ be the set of all goals G whose level does not exceed n. Let G be a goal and θ a substitution such that $G\theta \in \mathcal{G}_n$. With every equation $e = s \approx t \in G\theta$ we associate the pair $|e| = (\mathsf{level}(e), \{s,t\})$ whose second component is the multiset which contains both sides of e. We equip the set of these pairs with the order $\sqsupset^n = \mathsf{lex}(>, \succ^n_{\mathsf{mul}})$ where $\succ^n = (\rightarrow_n \cup \rhd)^+$. The multiset consisting of all $|e\theta|$ for $e \in G$ is denoted by $(G;\theta)$. Let $\Pi\colon G \rightarrow^* H$ and $\Pi'\colon G' \rightarrow^* H'$ be rewrite sequences such that $G \in \mathcal{G}_n$. We write $\Pi \gg \Pi'$ if $(G;\varepsilon) \sqsupset^n_{\mathsf{mul}} (G';\varepsilon)$. Note that this implies that $G' \in \mathcal{G}_n$.*

Since $\rightarrow_n$ is closed under contexts, the relation $\succ^n$ is a well-founded order for every level-terminating CTRS $\mathcal{R}$ and every $n \geqslant 0$. As in the previous section, it follows that $\gg$ is a well-founded order on rewrite sequences. Note that this order depends only on the initial goals of rewrite sequences. In the proof below we make use of the well-known equivalence of $M \succ_{\mathsf{mul}} N$ and $M - N \succ_{\mathsf{mul}} N - M$.

[1] Cf. also [28, Theorem 6.8.9].

Theorem 5.3 *Let $\mathcal{R}$ be a terminating and level-confluent CTRS. For every solution θ of a goal G there exists an* LCNC*-refutation $G \Rightarrow_\sigma^* \Box$ such that $\sigma \leqslant_{\mathcal{R}} \theta\ [\mathcal{V}\mathrm{ar}(G)]$.*

Proof We use well-founded induction with respect the well-founded order $\gg$ on rewrite sequences. Let $\Pi\colon G\theta \rightarrow^* \top$ be any rewrite sequence from $G\theta$ to $\top$ and let κ be the level of Π. In order to make the induction work we prove $\sigma \leqslant_{\mathcal{R}} \theta\ [W]$ for a finite set of variables W that includes $\mathcal{V}\mathrm{ar}(G)$. The base case is trivial since G must be the empty goal (and thus we can take $\sigma = \varepsilon$). For the induction step we proceed as follows. First we transform Π into a level-minimal rewrite sequence. (This does not affect the complexity $(G;\theta)$.) By rearranging the order of the equations in Π, we can assume that the level of the leftmost equation in $G\theta$ is minimal. Write $G = e, H$ and $e = s \approx t$. Since it does not effect level-minimality, we may swap rewrite steps that take place in different sides of $e\theta$ at will. This will simplify the notation in some of the cases below. For the same reason we may assume that in Π always the leftmost equation different from `true` is rewritten. Let $\ell = \mathsf{level}(e\theta)$. So $\kappa \geqslant \mathsf{level}(e'\theta) \geqslant \ell$ for all $e' \in H$. We will show the existence of an LCNC-step $\Psi_1\colon G \Rightarrow_{\sigma_1} G_1$ and a rewrite sequence $\Pi_1\colon G_1\theta_1 \rightarrow^* \top$ such that $\sigma_1\theta_1 \leqslant_{\mathcal{R}} \theta\ [W]$ and $\Pi \gg \Pi_1$. We distinguish the following cases.

1. Suppose $s, t \notin \mathcal{V}$. We distinguish two further cases, depending on what happens to $e\theta$ in Π.

   (a) Suppose no reduct of $e\theta$ is rewritten at position 1 or 2. We may write $s = f(s_1, \ldots, s_n)$ and $t = f(t_1, \ldots, t_n)$. For $1 \leqslant i \leqslant n$ let e_i be the equation $s_i \approx t_i$. Let $G_1 = e_1, \ldots, e_n, H$. We have $\Psi_1\colon G \Rightarrow_{[\mathrm{d}]} G_1$. Let $\sigma_1 = \varepsilon$ and $\theta_1 = \theta$. The rewrite sequence Π, which must be of the form

   $$G\theta \ \rightarrow_\ell^* \ f(u_1, \ldots, u_n) \approx f(u_1, \ldots, u_n), H\theta \ \rightarrow \ \mathtt{true}, H\theta \ \rightarrow^* \ \top,$$

   can be transformed into Π_1:

   $$G_1\theta_1 \ \rightarrow^* \ u_1 \approx u_1, \ldots, u_n \approx u_n, H\theta \ \rightarrow^* \ \top, H\theta \ \rightarrow^* \ \top.$$

   Clearly $\sigma_1\theta_1 = \theta$. It is also clear that $G_1\theta_1 \in \mathcal{G}_\kappa$. It remains to show that $\Pi \gg \Pi_1$. We have $(G;\theta) - (G_1;\theta_1) = \{|e\theta|\}$ and $(G_1;\theta_1) - (G;\theta) = \{|e_1\theta|, \ldots, |e_n\theta|\}$. For all $1 \leqslant i \leqslant n$ we have $s\theta \rhd s_i\theta$ and $t\theta \rhd t_i\theta$ and thus also $s\theta \succ^\kappa s_i\theta$ and $t\theta \succ^\kappa t_i\theta$.

Hence $\{s\theta, t\theta\} \succ^{\kappa}_{\mathsf{mul}} \{s_i\theta, t_i\theta\}$ and therefore it suffices to show that $\mathsf{level}(e\theta) \geqslant \mathsf{level}(e_i\theta)$. We have $s_i\theta \to^*_\ell u_i$ and $t_i\theta \to^*_\ell u_i$. Therefore $\mathcal{R}_\ell \vdash e_i\theta$ and hence the level of $e_i\theta$ does not exceed the level of $e\theta$. We conclude that $\Pi \gg \Pi_1$.

(b) Suppose a reduct of $e\theta$ is rewritten at position 1 or 2. Without loss of generality we assume that the first such reduct is rewritten at position 1, using the fresh variant $l \to r \Leftarrow c$ of a rewrite rule in $\mathcal{R}$ with $l = f(l_1, \ldots, l_n)$. We have $s = f(s_1, \ldots, s_n)$. The rewrite sequence Π is of the form

$$G\theta \ \to^*_\ell \ f(s'_1, \ldots, s'_n) \approx t\theta, H\theta \ \to_\ell \ r\tau \approx t\theta, H\theta \ \to^* \ \top$$

with $f(s'_1, \ldots, s'_n) = l\tau$ and $c\tau \to^*_{\ell-1} \top$. For $1 \leqslant i \leqslant n$ let $e_i = s_i \approx l_i$ and define $G_1 = e_1, \ldots, e_n, r \approx t, c, H$. We have $\Psi_1 \colon G \Rightarrow_{[\mathsf{o}]} G_1$. Let $\sigma_1 = \varepsilon$ and define θ_1 as the disjoint union of θ and τ. Because

$$G_1\theta_1 = s_1\theta \approx l_1\tau, \ldots, s_n\theta \approx l_n\tau, r\tau \approx t\theta, c\tau, H\theta$$

we obtain the rewrite sequence Π_1:

$$\begin{aligned} G_1\theta_1 \ &\to^*_\ell \ s'_1 \approx l_1\tau, \ldots, s'_n \approx l_n\tau, r\tau \approx t\theta, c\tau, H\theta \\ &\to^* \ \top, r\tau \approx t\theta, c\tau, H \ \to^* \ \top. \end{aligned}$$

Since $W \cap \mathcal{D}(\tau) = \varnothing$ we have $\sigma_1\theta_1 = \theta \ [W]$. In order to conclude that $\Pi \gg \Pi_1$ we have to show that $(G;\theta) - (G_1;\theta_1) = \{|e\theta|\} \sqsupset^{\kappa}_{\mathsf{mul}} \{|e_1\theta_1|, \ldots, |e_n\theta_1|, |r\tau \approx t\theta|\} \uplus (c;\tau) = (G_1;\theta_1) - (G;\theta)$. For all $1 \leqslant i \leqslant n$ we have $s\theta \rhd s_i\theta \to^*_\ell s'_i = l_i\tau$ and consequently $\{s\theta, t\theta\} \succ^{\kappa}_{\mathsf{mul}} \{s_i\theta, l_i\tau\}$. Furthermore, the level of $e_i\theta_1$ does not exceed ℓ. Hence $|e\theta| \sqsupset^{\kappa} |e_1\theta_1|, \ldots, |e_n\theta_1|$. Next consider the equation $r\tau \approx t\theta$. Since $r\tau \approx t\theta \to^*_\ell \mathtt{true}$, the level of $r\tau \approx t\theta$ is at most ℓ. Also, $s\theta \to^*_\ell l\tau \to_\ell r\tau$ and thus $\{s\theta, t\theta\} \succ^{\kappa}_{\mathsf{mul}} \{r\tau, t\theta\}$. Hence $|e\theta| \sqsupset^{\kappa} |r\tau \approx t\theta|$. Finally, from $c\tau \to_{\ell-1} \top$ we infer that the level of every equation in $c\tau$ is less than ℓ. Therefore also $\{|e\theta|\} \sqsupset^{\kappa}_{\mathsf{mul}} (c;\tau)$.

2. Suppose $s \notin \mathcal{V}$ and $t \in \mathcal{V}$. We distinguish two further cases.

(a) Suppose no reduct of $e\theta$ is rewritten at position 1. Write $s = f(s_1, \ldots, s_n)$. Let $\sigma_1 = \{t \mapsto f(x_1, \ldots, x_n)\}$ and $e_i = x_i \approx s_i$

for $1 \leqslant i \leqslant n$. Here $x_1, \dots, x_n$ are fresh variables. Define $G_1 = (e_1, \dots, e_n, H)\sigma_1$. We have $\Psi_1 \colon G \Rightarrow_{[\mathsf{i}], \sigma_1} G_1$. The rewrite sequence Π is of the form

$$G\theta \rightarrow_{\ell}^{*} f(s_1', \dots, s_n') \approx f(s_1', \dots, s_n'), H\theta \rightarrow \mathtt{true}, H\theta \rightarrow^{*} \top.$$

Define θ_1 as the disjoint union of θ and $\{x_1 \mapsto s_1', \dots, x_n \mapsto s_n'\}$. We have

$$G_1\theta_1 = s_1' \approx s_1\sigma_1\theta_1, \dots, s_n' \approx s_n\sigma_1\theta_1, H\sigma_1\theta_1.$$

Because $t\theta \rightarrow_{\ell}^{*} f(s_1', \dots, s_n') = f(x_1, \dots, x_n)\theta_1 = t\sigma_1\theta_1$ and $x\theta = x\sigma_1\theta_1$ for variables x different from $x_1, \dots, x_n, t$, we have $x\theta \rightarrow_{\ell}^{*} x\sigma_1\theta_1$ for all variables $x \in W$. Hence $\sigma_1\theta_1 =_{\mathcal{R}} \theta\ [W]$, $H\theta \rightarrow_{\ell}^{*} H\sigma_1\theta_1$, and $s_i\sigma_1\theta_1 \ {}_{\ell}^{*}\!\leftarrow s_i\theta \rightarrow_{\ell}^{*} s_i'$ for all $1 \leqslant i \leqslant n$. Since the level of every equation in $H\theta$ is at least ℓ, level-confluence yields $H\sigma_1\theta_1 \rightarrow^{*} \top$ such that for every equation $e' \in H$ the level of $e'\sigma_1\theta_1$ is less than or equal to the level of $e'\theta$. In addition, we obtain terms $u_1, \dots, u_n$ such that $s_i\sigma_1\theta_1 \rightarrow_{\ell}^{*} u_i \ {}_{\ell}^{*}\!\leftarrow s_i'$ for all $1 \leqslant i \leqslant n$. Hence we obtain the rewrite sequence Π_1:

$$G_1\theta_1 \rightarrow_{\ell}^{*} u_1 \approx u_1, \dots, u_n \approx u_n, H\sigma_1\theta_1 \rightarrow^{*} \top.$$

We show that $\Pi \gg \Pi_1$. We have $(G; \theta) = \{|e\theta|\} \uplus (H; \theta)$ and $(G_1; \theta_1) = \{|e_1\theta_1|, \dots, |e_n\theta_1|\} \uplus (H\sigma_1; \theta_1)$. First we show that $|e\theta| \sqsupset^{\kappa} |e_1\theta_1|, \dots, |e_n\theta_1|$. For all $1 \leqslant i \leqslant n$ we have $s\theta \rhd s_i\theta \rightarrow_{\ell}^{*} s_i'$ and $s\theta \rhd s_i\theta \rightarrow_{\ell}^{*} s_i\sigma_1\theta_1$ and thus also $s\theta \succ^{\ell} s_i'$ and $s\theta \succ^{\ell} s_i\sigma_1\theta_1$. Since $\succ^{\ell} \subseteq \succ^{\kappa}$ we obtain $\{s\theta, t\theta\} \succ_{\mathsf{mul}}^{\kappa} \{s_i', s_i\sigma_1\theta_1\}$. From $s_i\sigma_1\theta_1 \rightarrow_{\ell}^{*} u_i \ {}_{\ell}^{*}\!\leftarrow s_i'$ we infer that the level of $e_i\theta_1$ does not exceed ℓ and therefore $|e\theta| \sqsupset^{\kappa} |e_1\theta_1|, \dots, |e_n\theta_1|$. To conclude that $\Pi \gg \Pi_1$ it now suffices to show that $(H; \theta) \sqsupseteq_{\mathsf{mul}}^{\kappa} (H\sigma_1; \theta_1)$. Here $\sqsupseteq_{\mathsf{mul}}^{\kappa}$ denotes the reflexive closure of $\sqsupset_{\mathsf{mul}}^{\kappa}$. Let $e' = u \approx v \in H$. We already observed that $\mathsf{level}(e'\theta) \geqslant \mathsf{level}(e'\sigma_1\theta_1)$. Furthermore, $e'\theta \rightarrow_{\ell}^{*} e'\sigma_1\theta_1$ and thus $\{u\theta, v\theta\} \succeq_{\mathsf{mul}}^{\kappa} \{u\sigma_1\theta_1, v\sigma_1\theta_1\}$. Hence $|e'\theta| \sqsupseteq^{\kappa} |e'\sigma_1\theta_1|$, implying the desired $(H; \theta) \sqsupseteq_{\mathsf{mul}}^{\kappa} (H\sigma_1; \theta_1)$.

(b) Suppose a reduct of $e\theta$ is rewritten at position 1. In this case we proceed as in case 1(b).

3. Suppose $s \in \mathcal{V}$ and $t \notin \mathcal{V}$. This case is similar to case 2.

4. Suppose $s, t \in \mathcal{V}$. We distinguish two further cases.

   (a) Suppose $s = t$. Let $G_1 = H$. We have $\Psi_1 \colon G \Rightarrow_{[\mathsf{t}]} G_1$. Let $\sigma_1 = \varepsilon$ and $\theta_1 = \theta$. From Π we extract the rewrite sequence $\Pi_1 \colon G_1\theta_1 \to^* \top$. We clearly have $\sigma_1\theta_1 = \theta$ and $\Pi \gg \Pi_1$ as $(G_1;\theta_1) \subset (G;\theta)$.

   (b) Suppose $s \neq t$. Let $\sigma_1 = \{s \mapsto t\}$ and $G_1 = H\sigma_1$. We have $\Psi_1 \colon G \Rightarrow_{[\mathsf{v}],\sigma_1} G_1$. Let θ_1 be the $\mathcal{R}_\ell$-normal form of θ, i.e., $\theta_1(x) = x\theta{\downarrow_\ell}$ for all variables $x \in \mathcal{V}$. Here $x\theta{\downarrow_\ell}$ denotes the unique normal form of $x\theta$ with respect to the confluent and terminating TRS $\mathcal{R}_\ell$. Since $e\theta \to_\ell^* \texttt{true}$, we have $s\theta{\downarrow_\ell} = t\theta{\downarrow_\ell}$. Hence $s\theta \to_\ell^* t\theta{\downarrow_\ell} = s\sigma_1\theta_1$. Because $x\theta \to_\ell^* x\theta{\downarrow_\ell} = x\sigma_1\theta_1$ for variables x different from s, we obtain $x\theta \to_\ell^* x\sigma_1\theta_1$ for all variables $x \in W$. Therefore $\sigma_1\theta_1 =_\mathcal{R} \theta\ [W]$. From Π we extract the rewrite sequence $H\theta \to^* \top$. Since $H\theta \to_\ell^* H\sigma_1\theta_1 = G_1\theta_1$, level-confluence yields a rewrite sequence $\Pi_1 \colon G_1\theta_1 \to^* \top$ such that for every equation $e' \in H$ the level of $e'\sigma_1\theta_1$ is less than or equal to the level of $e'\theta$. Hence we obtain $(H;\theta) \sqsupseteq^\kappa_{\mathsf{mul}} (G_1;\theta_1)$ as in case 2(a) and thus $(G;\theta) = \{|e\theta|\} \uplus (H;\theta) \sqsupset^\kappa_{\mathsf{mul}} (G_1;\theta_1)$. We conclude that $\Pi \gg \Pi_1$.

Let $W' = \mathcal{V}\mathrm{ar}_W(\sigma_1) \cup \mathcal{V}\mathrm{ar}(G_1)$. Clearly $\mathcal{V}\mathrm{ar}(G_1) \subseteq W'$. Hence we can apply the induction hypothesis to $\Pi_1 \colon G_1\theta_1 \to^* \top$. This yields an LCNC-refutation $\Psi' \colon G_1 \Rightarrow^*_{\sigma'} \square$ such that $\sigma' \leqslant_\mathcal{R} \theta_1\ [W']$. Define $\sigma = \sigma_1\sigma'$. From $\sigma_1\theta_1 \leqslant_\mathcal{R} \theta\ [W]$, $\sigma' \leqslant_\mathcal{R} \theta_1\ [W']$, and $\mathcal{V}\mathrm{ar}_W(\sigma_1) \subseteq W'$ we infer that $\sigma \leqslant_\mathcal{R} \theta\ [W]$. Concatenating the LCNC-step Ψ_1 and the LCNC-refutation Ψ' yields the desired LCNC-refutation Ψ. $\square$

Let us illustrate the above proof on a small example.

Example 5.4 *Consider the CTRS $\mathcal{R}$ consisting of the rewrite rules*

$$\begin{array}{rcll} \mathsf{a} & \to & \mathsf{b} & \Leftarrow\ \mathsf{f}(x) \approx \mathsf{g}(\mathsf{b}) \\ \mathsf{f}(\mathsf{b}) & \to & \mathsf{g}(x) & \Leftarrow\ \mathsf{f}(x) \approx \mathsf{g}(\mathsf{b}) \\ \mathsf{f}(\mathsf{a}) & \to & \mathsf{g}(\mathsf{b}) & \end{array}$$

of Werner [33, Beispiel 9.1]. This 3-CTRS is level-confluent and terminating. Consider the goal $G = \mathsf{f}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b})$. The empty substitution ε is a solution of G because we have the rewrite sequence

$$\Pi \colon G\varepsilon = \mathsf{f}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b}) \ \to_2\ \mathsf{g}(\mathsf{a}) \approx \mathsf{g}(\mathsf{b}) \ \to_2\ \mathsf{g}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b}) \ \to_0\ \texttt{true}.$$

In both the first and the second rewrite step the instantiated condition of the applied rewrite rule is $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{b})$ *which rewrites to* `true` *by applying the third rewrite rule. Below we show a possible* LCNC*-refutation* Ψ *constructed in the above proof. (Note that in general there are several possibilities for* Ψ *as the equation in* $G\theta$ *of minimal level is not uniquely determined.) The selected equations are underlined and the numbers on the right refer to the various cases in the proof:*

$$
\begin{array}{cr}
\underline{\mathsf{f}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b})} & \\
\Downarrow_{[\mathrm{o}]},\ \mathsf{f}(\mathsf{b}) \to \mathsf{g}(x_1) \Leftarrow \mathsf{f}(x_1) \approx \mathsf{g}(\mathsf{b}) & \textit{1(b)} \\
\underline{\mathsf{b} \approx \mathsf{b}},\ \mathsf{g}(x_1) \approx \mathsf{g}(\mathsf{b}),\ \mathsf{f}(x_1) \approx \mathsf{g}(\mathsf{b}) & \\
\Downarrow_{[\mathrm{d}]} & \textit{1(a)} \\
\mathsf{g}(x_1) \approx \mathsf{g}(\mathsf{b}),\ \underline{\mathsf{f}(x_1) \approx \mathsf{g}(\mathsf{b})} & \\
\Downarrow_{[\mathrm{o}]},\ \mathsf{f}(\mathsf{a}) \to \mathsf{g}(\mathsf{b}) & \textit{1(b)} \\
\mathsf{g}(x_1) \approx \mathsf{g}(\mathsf{b}),\ \underline{x_1 \approx \mathsf{a}},\ \mathsf{g}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b}) & \\
\Downarrow_{[\mathrm{i}]},\ \{x_1 \mapsto \mathsf{a}\} & \textit{3(a)} \\
\mathsf{g}(\mathsf{a}) \approx \mathsf{g}(\mathsf{b}),\ \underline{\mathsf{g}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b})} & \\
\Downarrow_{[\mathrm{d}]} & \textit{1(a)} \\
\mathsf{g}(\mathsf{a}) \approx \mathsf{g}(\mathsf{b}),\ \underline{\mathsf{b} \approx \mathsf{b}} & \\
\Downarrow_{[\mathrm{d}]} & \textit{1(a)} \\
\underline{\mathsf{g}(\mathsf{a}) \approx \mathsf{g}(\mathsf{b})} & \\
\Downarrow_{[\mathrm{d}]} & \textit{1(a)} \\
\underline{\mathsf{a} \approx \mathsf{b}} & \\
\Downarrow_{[\mathrm{o}]},\ \mathsf{a} \to \mathsf{b} \Leftarrow \mathsf{f}(x_2) \approx \mathsf{g}(\mathsf{b}) & \textit{1(b)} \\
\underline{\mathsf{b} \approx \mathsf{b}},\ \mathsf{f}(x_2) \approx \mathsf{g}(\mathsf{b}) & \\
\Downarrow_{[\mathrm{d}]} & \textit{1(a)} \\
\underline{\mathsf{f}(x_2) \approx \mathsf{g}(\mathsf{b})} & \\
\Downarrow_{[\mathrm{o}]},\ \mathsf{f}(\mathsf{a}) \to \mathsf{g}(\mathsf{b}) & \textit{1(b)} \\
\underline{x_2 \approx \mathsf{a}},\ \mathsf{g}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b}) & \\
\Downarrow_{[\mathrm{i}]},\ \{x_2 \mapsto \mathsf{a}\} & \textit{3(a)}
\end{array}
$$

$$\begin{array}{cl} \underline{\mathsf{g}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b})} & \\ \Downarrow_{[\mathrm{d}]} & \quad\quad\quad\quad \mathit{1(a)} \\ \underline{\mathsf{b} \approx \mathsf{b}} & \\ \Downarrow_{[\mathrm{d}]} & \quad\quad\quad\quad \mathit{1(a)} \\ \square & \end{array}$$

It is interesting to note that basic conditional narrowing fails to solve the goal $\mathsf{f}(\mathsf{b}) \approx \mathsf{g}(\mathsf{b})$ (Werner [33]). Hence the completeness of LCNC for the above CTRS does not follow from the strong completeness result of the previous section. An interesting question for future research is to establish or refute the strong completeness of LCNC for the class of terminating and level-confluent CTRSs and, if strong completeness fails to hold, to identify complete selection functions.

6 Conclusion

In this paper we presented a number of completeness results for the lazy conditional narrowing calculus LCNC. The only result that does not rely on some kind of termination condition is the one of Section 3, for confluent 1-CTRSs with respect to normalized solutions and leftmost selection. However, unlike the results of Sections 4 and 5, this result does not permit extra variables in the conditions and right-hand sides of the rewrite rules. An important question is to find a class of non-terminating CTRSs with extra variables for which (strong) completeness of LCNC (as well as ordinary conditional narrowing) can be established. We believe that the class proposed by Suzuki *et al.* [32] is a promising candidate.

Even when completeness with respect to a specific selection function like $\mathcal{S}_{\mathrm{left}}$ is known, the search space of LCNC is still very large, owing mainly to the delayed matching of patterns in the application of the outermost narrowing rule as well as the non-determinism due to the choice of the inference rule. In future work we would like to investigate under what conditions we can eliminate this non-determinism. In the unconditional case this question has been fully answered ([25]), but it seems doubtful whether the same conditions work for arbitrary confluent (1-)CTRSs. In [13] Ida and Hamada present an implementation of LCNC_{d} in the symbolic computation language Mathematica. LCNC_{d} is the conditional counterpart of the deterministic calculus LNC_{d}

([25]) and incorporates leftmost selection. It is unknown for which classes of CTRSs and solutions this calculus is complete. (No completeness results are reported in [13].)

Acknowledgements

Taro Suzuki is partially supported by the Grant-in-Aid for Encouragement of Young Scientist 11780204. Aart Middeldorp is partially supported by the Grant-in-Aid for Scientific Research (B) 12480066 of the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

[1] S. Antoy, R. Echahed, and M. Hanus. A needed narrowing strategy. *Journal of the ACM*, 47(4):776–822, 2000.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[3] A. Bockmayr. *Beiträge zur Theorie des Logisch-Funktionalen Programmierens.* PhD thesis, Universität Karlsruhe, 1990. In German.

[4] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–320. North-Holland, 1990.

[5] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22:465–476, 1979.

[6] N. Dershowitz, M. Okada, and G. Sivakumar. Canonical conditional rewrite systems. In *Proceedings of the 9th International Conference on Automated Deduction*, volume 310 of *LNCS*, pages 538–549, 1988.

[7] M. Fay. First-order unification in equational theories. In *Proceedings of the 4th Conference on Automated Deduction*, pages 161–167, 1979.

[8] J. Gallier and W. Snyder. Complete sets of transformations for general E-unification. *Theoretical Computer Science*, 67:203–260, 1989.

[9] E. Giovannetti and C. Moiso. A completeness result for E-unification algorithms based on conditional narrowing. In *Proceedings of the Workshop on Foundations of Logic and Functional Programming*, volume 306 of *LNCS*, pages 157–167, 1986.

[10] J.C. González-Moreno, M.T. Hortalá-González, and M. Rodríguez-Artalejo. A higer-order rewriting logic for functional logic programming. In *Proceedings of the International Conference on Logic Programming*, pages 153–167. The MIT Press, 1997.

[11] J.C. González-Moreno, M.T. Hortalá-González, and M. Rodríguez-Artalejo. Polymorphic types in functional logic programming. *Journal of Functional and Logic Programming*, 2001(1), 2001.

[12] M. Hamada. Strong completeness of a narrowing calculus for conditional rewrite systems with extra variables. In *Proceedings of the 6th Australasian Theory Symposium, Electronic Notes in Theoretical Computer Science*, volume 31. Elsevier Science Publishers, 2000.

[13] M. Hamada and T. Ida. Deterministic and non-deterministic lazy conditional narrowing and their implementations. *Transactions of Information Processing Society of Japan*, 39(3):656–663, 1998.

[14] M. Hamada and A. Middeldorp. Strong completeness of a lazy conditional narrowing calculus. In *Proceedings of the 2nd Fuji International Workshop on Functional and Logic Programming*, pages 14–32, Shonan Village, 1997. World Scientific.

[15] M. Hamada, A. Middeldorp, and T. Suzuki. Completeness results for a lazy conditional narrowing calculus. In *Combinatorics, Computation and Logic: Proceedings of 2nd Discrete Mathematics and Theoretical Computer Science Conference and the 5th Australasian Theory Symposium*, pages 217–231, Auckland, 1999. Springer-Verlag Singapore.

[16] M. Hanus. The integration of functions into logic programming: From theory to practice. *Journal of Logic Programming*, 19 & 20:583–628, 1994.

[17] M. Hanus. Lazy unification with simplification. In *Proceedings of the 5th European Symposium on Programming*, volume 788 of *LNCS*, pages 272–286, 1994.

[18] S. Hölldobler. *Foundations of Equational Logic Programming*, volume 353 of *LNAI.* 1989.

[19] J.-M. Hullot. Canonical forms and unification. In *Proceedings of the 5th Conference on Automated Deduction*, volume 87 of *LNCS*, pages 318–334, 1980.

[20] S. Kaplan. Conditional rewrite rules. *Theoretical Computer Science*, 33:175–193, 1984.

[21] J.W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 1–116. Oxford University Press, 1992.

[22] M. Marin, T. Ida, and T. Suzuki. On reducing the search space of higer-order lazy narrowing. In *Proceedings of the 4th Fuji International Symposium on Functional and Logic Programming*, volume 1722 of *LNCS*, pages 319–334, 1999.

[23] A. Martelli, C. Moiso, and G.F. Rossi. Lazy unification algorithms for canonical rewrite systems. In H. Aït-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures, Vol. II, Rewriting Techniques*, pages 245–274. Academic Press Press, 1989.

[24] A. Middeldorp and E. Hamoen. Completeness results for basic narrowing. *Applicable Algebra in Engineering, Communication and Computing*, 5:213–253, 1994.

[25] A. Middeldorp and S. Okui. A deterministic lazy narrowing calculus. *Journal of Symbolic Computation*, 25(6):733–757, 1998.

[26] A. Middeldorp, S. Okui, and T. Ida. Lazy narrowing: Strong completeness and eager variable elimination. *Theoretical Computer Science*, 167(1,2):95–130, 1996.

[27] C. Prehofer. A call-by-need strategy for higher-order functional-logic programming. In *Proceedings of the International Symposium on Logic Programming*, pages 147–161. MIT Press, 1995.

[28] C. Prehofer. *Solving Higher-Order Equations: From Logic to Programming.* PhD thesis, Technische Universität München, 1995. Appeared as Technical Report 19508.

[29] C. Prehofer. *Solving Higher-Order Equations: From Logic to Programming.* Progress in Theoretical Computer Science. Birkäuser, 1998.

[30] W. Snyder. *A Proof Theory for General Unification*, volume 11 of *Progress in Computer Science and Applied Logic.* Birkäuser, 1991.

[31] T. Suzuki and A. Middeldorp. A complete selection function for lazy conditional narrowing. In *Proceedings of the 5th International Symposium on Functional and Logic Programming*, volume 2024 of *Lecture Notes in Computer Science*, pages 201–215, Tokyo, 2001. Springer-Verlag.

[32] T. Suzuki, A. Middeldorp, and T. Ida. Level-confluence of conditional rewrite systems with extra variables in right-hand sides. In *Proceedings of the 6th International Conference on Rewriting Techniques and Applications*, volume 914 of *Lecture Notes in Computer Science*, pages 179–193, Kaiserslautern, 1995. Springer-Verlag.

[33] A. Werner. *Untersuchung von Strategien für das logisch-funktionale Programmieren.* Shaker-Verlag Aachen, March 1998. In German.

A Appendix

In this appendix we prove Lemmata 4.4 and 4.13. Most of the work for the former is done in the following preliminary lemma.

Lemma A.1 *Let* $\Pi\colon G \rightsquigarrow^*_\theta \top$ *be a* CNC*-refutation,* W *a set of variables, and* γ *a substitution such that* $\mathcal{V}\mathrm{ar}(G) \subseteq W$, $\gamma \leqslant \theta$ $[W]$*, and the variables in* $\mathcal{D}(\gamma) \cup \mathcal{I}(\gamma)$ *are different from the variables in the employed rewrite rules. There exists a* CNC*-refutation* $\Pi'\colon G\gamma \rightsquigarrow^*_{\theta'} \top$ *such that* Π *subsumes* Π', $\Pi\theta = \Pi'\theta'$*, and* $\gamma\theta' = \theta$ $[W]$.

Proof We use induction on the length n of Π. The case $n = 0$ is obvious. Suppose $n > 0$. Let the first step of Π be

$$G = (G', e, G'') \rightsquigarrow_{\sigma_1, p_1, l_1 \to r_1 \Leftarrow c_1, e} (G', e[r_1]_{p_1}, c_1, G'')\sigma_1 = G_1$$

and let $\Pi_1\colon G_1 \rightsquigarrow^*_{\theta_1} \top$ be the remainder of Π, so $\sigma_1\theta_1 = \theta$. Let X be the set of variables in the rewrite rules employed in Π. Without loss of generality we assume that $\mathcal{V}\mathrm{ar}(G) \cap X = \varnothing$ and $X' \cap (\mathcal{D}(\sigma_1) \cup \mathcal{I}(\sigma_1)) = \varnothing$. Here X'

is defined as the set $X \setminus \mathcal{V}\text{ar}(l_1 \to r_1 \Leftarrow c_1)$. These two assumptions simply state that the variables in the rewrite rules are sufficiently fresh.

First we show that we can mimic the first step of Π starting from the goal $G\gamma$. From $\gamma \leqslant \theta\ [W]$ we obtain a substitution δ such that $\gamma\delta = \theta\ [W]$. Without loss of generality we may assume that $\mathcal{D}(\delta) \subseteq \mathcal{V}\text{ar}_W(\gamma)$. Hence the substitution δ' defined by

$$\delta'(x) = \begin{cases} x\delta & \text{if } x \in \mathcal{V}\text{ar}_W(\gamma), \\ x\theta & \text{otherwise} \end{cases}$$

satisfies $\gamma\delta' = \theta\ [W \cup X]$ because $(\mathcal{D}(\gamma) \cup \mathcal{I}(\gamma)) \cap X = \varnothing$. Since $\mathcal{V}\text{ar}(e) \subseteq \mathcal{V}\text{ar}(G) \subseteq W$ and $\mathcal{D}(\gamma) \cap X = \varnothing$ we have $e\gamma_{|p_1}\delta' = e_{|p_1}\gamma\delta' = e_{|p_1}\theta = e_{|p_1}\sigma_1\theta_1 = l_1\sigma_1\theta_1 = l_1\theta = l_1\gamma\delta' = l_1\delta'$, so $e\gamma_{|p_1}$ and l_1 are unifiable. Hence we obtain the CNC-step

$$G\gamma \ \leadsto_{p_1,\, \sigma_1',\, l_1 \to r_1 \Leftarrow c_1,\, e\gamma} \ (G'\gamma, e\gamma[r_1]_{p_1}, c_1, G''\gamma)\sigma_1'$$

where σ_1' is any idempotent most general unifier of $e\gamma_{|p_1}$ and l_1.

Next we show that $(G'\gamma, e\gamma[r_1]_{p_1}, c_1, G''\gamma)\sigma_1' = G_1\gamma_1$ for a substitution γ_1 satisfying $\gamma_1 \leqslant \theta_1\ [W']$ for a set of variables W' that includes $\mathcal{V}\text{ar}(G_1)$. We have $e_{|p_1}\gamma\sigma_1' = e\gamma_{|p_1}\sigma_1' = l_1\sigma_1' = l_1\gamma\sigma_1'$, so $\gamma\sigma_1'$ is a unifier of $e_{|p_1}$ and l_1. Since σ_1 is a most general unifier of $e_{|p_1}$ and l_1 there exists a substitution γ' such that $\sigma_1\gamma' = \gamma\sigma_1'$. Let $W' = \mathcal{V}\text{ar}_{W \cup X}(\sigma_1)$ and $\gamma_1 = \gamma' \restriction_{W'}$. It is easy to show that $\mathcal{V}\text{ar}(G_1) \subseteq W'$. We have $\sigma_1\gamma_1 = \gamma\sigma_1'\ [W \cup X]$ and because $\mathcal{D}(\gamma) \cap X = \varnothing$ we obtain $(G'\gamma, e\gamma[r_1]_{p_1}, c_1, G''\gamma)\sigma_1' = (G'\gamma, e\gamma[r_1\gamma]_{p_1}, c_1\gamma, G''\gamma)\sigma_1' = G_1\gamma_1$. Because δ' is a unifier of $e\gamma_{|p_1}$ and l_1, there exists a substitution δ'' such that $\sigma_1'\delta'' = \delta'$. Using $\sigma_1\gamma_1 = \gamma\sigma_1'\ [W \cup X]$ we obtain $\sigma_1\gamma_1\delta'' = \gamma\sigma_1'\delta'' = \gamma\delta' = \sigma_1\theta_1\ [W \cup X]$ and thus $\gamma_1 \leqslant \theta_1\ [W']$.

We still have to show that $(\mathcal{D}(\gamma_1) \cup \mathcal{I}(\gamma_1)) \cap X' = \varnothing$ before we can apply the induction hypothesis to Π_1. Because σ_1' is an idempotent most general unifier of $e\gamma_{|p_1}$ and l_1, we have $\mathcal{D}(\sigma_1') \cup \mathcal{I}(\sigma_1') = \mathcal{V}\text{ar}(e\gamma_{|p_1}) \cup \mathcal{V}\text{ar}(l_1)$. Clearly $\mathcal{V}\text{ar}(e\gamma_{|p_1}) \subseteq \mathcal{V}\text{ar}(G) \cup \mathcal{I}(\gamma)$ and thus $\mathcal{V}\text{ar}(e\gamma_{|p_1}) \cap X = \varnothing$ by assumption. Also $\mathcal{V}\text{ar}(l_1) \subseteq X \setminus X'$. Hence $(\mathcal{D}(\sigma_1') \cup \mathcal{I}(\sigma_1')) \cap X' = \varnothing$. Together with $\mathcal{D}(\sigma_1) \cap X' = \varnothing$, $\mathcal{D}(\gamma) \cap X = \varnothing$, and $\sigma_1\gamma_1 = \gamma\sigma_1'\ [W \cup X]$ this implies $\gamma_1 = \varepsilon\ [X']$, i.e., $\mathcal{D}(\gamma_1) \cap X' = \varnothing$. It remains to show that $\mathcal{I}(\gamma_1) \cap X' = \varnothing$. First observe that from $\mathcal{I}(\gamma\sigma_1') \subseteq \mathcal{I}(\gamma) \cup \mathcal{I}(\sigma_1')$, $\mathcal{I}(\gamma) \cap X = \varnothing$, and $\mathcal{I}(\sigma_1') \cap X' = \varnothing$ it follows that $\mathcal{I}(\gamma\sigma_1') \cap X' = \varnothing$. Suppose to the contrary that $x \in \mathcal{I}(\gamma_1) \cap X'$ for some variable x. So there exists a variable $y \in \mathcal{D}(\gamma_1)$ such that $x \in \mathcal{V}\text{ar}(y\gamma_1)$. We have $\mathcal{D}(\gamma_1) \subseteq ((W \cup X) \setminus \mathcal{D}(\sigma_1)) \cup \mathcal{I}(\sigma_1 \restriction_{W \cup X})$. If $y \in (W \cup X) \setminus \mathcal{D}(\sigma_1)$ then $y\gamma_1 = y\sigma_1\gamma_1 = y\gamma\sigma_1'$, so $x \in \mathcal{I}(\gamma\sigma_1')$. This contradicts $\mathcal{I}(\gamma\sigma_1') \cap X' = \varnothing$. If $y \in \mathcal{I}(\sigma_1\restriction_{W \cup X})$ then there exists a variable $z \in W \cup X$ with $y \in \mathcal{V}\mathrm{ar}(z\sigma_1)$ and thus $x \in \mathcal{V}\mathrm{ar}(z\sigma_1\gamma_1) = \mathcal{V}\mathrm{ar}(z\gamma\sigma_1')$, again contradicting $\mathcal{I}(\gamma\sigma_1') \cap X' = \varnothing$.

Now we are in a position to apply the induction hypothesis to Π_1. This yields a CNC-refutation $G_1\gamma_1 \rightsquigarrow^*_{\theta_1'} \top$ such that $\gamma_1\theta_1' = \theta_1\ [W']$. Concatenating this CNC-refutation with the CNC-step $G\gamma \rightsquigarrow_{\sigma_1'} G_1\gamma_1$ yields the CNC-refutation Π': $G\gamma \rightsquigarrow^*_{\theta'} \top$. Here $\theta' = \sigma_1'\theta_1'$. From $\gamma_1\theta_1' = \theta_1\ [W']$ we infer $\sigma_1\gamma_1\theta_1' = \sigma_1\theta_1 = \theta\ [W \cup X]$ and thus $\gamma\theta' = \gamma\sigma_1'\theta_1' = \sigma_1\gamma_1\theta_1' = \theta\ [W \cup X]$. Hence also $\gamma\theta' = \theta\ [W]$. From the construction of Π' it follows that Π subsumes Π' and $\Pi\theta = \Pi'\theta'$. $\square$

Proof of Lemma 4.4. Let $t\theta = f(t_1, \dots, t_n)$ and define the substitution σ' as follows:

$$\sigma'(x) = \begin{cases} t_i & \text{if } x = x_i \text{ for some } 1 \leqslant i \leqslant n, \\ x\theta & \text{otherwise.} \end{cases}$$

We clearly have $\sigma_1\sigma' = \theta\ [W]$ and thus $\sigma_1 \leqslant \theta\ [W]$. An application of Lemma A.1 yields the desired result. $\square$

Proof of Lemma 4.13. Let $G_1 = G', e_1, G'', e_2, G'''$. Since we may assume that the variables in $l_2 \to r_2 \Leftarrow c_2$ are fresh, we have $\mathcal{D}(\tau_1) \cap \mathcal{V}\mathrm{ar}(l_2 \to r_2 \Leftarrow c_2) = \varnothing$. Hence $r_2\tau_1 = r_2$, $c_2\tau_1 = c_2$, and thus

$$\begin{aligned} G_3 &= ((G', e_1[r_1]_{p_1}, c_1, G'')\tau_1, e_2\tau_1[r_2]_{p_2}, c_2, G'''\tau_1)\tau_2 \\ &= (G', e_1[r_1]_{p_1}, c_1, G'', e_2[r_2]_{p_2}, c_2, G''')\tau_1\tau_2. \end{aligned}$$

From $\mathcal{D}(\tau_1) \cap \mathcal{V}\mathrm{ar}(l_2) = \varnothing$ we also infer that $e_{2|p_2}\tau_1\tau_2 = e_2\tau_{1|p_2}\tau_2 = l_2\tau_2 = l_2\tau_1\tau_2$, so $e_{2|p_2}$ and l_2 are unifiable. Let υ_2 be an idempotent most general unifier of these two terms. By definition of most general unifier there exists a substitution ρ such that $\upsilon_2\rho = \tau_1\tau_2$. We have $H_2 = (G', e_1, G'', e_2[r_2]_{p_2}, c_2, G''')\upsilon_2$ and $\mathcal{D}(\upsilon_2) \subseteq \mathcal{V}\mathrm{ar}(e_{2|p_2}) \cup \mathcal{V}\mathrm{ar}(l_2)$ by idempotency of υ_2. Because we may assume that $\mathcal{V}\mathrm{ar}(l_1 \to r_1 \Leftarrow c_1) \cap (\mathcal{V}\mathrm{ar}(e_2) \cup \mathcal{V}\mathrm{ar}(l_2 \to r_2 \Leftarrow c_2)) = \varnothing$, we obtain $\mathcal{D}(\upsilon_2) \cap \mathcal{V}\mathrm{ar}(l_1 \to r_1 \Leftarrow c_1) = \varnothing$. Hence $e_1\upsilon_{2|p_1}\rho = e_{1|p_1}\upsilon_2\rho = e_{1|p_1}\tau_1\tau_2 = l_1\tau_1\tau_2 = l_1\upsilon_2\rho = l_1\rho$. So the terms $e_1\upsilon_{2|p_1}$ and l_1 are unifiable. Let σ be a most general unifier. We have $\sigma \leqslant \rho$. It follows that $\upsilon_2\sigma \leqslant \tau_1\tau_2$. Using $\mathcal{D}(\upsilon_2) \cap \mathcal{V}\mathrm{ar}(l_1) = \varnothing$ we obtain $e_{1|p_1}\upsilon_2\sigma = e_1\upsilon_{2|p_1}\sigma = l_1\sigma = l_1\upsilon_2\sigma$, so $\upsilon_2\sigma$ is a unifier of $e_{1|p_1}$ and l_1. Because τ_1 is a most general unifier of these two terms, we must have $\tau_1 \leqslant \upsilon_2\sigma$. Let γ be any substitution satisfying $\tau_1\gamma = \upsilon_2\sigma$. With help of $\mathcal{D}(\tau_1) \cap \mathcal{V}\mathrm{ar}(l_2) = \varnothing$ we obtain $e_2\tau_{1|p_2}\gamma = e_{2|p_2}\tau_1\gamma = e_{2|p_2}\upsilon_2\sigma = l_2\upsilon_2\sigma = l_2\tau_1\gamma = l_2\gamma$. Hence we obtain $\tau_2 \leqslant \gamma$ from the fact that τ_2 is a most general unifier of $e_2\tau_{1|p_2}$ and l_2. Therefore $\tau_1\tau_2 \leqslant \tau_1\gamma = \upsilon_2\sigma$. Since we also have $\upsilon_2\sigma \leqslant \tau_1\tau_2$, there is a variable renaming δ such that $\upsilon_2\sigma\delta = \tau_1\tau_2$. Now define $\upsilon_1 = \sigma\delta$. Since most general unifiers are closed under variable renaming, υ_1 is a most general unifier of $e_1\upsilon_{2|p_1}$ and l_1. From $\mathcal{D}(\upsilon_2) \cap \mathcal{V}\mathrm{ar}(l_1 \to r_1 \Leftarrow c_1) = \varnothing$ we infer $r_1\upsilon_2 = r_1$, $c_1\upsilon_2 = c_1$, and thus

$$\begin{aligned} H_3 &= (G'\upsilon_2, e_1\upsilon_2[r_1]_{p_1}, c_1, (G'', e_2[r_2]_{p_2}, c_2, G''')\upsilon_2)\upsilon_1 \\ &= (G', e_1[r_1]_{p_1}, c_1, G'', e_2[r_2]_{p_2}, c_2, G''')\upsilon_2\upsilon_1. \end{aligned}$$

Since $\tau_1\tau_2 = \upsilon_2\upsilon_1$ we conclude that $G_3 = H_3$. $\square$