The 2D Dependency Pair Framework for conditional rewrite systems. Part I: definition and basic processors[☆]

Salvador Lucas[a], José Meseguer[b], Raúl Gutiérrez[a]

[a] *DSIC, Universitat Politècnica de València*
[b] *CS Dept. at the University of Illinois at Urbana-Champaign*

Abstract

Different termination properties of *conditional term rewriting systems* have been recently described emphasizing the *bidimensional* nature of the termination behavior of conditional rewriting. The absence of infinite sequences of rewriting steps (*termination* in the usual sense), provides the *horizontal* dimension. The absence of infinitely many attempts to launch the *subsidiary processes* that are required to check the rule's condition and perform a *single* rewriting step has been called *V-termination* and provides the *vertical* dimension. We have characterized these properties by means of appropriate notions of *dependency pairs* and *dependency chains*. In this paper we introduce a *2D Dependency Pair Framework* for automatically proving and disproving all these termination properties. Our implementation of the framework as part of the termination tool MU-TERM and the benchmarks obtained so far suggest that the 2D Dependency Pair Framework is currently the most powerful technique for proving operational termination of conditional term rewriting systems.

Keywords: Conditional term rewriting, dependency pairs, program analysis, operational termination

1. Introduction

Computations with *Conditional Term Rewriting Systems* (CTRSs [3, 7, 16]) can be defined by means of an *Inference System* where each rewriting step $s \to_{\mathcal{R}} t$ requires a *proof* of the goal $s \to t$ before it can be considered part of the one-step rewriting relation associated to $\mathcal{R}$ (see Figure 1). Similarly, the fact that a term s rewrites (in zero or more rewriting steps) to another term t using $\mathcal{R}$ (denoted $s \to^*_{\mathcal{R}} t$) is witnessed by the existence of a *proof* of the goal $s \to^* t$.

Remark 1. *All rules in the inference system in Figure 1 are* schematic *in that each inference rule* $\frac{B_1 \ \cdots \ B_n}{A}$ *can be used under any* instance $\frac{\sigma(B_1) \ \cdots \ \sigma(B_n)}{\sigma(A)}$ *of*

[☆] Partially supported by the EU (FEDER), Spanish MINECO project TIN2015-69175-C4-1-R, GV project PROMETEOII/2015/013, and NSF grant CNS 13-19109. Raúl Gutiérrez is also supported by Juan de la Cierva Fellowship JCI-2012-13528.

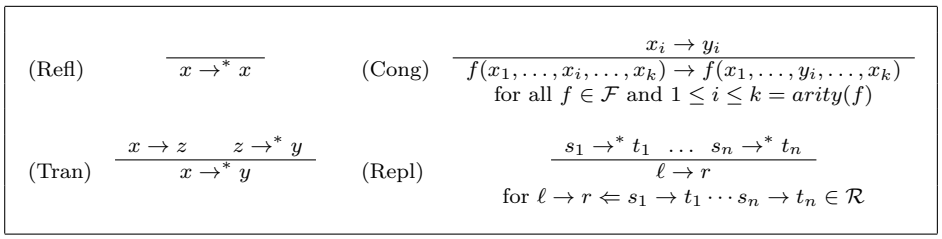


Figure 1: Inference rules for conditional rewriting with a CTRS $\mathcal{R}$ with signature $\mathcal{F}$

the rule by a substitution σ [29]. For instance, (Repl) actually establishes that, for every rule $\ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n$ in the CTRS $\mathcal{R}$, every instance $\sigma(\ell)$ by a substitution σ rewrites into $\sigma(r)$ provided that, for each $s_i \to t_i$, with $1 \le i \le n$, the reachability condition $\sigma(s_i) \to^* \sigma(t_i)$ can be proved.

Following this *proof theoretic* approach to the operational semantics of CTRSs, the definition of their *termination properties* concerns the absence or existence of infinite proof trees. The notion of *operational termination* captures this idea, meaning that, given an initial goal, an interpreter will either succeed in finite time in producing a proof tree, or will fail in finite time, not being able to close or extend further any of the possible proof trees, after exhaustively searching for all such proof trees [22].

Advanced declarative programming languages like ASF+SDF [5], CafeOBJ [8], ELAN [4], Haskell [15], OBJ [13], or Maude [6] use CTRSs as an appropriate basis for modeling computations with programs written in such languages. For this reason, in [24] we have investigated characterizations of termination properties of CTRSs to faithfully capture the *bidimensional nature* of infinite computations with CTRSs [24, Section 3]: besides infinite sequences of rewriting steps (a *horizontal* dimension, corresponding to the usual notion of *nonterminating* rewriting computation), there can be infinitely many attempts to satisfy a rule's condition when a *single* rewriting step is attempted (a *vertical* dimension). The second (vertical) dimension is due to the use of *conditional rules* $\ell \to r \Leftarrow c$ where the *conditional part* c consists of sequences of conditions $s \to t$ which are treated as reachability tests $\sigma(s) \to^* \sigma(t)$ for the *matching substitution* σ which is used to (try to) apply a single rewriting step. Such tests may start subsidiary computations that may run forever. In [24, Section 3] we have proved that operational termination of CTRSs is equivalent to the conjunction of two properties: termination and V-termination. Here, *termination* of a CTRS $\mathcal{R}$ is (as usual) the absence of infinite sequences $t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \cdots$ of *rewriting steps*; and *V-termination* is a new property which captures the absence of infinitely many attempts to launch the subsidiary processes that are required to perform a single rewriting step (the precise definition is given below).

These termination properties of a CTRS $\mathcal{R}$ are captured using *two* CTRSs $\mathsf{DP}_H(\mathcal{R})$ (for termination) and $\mathsf{DP}_V(\mathcal{R})$ (for V-termination) which are obtained from $\mathcal{R}$ [24, Sections 4 and 5]. As in the corresponding approach for Term Rewriting Systems (TRSs) [1], we call them *Dependency Pairs*. We collectively

call $\mathsf{DP}_H(\mathcal{R})$ and $\mathsf{DP}_V(\mathcal{R})$ the *2D Dependency Pairs* (2D DPs) of $\mathcal{R}$.

1. The CTRS $\mathsf{DP}_H(\mathcal{R})$ of *horizontal* dependency pairs contains, for each rule $\ell \to r \Leftarrow c$ in $\mathcal{R}$, all rules $\ell^\sharp \to v^\sharp \Leftarrow c$ that are obtained for each *defined* subterm v of r, i.e., v is a subterm of r and the root symbol of v is a *defined* symbol.[1] The notation $t^\sharp$ for a term t represents the *marking* of the *root symbol* f of t. We often capitalize f into F instead of writing $f^\sharp$.
2. The CTRS $\mathsf{DP}_V(\mathcal{R})$ of *vertical* dependency pairs contains rules $\ell^\sharp \to v^\sharp \Leftarrow s_1 \to t_1, \ldots, s_{i-1} \to t_{i-1}$ for each rule $\ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n$ and i, $1 \leq i \leq n$ such that v is a defined subterm of s_i.

Example 2. *The following CTRS $\mathcal{R}$ in [28, Example 7.1.5] implements the computation of the quotient and remainder of two natural numbers represented in Peano's notation where $\underbrace{\mathsf{s}(\cdots\mathsf{s}(}_{n}\,\mathsf{0}\,\underbrace{)\cdots)}_{n}$ represents the natural number n:*

$$\begin{aligned}
\mathsf{less}(x,\mathsf{0}) &\to \mathsf{false} && (1)\\
\mathsf{less}(\mathsf{0},\mathsf{s}(x)) &\to \mathsf{true} && (2)\\
\mathsf{less}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{less}(x,y) && (3)\\
\mathsf{monus}(\mathsf{0},\mathsf{s}(y)) &\to \mathsf{0} && (4)\\
\mathsf{monus}(x,\mathsf{0}) &\to x && (5)\\
\mathsf{monus}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{monus}(x,y) && (6)\\
\mathsf{quotrem}(\mathsf{0},\mathsf{s}(y)) &\to \mathsf{pair}(\mathsf{0},\mathsf{0}) && (7)\\
\mathsf{quotrem}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{pair}(\mathsf{0},\mathsf{s}(x)) \Leftarrow \mathsf{less}(x,y) \to \mathsf{true} && (8)\\
\mathsf{quotrem}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{pair}(\mathsf{s}(q),r) && (9)\\
&\Leftarrow \mathsf{less}(x,y) \to \mathsf{false}, \mathsf{quotrem}(\mathsf{monus}(x,y),\mathsf{s}(y)) \to \mathsf{pair}(q,r)
\end{aligned}$$

The CTRS $\mathsf{DP}_H(\mathcal{R})$ consists of the rules:

$$\begin{aligned}
\mathsf{LESS}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{LESS}(x,y) && (10)\\
\mathsf{MONUS}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{MONUS}(x,y) && (11)
\end{aligned}$$

The CTRS $\mathsf{DP}_V(\mathcal{R})$ consists of the rules:

$$\begin{aligned}
\mathsf{QUOTREM}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{LESS}(x,y) && (12)\\
\mathsf{QUOTREM}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{QUOTREM}(\mathsf{monus}(x,y),\mathsf{s}(y)) && (13)\\
&\Leftarrow \mathsf{less}(x,y) \to \mathsf{false}\\
\mathsf{QUOTREM}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{MONUS}(x,y) \Leftarrow \mathsf{less}(x,y) \to \mathsf{false} && (14)
\end{aligned}$$

Termination, V-termination, and operational termination of CTRSs $\mathcal{R}$ can be investigated as the absence of infinite *chains* of dependency pairs from $\mathsf{DP}_H(\mathcal{R})$ and $\mathsf{DP}_V(\mathcal{R})$ [24, Section 7]. However, the theory in [24] does *not* provide

[1] A k-ary symbol f is *defined* in a CTRS $\mathcal{R}$ if there is a rule $f(\ell_1,\ldots,\ell_k) \to r \Leftarrow c$ in $\mathcal{R}$.

any practical methodology to tackle this task. In this paper we close this gap by extending to CTRSs the *Dependency Pair Framework* for Term Rewriting Systems (TRSs) [10, 12], which is very useful for the development of tools for automatically proving termination. We define a *2D Dependency Pair Framework* to mechanically prove or disprove termination, V-termination, and operational termination of CTRSs using 2D DPs.

1.1. Contributions of the paper

The contributions of this paper are developed after the preliminary Sections 2 and 3 (the last one contains the material from [24] which is used in this paper to make our presentation self-contained). In Section 4, we pay some attention to the role of *infeasible rules*, i.e., rules whose conditional part cannot be satisfied by any substitution and therefore cannot be used to perform any rewriting step. In Section 5 we define our *2D DP Framework* to mechanically prove or disprove termination, V-termination, and operational termination of CTRSs using 2D DPs. For instance, the framework can be used to prove *termination* of CTRSs which are *not* operationally terminating and also to prove V-termination of CTRSs which are *not* terminating, and is also able to *disprove* operational termination of both of them.

Example 3. *Consider the following CTRS $\mathcal{R}$ [24, Example 2]:*

$$\mathsf{g}(\mathsf{a}) \to \mathsf{c}(\mathsf{b}) \tag{15}$$

$$\mathsf{b} \to \mathsf{f}(\mathsf{a}) \tag{16}$$

$$\mathsf{f}(x) \to x \Leftarrow \mathsf{g}(x) \to \mathsf{c}(y) \tag{17}$$

As claimed in [24], $\mathcal{R}$ is terminating*, but it is not V-terminating. We can mechanically prove both claims using our framework (see Examples 59 and 92). Of course, we also conclude that $\mathcal{R}$ is not operationally terminating.*

As in the DP Framework for TRSs [10, 12], the notion of *processor* is central. Processors *transform termination problems* into *sets* of *simpler* termination problems which can then be handled independently. This *divide and conquer* approach is paramount in the (2D) DP Framework. Section 6 discusses the practical use of the 2D DP Framework. In contrast to the usual practice of the DP Framework for TRSs, where a single termination property (termination of a TRS) is proved or disproved, we aim at dealing with the three different properties mentioned above. This requires further flexibility, which we introduce through what we call the *open* 2D DP Framework. In Section 7, we introduce several *processors* for their use in proofs of (operational) termination of CTRSs within the (open) 2D DP Framework. We have implemented our 2D DP Framework as part of the termination tool MU-TERM [2]. Section 8 discusses related work. Section 9 concludes. In order to ease the reading of the paper, we have collected the proofs of all theorems in an appendix.

This paper is an extended and completely revised version of [23, Sections 5 and 6] and [26, Sections 4,5,7, and 8]. After the new developments in [24] leading

to a characterization of operational termination of CTRSs as the conjunction of termination and the new property of V-termination, we have revised the main essential notions (e.g., CTRS problem and processor) leading to a new, more general and powerful approach. In particular, the material in Section 6 is completely new. The specific processors discussed here have also been revised with regard to their definitions and use. Besides, we provide full proofs of all results formalizing the use of the 2D DP Framework and the aforementioned processors.

2. Preliminaries: relations, terms and substitutions

The material in this section follows [28]. A binary relation R on a set A is *terminating* (or *well-founded*) if there is no infinite sequence $a_1 \mathsf{R} a_2 \mathsf{R} a_3 \cdots$. For relations $\mathsf{R}, \mathsf{S} \subseteq A \times A$, we let $\mathsf{R} \circ \mathsf{S} = \{(a, b) \in A \times A \mid \exists c \in A, a \mathsf{R} c \wedge c \mathsf{S} b\}$. We say that R is *compatible* with S if $\mathsf{R} \circ \mathsf{S} \subseteq \mathsf{S}$ or $\mathsf{S} \circ \mathsf{R} \subseteq \mathsf{S}$

Throughout the paper $\mathcal{X}$ denotes a countable set of variables and $\mathcal{F}$ denotes a signature, i.e., a set of function symbols $\{\mathtt{f}, \mathtt{g}, \ldots\}$, each having a fixed arity given by a mapping $ar : \mathcal{F} \to \mathbb{N}$. The set of terms built from $\mathcal{F}$ and $\mathcal{X}$ is denoted $\mathcal{T}(\mathcal{F}, \mathcal{X})$. $\mathcal{V}ar(t)$ is the set of variables occurring in a term t. A term t is *ground* if it contains no variable (i.e., $\mathcal{V}ar(t) = \emptyset$). A term is said to be *linear* if it does not contain multiple occurrences of the same variable.

Terms are viewed as labeled trees in the usual way. Positions $p, q, \ldots$ are sequences of positive natural numbers used to address subterms of t. We denote the empty sequence (and root position of a term) by Λ. The set of positions of a term t is $\mathcal{P}os(t)$. Positions of nonvariable symbols in t are denoted as $\mathcal{P}os_{\mathcal{F}}(t)$. The subterm of t at position p is denoted as $t|_p$, and $t[s]_p$ is the term t with the subterm at position p replaced by s. We write $s \trianglerighteq t$, read *t is a subterm of s*, if $t = s|_p$ for some $p \in \mathcal{P}os(s)$ and $s \triangleright t$ if $s \trianglerighteq t$ and $s \neq t$. We write $s \not\triangleright t$ if $s \triangleright t$ does not hold. The symbol labeling the root of t is denoted as $root(t)$.

A substitution is a mapping $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$. The 'identity' substitution $x \mapsto x$ for all $x \in \mathcal{X}$ is denoted ε. The set $\mathcal{D}om(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$ is called the *domain* of σ. We do *not* require that the domain of the substitutions be finite. This is usual practice in the dependency pair approach, where a single substitution is used to instantiate an infinite number of variables coming from renamed versions of the dependency pairs (see below). When substitutions with *finite* domain are assumed, we explicitly call them *finite* substitutions. A *renaming* is a bijective substitution ρ such that $\rho(x) \in \mathcal{X}$ for all $x \in \mathcal{X}$. A finite substitution σ such that $\sigma(s) = \sigma(t)$ for two terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is called a *unifier* of s and t; we also say that s and t *unify* (with substitution σ). If two terms s and t unify, then there is a unique (up to renaming of variables) *most general unifier* (*mgu*) σ such that for every other unifier τ, there is a finite substitution θ such that $\theta \circ \sigma = \tau$. In the following we often write $s =^?_\sigma t$ if s and t unify with *mgu* σ. A substitution σ *unifies* a set of equations E iff[2] for

[2] In the following, *iff* abbreviates *if and only if*.

all $s =^? t \in E$, $\sigma(s) = \sigma(t)$; we also say that σ is a unifier of E.

3. Conditional rewriting, operational termination, and 2D DPs

An (*oriented*) CTRS is a pair $\mathcal{R} = (\mathcal{F}, R)$ where $\mathcal{F}$ is a signature and R a set of rules $\ell \to r \Leftarrow c$ where c is a sequence $s_1 \to t_1, \cdots, s_n \to t_n$ for some $n \geq 0$ and terms $\ell, r, s_1, \ldots, t_n$ such that $\ell \notin \mathcal{X}$. As usual, ℓ and r are called the *left-* and *right-hand sides* of the rule, and c is the *conditional part* of the rule. *Labeled* rules are written $\alpha : \ell \to r \Leftarrow c$, where α is a *label*, which is often used by itself to refer to the rule. In the following, if c is a sequence $s_1 \to t_1, \ldots, s_n \to t_n$ of conditions, and $1 \leq i \leq j \leq n$, $[c]_i^j$ denotes the subsequence $s_i \to t_i, \ldots, s_j \to t_j$; if $i > j$, then $[c]_i^j$ denotes the empty sequence. We often abbreviate $[c]_1^j$ as $[c]^j$. Also, given a CTRS $\mathcal{R}$ and a rule α, the (possible) *replacement* $\mathcal{R}[\mathcal{S}]_\alpha$ of α in $\mathcal{R}$ by the (possibly empty) set of rules $\mathcal{S}$ is $\mathcal{R}[\mathcal{S}]_\alpha = (\mathcal{R} - \{\alpha\}) \cup \mathcal{S}$ if $\alpha \in \mathcal{R}$; and $\mathcal{R}[\mathcal{S}]_\alpha = \mathcal{R}$ otherwise.

Rewrite rules $\ell \to r \Leftarrow c$ are classified according to the distribution of variables among ℓ, r, and c, as follows: *type 1*, if $\mathcal{V}ar(r) \cup \mathcal{V}ar(c) \subseteq \mathcal{V}ar(\ell)$; *type 2*, if $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)$; *type 3*, if $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell) \cup \mathcal{V}ar(c)$; and *type 4*, if no restriction is given. A rule of type n is often called an n-rule. An n-CTRS contains only m-rules for $m \leq n$. A TRS is a 1-CTRS whose rules have no conditional part; we write them $\ell \to r$. A 3-CTRS $\mathcal{R}$ is called *deterministic* if for each rule $\ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n$ in $\mathcal{R}$ and each $1 \leq i \leq n$, we have $\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l) \cup \bigcup_{j=1}^{i-1} \mathcal{V}ar(t_j)$. Given $\mathcal{R} = (\mathcal{F}, R)$, we consider $\mathcal{F}$ as the disjoint union $\mathcal{F} = \mathcal{C} \uplus \mathcal{D}$ of symbols $c \in \mathcal{C}$ (called *constructors*) and symbols $f \in \mathcal{D}$ (called *defined functions*), where $\mathcal{D} = \{root(\ell) \mid (\ell \to r \Leftarrow c) \in R\}$ and $\mathcal{C} = \mathcal{F} - \mathcal{D}$. If necessary, we may write $\mathcal{C}_\mathcal{R}$ and $\mathcal{D}_\mathcal{R}$ to make explicit the CTRS $\mathcal{R}$ which is used to establish the partitioning of $\mathcal{F}$ into constructor and defined symbols. Terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $root(t) \in \mathcal{D}$ are called *defined* terms. $\mathcal{P}os_\mathcal{D}(t)$ is the set of positions p of subterms $t|_p$ such that $root(t|_p) \in \mathcal{D}$.

3.1. Proof trees

Given an atomic formula A of the form $s \to t$ or $s \to^* t$, $pred(A)$ refers to its *predicate* symbol $\to$ or $\to^*$, respectively, and $left(A)$ refers to s. Given a CTRS $\mathcal{R}$, a finite proof tree T (for the inference system in Figure 1) is either: (i) an *open goal* G of the form $s \to t$ or $s \to^* t$ for terms s, t; then, we denote $root(T) = G$; otherwise, (ii) is a *derivation tree* with root G, denoted as

$$\frac{T_1 \quad \cdots \quad T_n}{G}(\rho) \tag{18}$$

where G is as above, $T_1, \ldots, T_n$ are finite proof trees (for $n \geq 0$), and $\rho : \frac{B_1 \ldots B_n}{A}$ is an inference rule such that $G = \sigma(A)$ and $root(T_i) = \sigma(B_i)$ for some substitution σ and $1 \leq i \leq n$. We write $root(T) = G$. A finite proof tree T is *closed* if it contains no open goals. A finite proof tree T is a *proper prefix* of a finite proof tree T' if there are one or more open goals $G_1, \ldots, G_n$ in T such that T' is obtained from T by replacing each G_i by a derivation tree T_i with root G_i. We

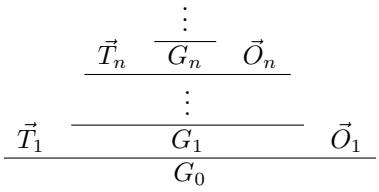


Figure 2: Structure of an infinite well-formed proof tree T

denote this as $T \subset T'$. An *infinite proof tree* T is a sequence $\{T_i\}_{i\in\mathbb{N}}$ such that for all i, $T_i \subset T_{i+1}$. We let $root(T) = root(T_0)$.

A finite proof tree T is *well-formed* if it is either an open goal, or a closed proof tree, or a derivation tree like (18) where there is i, $1 \leq i \leq n$, such that $T_1, \ldots, T_{i-1}$ are closed, T_i is a well-formed but not closed finite proof tree, and $T_{i+1}, \ldots, T_n$ are open goals. An infinite proof tree is *well-formed* if it is an increasing chain of well-formed finite proof trees. Infinite well-formed proof trees T have a *single* infinite branch as shown in Figure 2, where for all $i \geq 1$, $\vec{T}_i$ are sequences of closed (hence finite) proof trees and $\vec{O}_i$ are sequences of open goals. For all $i \geq 0$, goal G_i is the root of the infinite subtree immediately above it. Formally, T is the limit of the sequence $\{S_i\}_{i\in\mathbb{N}}$, where $S_0 = G_0$ and for all $i \geq 0$, S_{i+1} is obtained from S_i by replacing the open goal G_i by $\dfrac{\vec{T}_{i+1} \quad G_{i+1} \quad \vec{O}_{i+1}}{G_i}$. Note that, for all $i \geq 0$, $S_i \subseteq S_{i+1}$, as required. The infinite sequence $(G_i)_{i\in\mathbb{N}}$ is called the *spine* of T, written $spine(T)$.

3.2. Conditional rewriting and termination properties of CTRSs

We write $s \to_{\mathcal{R}} t$ (resp. $s \to^*_{\mathcal{R}} t$) iff there is a well-formed closed proof tree for $s \to t$ (resp. $s \to^* t$) using $\mathcal{R}$. We often drop $\mathcal{R}$ from $\to_{\mathcal{R}}$ or $\to^*_{\mathcal{R}}$ if no confusion arises. Note that $s \to_{\mathcal{R}} t$ iff there is $p \in \mathcal{P}os(s)$, $\ell \to r \Leftarrow c \in \mathcal{R}$ and a substitution σ such that $\sigma(u) \to^*_{\mathcal{R}} \sigma(v)$ for all $u \to v \in c$, $s|_p = \sigma(\ell)$ and $t = s[\sigma(r)]_p$; we often write $s \stackrel{>\Lambda}{\to}_{\mathcal{R}} t$ if $p \neq \Lambda$. It is easy to prove that $s \to^*_{\mathcal{R}} t$ holds iff there is a sequence $s_1, \ldots, s_n$ of terms for some $n \geq 1$ such that $s = s_1$, $t = s_n$ and for all i, $1 \leq i < n$, $s_i \to_{\mathcal{R}} s_{i+1}$; in particular, we write $s \stackrel{>\Lambda}{\longrightarrow}{}^* t$ iff $s \to^* t$ and for all $i \geq 0$, $s_i \stackrel{>\Lambda}{\to} s_{i+1}$ holds. Given CTRSs $\mathcal{R}$ and $\mathcal{S}$, and terms s, t, we write $s \stackrel{\Lambda}{\to}_{\mathcal{S},\mathcal{R}} t$ if there is $\ell \to r \Leftarrow c \in \mathcal{S}$ and a substitution σ such that $s = \sigma(\ell)$, $t = \sigma(r)$ and $\sigma(u) \to^*_{\mathcal{R}} \sigma(v)$ for all $u \to v \in c$; thus, the rewriting step is performed (at the root of s) by using ℓ and r from a rule $\ell \to r \Leftarrow c \in \mathcal{S}$, but the conditional part c of the rule is evaluated using $\mathcal{R}$. We call a term t [24]:

- *terminating* iff there is no infinite rewrite sequence $t = t_1 \to t_2 \to \cdots$
- *V-terminating* iff there is no infinite well-formed proof tree T such that $left(root(T)) = t$ and $pred(G) = \to$ for infinitely many goals $G \in spine(T)$.
- *operationally terminating* if there is no infinite well-formed proof tree T with $left(root(T)) = t$.

A CTRS $\mathcal{R}$ is *terminating* (resp. *V-terminating*, *operationally terminating*) iff every term t is terminating (resp. V-terminating, operationally terminating).

Let $\mathcal{R}$ be a CTRS, $\alpha : \ell \to r \Leftarrow c \in \mathcal{R}$ and σ be a substitution terminating over $\mathcal{V}ar(\ell)$ (i.e., for all $x \in \mathcal{V}ar(\ell)$, $\sigma(x)$ is terminating). We say that α *preserves termination* of σ iff σ is terminating over $\mathcal{V}ar(r)$ whenever $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ for all $s \to t \in c$. We say that $\mathcal{R}$ *preserves terminating substitutions* if for all substitutions σ and $\alpha : \ell \to r \Leftarrow c \in \mathcal{R}$, if σ is terminating over $\mathcal{V}ar(\ell)$, then α preserves termination of σ. Every 2-CTRS preserves terminating substitutions.

3.3. Dependency Pairs for CTRSs

Let $\mathcal{R}$ be a CTRS and t be a term. The set of *defined subterms* of t is $\mathcal{D}_{\trianglerighteq}(\mathcal{R}, t) = \{t|_p \mid p \in \mathcal{P}os_{\mathcal{D}_{\mathcal{R}}}(t)\}$. Let $RULES_{\Lambda}(\mathcal{R}, t) = \{\ell \to r \Leftarrow c \in \mathcal{R} \mid root(\ell) = root(t)\}$ and

$$\mathcal{U}_{\Lambda}(\mathcal{R}, t) = RULES_{\Lambda}(\mathcal{R}, t) \cup \bigcup_{\ell \to r \Leftarrow c \in RULES_{\Lambda}(\mathcal{R},t)} \mathcal{U}_{\Lambda}(\mathcal{R} - RULES_{\Lambda}(\mathcal{R}, t), r)$$

In the following, given a CTRS $\mathcal{P}$, we let $\mathcal{U}_{\Lambda}(\mathcal{R}, \mathcal{P}) = \bigcup_{\ell \to r \Leftarrow c \in \mathcal{P}} \mathcal{U}_{\Lambda}(\mathcal{R}, r)$.

Given a signature $\mathcal{F}$ and $f \in \mathcal{F}$, we let $f^\sharp$ (or just F) be a fresh symbol associated to f. For $t = f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write $t^\sharp$ to denote the *marked* term $f^\sharp(t_1, \ldots, t_k)$.

Definition 4. [24] *Let $\mathcal{R}$ be a CTRS. The CTRSs $\mathsf{DP}_H(\mathcal{R})$ and $\mathsf{DP}_V(\mathcal{R})$ of* horizontal *and* vertical *dependency pairs are*[3]*:*

$$\mathsf{DP}_H(\mathcal{R}) = \{\ell^\sharp \to v^\sharp \Leftarrow c \mid \ell \to r \Leftarrow c \in \mathcal{R}, v \in \mathcal{D}_{\trianglerighteq}(\mathcal{R}, r), \ell \not\trianglerighteq v\}$$
$$\mathsf{DP}_V(\mathcal{R}) = \{\ell^\sharp \to v^\sharp \Leftarrow [c]^{i-1} \mid \ell \to r \Leftarrow c \in \mathcal{R}, 1 \le i \le n, v \in \mathcal{D}_{\trianglerighteq}(\mathcal{R}, s_i), \ell \not\trianglerighteq v\}.$$

Now, we let $\mathsf{DP}_{VH}(\mathcal{R}) = \mathcal{U}_{\Lambda}(\mathsf{DP}_H(\mathcal{R}), \mathsf{DP}_V(\mathcal{R}))$. Following [23], we call them *connecting* dependency pairs.

Example 5. *For $\mathcal{R}$ in Example 2, $\mathsf{DP}_{VH}(\mathcal{R}) = \mathsf{DP}_H(\mathcal{R})$ because for the rhs v_{12} of pair (12) in $\mathsf{DP}_V(\mathcal{R})$, we have $\mathcal{U}_{\Lambda}(\mathsf{DP}_H(\mathcal{R}), v_{12}) = \{(10)\}$; for the rhs v_{13} of pair (13) in $\mathsf{DP}_V(\mathcal{R})$, we have $\mathcal{U}_{\Lambda}(\mathsf{DP}_H(\mathcal{R}), v_{13}) = \emptyset$; and for the rhs v_{14} of pair (14) in $\mathsf{DP}_V(\mathcal{R})$, we have $\mathcal{U}_{\Lambda}(\mathsf{DP}_H(\mathcal{R}), v_{14}) = \{(11)\}$.*

Example 6. *For $\mathcal{R}$ in Example 3,*

$$\mathsf{DP}_H(\mathcal{R}): \quad \mathsf{G}(\mathsf{a}) \to \mathsf{B} \tag{19}$$
$$\mathsf{B} \to \mathsf{F}(\mathsf{a}) \tag{20}$$
$$\mathsf{DP}_V(\mathcal{R}): \quad \mathsf{F}(x) \to \mathsf{G}(x) \tag{21}$$

and $\mathsf{DP}_{VH}(\mathcal{R}) = \mathsf{DP}_H(\mathcal{R})$.

[3]The signatures $\mathcal{F}'$ of CTRSs obtained by marking rules of other CTRSs are given by just extracting all function symbols (with their arities) from the obtained rules.

Definition 7. [24, Definition 71] *Let $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ be CTRSs. A $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain*[4] *is a finite or infinite sequence of (renamed) rules $u_i \to v_i \Leftarrow c_i \in \mathcal{P}$, which are viewed as* conditional dependency pairs*, together with a substitution σ satisfying that, for all $i \geq 1$,*

1. *for all $s \to t \in c_i$, $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ and*
2. *$\sigma(v_i)(\to_{\mathcal{R}} \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^* \sigma(u_{i+1})$.*

A $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain is called minimal *if for all $i \geq 1$, whenever*

$$\sigma(v_i) = w_{i1}(\to_{\mathcal{R}}^* \circ \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}}) w_{i2}(\to_{\mathcal{R}}^* \circ \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}}) \cdots (\to_{\mathcal{R}}^* \circ \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}}) w_{im_i} \to_{\mathcal{R}}^* \sigma(u_{i+1}),$$

in the chain, then for all j, $1 \leq j \leq m_i$, w_{ij} is $\mathcal{R}$-operationally terminating.

Theorem 8. [24, Section 7] *Let $\mathcal{R}$ be a CTRS.*

1. *If there is no infinite $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R})$-O-chain and $\mathcal{R}$ preserves terminating substitutions, then $\mathcal{R}$ is terminating.*
2. *If there is an infinite $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R})$-O-chain, then $\mathcal{R}$ is nonterminating.*
3. *If there is no infinite $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R})$-O-chain and $\mathcal{R}$ is a deterministic 3-CTRS, then $\mathcal{R}$ is V-terminating.*
4. *If there is an infinite $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R})$-O-chain, then $\mathcal{R}$ is non-V-terminating.*
5. *If there is no infinite minimal $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R})$-O-chain, there is no infinite minimal $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R})$-O-chain and $\mathcal{R}$ is a deterministic 3-CTRS, then $\mathcal{R}$ is operationally terminating.*
6. *If there is an infinite $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R})$- or $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R})$-O-chain, then $\mathcal{R}$ is operationally nonterminating.*

4. Rules with infeasible conditions

In the literature about *confluence* of conditional rewriting, the so-called *infeasible* Conditional Critical Pairs (CCPs, see [28, Definition 7.1.8(1)]) are those whose *conditional parts are not satisfiable*. The following definition borrows [28, Definition 7.1.8(3)] for feasibility of CCPs.

Definition 9. [21] *Let $\mathcal{R}$ be a CTRS and $\alpha : \ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n$ be a rule. We say that (the conditional part of) α is* $\mathcal{R}$-feasible *if there is a substitution σ such that for all $1 \leq i \leq n$, $\sigma(s_i) \to_{\mathcal{R}}^* \sigma(t_i)$. Otherwise, it is called* $\mathcal{R}$-infeasible*.*

Note that α in Definition 9 can also be a rule of $\mathcal{R}$.

[4] In [24], three notions of *chain* of dependency pairs (namely, H-, V-, and O-chains) were introduced and applied to prove the different termination properties considered in this paper (*termination*, *V-termination* and *operational termination*, respectively). *H*-chains, though, use dependency pairs that we do not consider here. Fortunately, we can use O-chains to prove all these termination properties, although with some limitations in proofs of *termination*.

Example 10. *Consider the CTRS $\mathcal{R}$ [28, Example 7.2.45]:*

$$\mathsf{a} \to \mathsf{a} \Leftarrow \mathsf{b} \to x, \mathsf{c} \to x \quad (22) \qquad \mathsf{c} \to \mathsf{d} \Leftarrow \mathsf{d} \to x, \mathsf{e} \to x \quad (24)$$
$$\mathsf{b} \to \mathsf{d} \Leftarrow \mathsf{d} \to x, \mathsf{e} \to x \quad (23)$$

where $\mathsf{a}, \ldots, \mathsf{e}$ *are* constants *and* x *is a* variable*. Clearly,* (23) *and* (24) cannot *be used in any rewriting step: since* d *and* e *are irreducible, the only way for condition* $\mathsf{d} \to x, \mathsf{e} \to x$ *to be satisfied is instantiating* x *to both* d *and* e*, which is* not *possible. They are, therefore, infeasible. Hence, (22) is infeasible as well.*

In the following, we often assume $\mathcal{R}$ partitioned as $\mathcal{R} = \mathcal{R}_F \uplus \mathcal{R}_I$, with $\mathcal{R}_I$ a (possibly empty) set of $\mathcal{R}$*-infeasible* rules of $\mathcal{R}$ and $\mathcal{R}_F$ containing any other rule. Since $\mathcal{R}$-infeasibility is, in general, undecidable, we just assume that rules in $\mathcal{R}_I$ are really (i.e., proved to be) infeasible ([21, 30, 31] develop some specific criteria for infeasibility). Some rules in $\mathcal{R}_F$ can also be infeasible, though.

Remark 11 (Disregarding infeasible rules in chains). *Provided that $\mathcal{P}_I$, $\mathcal{Q}_I$, and $\mathcal{R}_I$ consist of $\mathcal{R}$-infeasible rules only, we can replace $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ by $\mathcal{P}_F$, $\mathcal{Q}_F$, and $\mathcal{R}_F$ in Definition 7 without losing $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chains.*

Removing infeasible rules from $\mathcal{R}$ may change *nonminimality* of $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chains due to the lack of preservation of operational termination of $\mathcal{R}$ under addition of rules, even if such rules are infeasible.

Example 12. *Consider the CTRS $\mathcal{R}$*

$$\mathsf{f}(x) \to \mathsf{f}(\mathsf{b}) \quad (25)$$
$$\mathsf{b} \to \mathsf{c} \Leftarrow \mathsf{b} \to \mathsf{d}, \mathsf{c} \to \mathsf{d} \quad (26)$$

and $\mathsf{DP}_H(\mathcal{R}) = \{\mathsf{F}(x) \to \mathsf{F}(\mathsf{b}), \mathsf{F}(x) \to \mathsf{B}\}$*. The* $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R})$*-O-chain:*

$$\mathsf{F}(x_1) \to \mathsf{F}(\mathsf{b}), \mathsf{F}(x_2) \to \mathsf{F}(\mathsf{b}), \ldots \quad (27)$$

with $\sigma(x_i) = \mathsf{b}$ *for all* $i \geq 1$ *is* not *minimal because* b *is not $\mathcal{R}$-operationally terminating. However, if we remove* (26) *(clearly $\mathcal{R}$-infeasible), then (27) becomes minimal (with respect to $\mathcal{R} - \{(26)\}$).*

Remark 13 (Feasibility assumption). *According to Remark 11 and taking into account Example 12, in the following we often assume that all rules in $\mathcal{P}$ and $\mathcal{Q}$ are $\mathcal{R}$-feasible in all considered $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chains.*

5. The 2D DP Framework for CTRSs

In the following, $\mathsf{F} = \{\mathsf{a}, \mathsf{m}\}$ is a signature of *flag constants* f referring to arbitrary (resp. minimal) $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chains if $f = \mathsf{a}$ (resp. $f = \mathsf{m}$). We write $\mathsf{m} < \mathsf{a}$ to denote that arbitrary O-chains *include* minimal ones. Let's introduce the *problems* at stake in our *incremental* proofs of CTRS termination properties.

Definition 14 (CTRS problem). *A CTRS problem is a tuple $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$, where $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ are CTRSs, and $f \in \mathsf{F}$.*

Accordingly, we speak of $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m})$-chains (or just τ-chains if $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m})$) instead of *minimal* $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chains; and of $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{a})$-chains (resp. τ-chains if $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{a})$) instead of *arbitrary* $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chains.

Definition 15 (Finite CTRS problem). *A CTRS problem $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ with $f \in \mathsf{F}$ is* finite *iff there is no infinite τ-chain; τ is* infinite *iff there is an infinite τ-chain (iff τ is* not *finite).*

Note that $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ is trivially *finite* if $\mathcal{P}$ contains no rule (written $\mathcal{P} = \emptyset$ for short). We call them *trivial* CTRS problems. We can now recast Theorem 8 as follows:

Theorem 16. *Let $\mathcal{R}$ be a CTRS.*

1. *If $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, \mathsf{a})$ is finite and $\mathcal{R}$ preserves terminating substitutions, then $\mathcal{R}$ is* terminating.
2. *If $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, f)$ is infinite for some $f \in \mathsf{F}$, then $\mathcal{R}$ is* nonterminating.
3. *If $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, \mathsf{a})$ is finite and $\mathcal{R}$ is a deterministic 3-CTRS, then $\mathcal{R}$ is V-*terminating.
4. *If $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, f)$ is infinite for some $f \in \mathsf{F}$, then $\mathcal{R}$ is* non-V-terminating.
5. *If both $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, f)$ and $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, f')$ are finite for some $f, f' \in \mathsf{F}$ and $\mathcal{R}$ is a deterministic 3-CTRS, then $\mathcal{R}$ is* operationally terminating.
6. *If $(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, f)$ or $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, f')$ are infinite for some $f, f' \in \mathsf{F}$, then $\mathcal{R}$ is* operationally nonterminating.

The *DP problems* of [10, 12] are infinite already if $\mathcal{R}$ is nonterminating and there are DP problems that are both finite and infinite (e.g., with $\mathcal{P}$ a TRS with an *empty* set of rules and $\mathcal{R}$ consisting of the rule $\mathsf{a} \to \mathsf{a}$). This is *not* possible for CTRS problems. Actually, declaring a CTRS problem $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ *infinite* if $\mathcal{R}$ is *operationally nonterminating* would risk reaching wrong conclusions.

Example 17. *The CTRS $\mathcal{R}$ in Example 3 is not operationally terminating. But it* is *terminating (see Example 59). If $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ is infinite whenever $\mathcal{R}$ is operationally nonterminating, by Theorem 16(2) we would* wrongly *conclude nontermination of $\mathcal{R}$. Similarly, every nonterminating TRS $\mathcal{R}$, viewed as a CTRS without conditional rules, is V-terminating (because $\mathsf{DP}_V(\mathcal{R}) = \emptyset$). However, considering τ infinite whenever $\mathcal{R}$ is* nonterminating *would wrongly lead us to conclude non-V-termination of TRSs (using Theorem 16(4)).*

See Section 8 for a more detailed comparison with the DP Framework for TRSs.

5.1. Processors

Processors *transform* CTRS problems $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ into possibly empty sets $\{\tau_1, \ldots, \tau_n\}$ of (*simpler*) CTRS problems $\tau_i = (\mathcal{P}_i, \mathcal{Q}_i, \mathcal{R}_i, f_i)$. 'Simpler' often means that the $\mathcal{P}_i$ (resp. $\mathcal{Q}_i$, $\mathcal{R}_i$) are smaller than $\mathcal{P}$ (resp. $\mathcal{Q}$, $\mathcal{R}$). Some processors, though, *transform* the rules in $\mathcal{P}$, $\mathcal{Q}$, or $\mathcal{R}$ so that the obtained CTRS problems define 'fewer' chains. Processors may also return *"*no*"*, with the intended meaning of τ being *infinite* (but see Definition 22 below).

Definition 18 (CTRS processor). *A* CTRS processor P *is a partial function from CTRS problems into sets of CTRS problems. Alternatively, it can return* "no"*. The* domain *of* P*, denoted* $\mathsf{Dom}(\mathsf{P})$*, is the set of CTRS problems* τ *for which* P *is defined.*

In the literature, processors are usually defined to be *mappings*, i.e., total functions. We can easily fit this setting by just *completing* the definition of a CTRS processor P into a mapping $\overline{\mathsf{P}}$ as follows:

$$\overline{\mathsf{P}}(\tau) = \begin{cases} \mathsf{P}(\tau) & \text{if } \tau \in \mathsf{Dom}(\mathsf{P}) \\ \{\tau\} & \text{otherwise} \end{cases}$$

Indeed, most definitions of processors in the literature have *this shape*. This suggests that considering processors as partial functions may be more natural. In the following we will speak of 'processors' rather than 'CTRS processors' if no confusion arises. The definition of our first simple processor P_{Fin} illustrates the following notational conventions to be followed in the sequel:

1. We often write $\mathsf{P}(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ rather than $\mathsf{P}((\mathcal{P}, \mathcal{Q}, \mathcal{R}, f))$ to avoid duplicate parentheses in the definition of processor P.
2. When defining P we provide *the conditions that CTRS problems* τ *must satisfy* (if any) for P to be applied.

Definition 19 (Trivial finiteness processor). *Let* $\mathcal{P}$*,* $\mathcal{Q}$*, and* $\mathcal{R}$ *be CTRSs and* $f \in \mathsf{F}$*. Then,* P_{Fin} *is given by* $\mathsf{P}_{Fin}(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) = \emptyset$*, if* $\mathcal{P} = \emptyset$*.*

Thus, in Definition 19, the *condition* for the application of P_{Fin} is "$\mathcal{P} = \emptyset$". The following processor detects a simple kind of infinite CTRS problems.

Definition 20 (Infiniteness processor). *Let* $\mathcal{P}$*,* $\mathcal{Q}$*, and* $\mathcal{R}$ *be CTRSs,* $u \to v \Leftarrow c \in \mathcal{P}$*, and* $f \in \mathsf{F}$*. Then,* P_{Inf} *is given by* $\mathsf{P}_{Inf}(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) = \mathsf{no}$ *if there are substitutions* θ, η *such that for all* $s \to t \in c$*,* $\eta(s) \to_{\mathcal{R}}^* \eta(t)$ *and* $\eta(v) = \theta(\eta(u))$*.*

Ensuring that the reachability conditions $\eta(s) \to_{\mathcal{R}}^* \eta(t)$ hold is essential for making sense of a 'negative' answer of the processor.

Example 21. *Consider the CTRS* $\mathcal{P} = \{\mathsf{a} \to \mathsf{a} \Leftarrow \mathsf{b} \to \mathsf{c}\}$ *and* $\mathcal{Q} = \mathcal{R} = \emptyset$*. The rule* $u \to v \Leftarrow c$ *in* $\mathcal{P}$ *satisfies the condition* $\eta(v) = \theta(\eta(u))$ *if* $\eta = \theta = \varepsilon$*, the empty substitution. However, there is no infinite* $(\mathcal{P}, \mathcal{R}, \mathcal{Q})$*-O-chain because the condition* $\mathsf{b} \to_{\mathcal{R}}^* \mathsf{c}$ *in* c *cannot be satisfied due to the emptiness of* $\mathcal{R}$*.*

Rules $u \to v \in \mathcal{P}$ *without conditional part* are suitable candidates to be used with P_{Inf} as we do not need to check the condition $\eta(s) \to_{\mathcal{R}}^* \eta(t)$ prescribed in Definition 20 (hence we let η be the identity substitution). For pairs $u \to v \Leftarrow c \in \mathcal{P}$ where c is not empty, we can use the processor only if we find a substitution η such that $\eta(s) \to_{\mathcal{R}}^* \eta(t)$ holds for all $s \to t \in c$. In general, this is not computable. However, we can easily apply P_{Inf} if there is a substitution η such that $\eta(s) = \eta(t)$ for all $s \to t \in c$ (unification) and $\eta(v) = \theta(\eta(u))$ for some substitution θ (related to semi-unification [17]). Another simple case is when all conditions $s \to t \in c$ are *instances* of unconditional rules $\ell \to r \in \mathcal{R}$, i.e., $s \to t = \theta(\ell \to r)$ for some substitution θ.

5.2. Soundness and Completeness

The most relevant properties to be established when using CTRS processors in the 2D DP Framework are *soundness* and *completeness*. Soundness is essential to prove CTRS problems *finite* by using combinations of processors (see Section 5.3); completeness for proving *infiniteness*.

Definition 22 (Soundness and completeness). *Let* P *be a processor and* $\tau \in \mathsf{Dom}(\mathsf{P})$. *We say that* P *is:*

- τ-sound *iff* τ *is finite whenever* $\mathsf{P}(\tau) \neq \mathsf{no}$ *and for all* $\tau' \in \mathsf{P}(\tau)$, τ' *is finite.*
- τ-complete *iff* τ *is infinite whenever* $\mathsf{P}(\tau) = \mathsf{no}$ *or there is* $\tau' \in \mathsf{P}(\tau)$ *such that* τ' *is infinite.*

Accordingly:

1. *Given* $f \in \mathsf{F}$, *we say that* P *is* f-sound *(resp.* f-complete*) if it is* τ*-sound (*τ*-complete) for all* $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f') \in \mathsf{Dom}(\mathsf{P})$ *such that* $f = f'$.
2. P *is* sound *(resp.* complete*) if it is* f*-sound (*f*-complete) for all* $f \in \mathsf{F}$.

Theorem 23. P_{Fin} *is sound and complete.* P_{Inf} *is sound and* a*-complete.* P_{Inf} *is* τ*-complete if* $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m})$ *and* $u \to v \Leftarrow c$ *in Definition 20 is such that* v *is ground and contains no symbol from* $\mathcal{D}_{\mathcal{R}}$.

P_{Inf} may *fail* to be τ-complete for CTRS problems $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ with $f = \mathsf{m}$.

Example 24. *Let* $\mathcal{P} = \{\mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{a})\}$, $\mathcal{Q} = \emptyset$, *and* $\mathcal{R} = \{\mathsf{a} \to \mathsf{a}\}$. *Then, there is no minimal* $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$*-O-chain. Therefore,* $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m})$ *is finite. However,* $\mathsf{P}_{Inf}(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m}) = \mathsf{no}$. *Thus,* P_{Inf} *is* not τ*-complete.*

P_{Inf} illustrates the typical use of the flag component f of CTRS problems $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ in the description of processors. Although f plays *no role* in the *definition* of P_{Inf}, it is crucial to prove it *complete*. A key point is noticing two essential *roles* played by processors in the 2D DP Framework, which rely on *different components* of CTRS problems τ:

1. Processors are *defined* as *transformations* of CTRS problems τ into sets of CTRS problems (may return no). Such a *transformation* usually involves the CTRS components $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ of τ only (see Definitions 19 and 20).
2. Processors are qualified to be *sound* and/or *complete*. Such a qualification often depends on the flag $f \in \mathsf{F}$ of τ only (see Theorem 23).

The following result is used to formalize the use of the processors in Section 7.

Proposition 25. *Let* P *be a processor such that, for all CTRS problems* $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) \in \mathsf{Dom}(\mathsf{P})$, $\mathsf{P}(\tau) \neq \mathsf{no}$ *and if* $\tau' = (\mathcal{P}', \mathcal{Q}', \mathcal{R}', f') \in \mathsf{P}(\tau)$, *then* $\mathcal{P}' \subseteq \mathcal{P}$, $\mathcal{Q}' \subseteq \mathcal{Q}$, $\mathcal{R}' \subseteq \mathcal{R}$ *and (i)* $f = \mathsf{a}$ *or (ii)* $f' = f = \mathsf{m}$ *and* $\mathcal{R}' = \mathcal{R}$. *Then,* P *is complete.*

If $f' = \mathsf{a}$ and $f = \mathsf{m}$ in Proposition 25, completeness may fail to hold.

Example 26. *Let* P *be a processor such that for all CTRSs* $\mathcal{P}$, $\mathcal{Q}$, *and* $\mathcal{R}$, *and* $f \in \mathsf{F}$, $\mathsf{P}(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) = \{(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{a})\}$. P *is* not *complete: for* $\mathcal{P}$ *and* $\mathcal{R}$ *consisting of the rule* $\mathsf{b} \to \mathsf{b}$ *there is an infinite* $(\mathcal{P}, \emptyset, \mathcal{R})$*-chain:* $\mathsf{b} \to \mathsf{b}, \mathsf{b} \to \mathsf{b}, \ldots$, *but there is no minimal* $(\mathcal{P}, \emptyset, \mathcal{R})$*-O-chain (because '*$\mathsf{b}$*' is not* $\mathcal{R}$*-terminating).*

Corollary 27. *Let* P *be a processor such that, for all CTRS problems* $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) \in \mathsf{Dom}(\mathsf{P})$, $\mathsf{P}(\tau) \neq \mathsf{no}$ *and if* $\tau' = (\mathcal{P}', \mathcal{Q}', \mathcal{R}', f') \in \mathsf{P}(\tau)$, *then* $\mathcal{P}' \subseteq \mathcal{P}$, $\mathcal{Q}' \subseteq \mathcal{Q}$, $\mathcal{R}' = \mathcal{R}$ *and* $f' \leq f$. *Then,* P *is complete.*

Proposition 25 and Corollary 27 concern processors P which are *unable* to qualify a CTRS problem τ as infinite (by returning "no").

5.3. CTRS Proof Trees and the 2D DP Framework

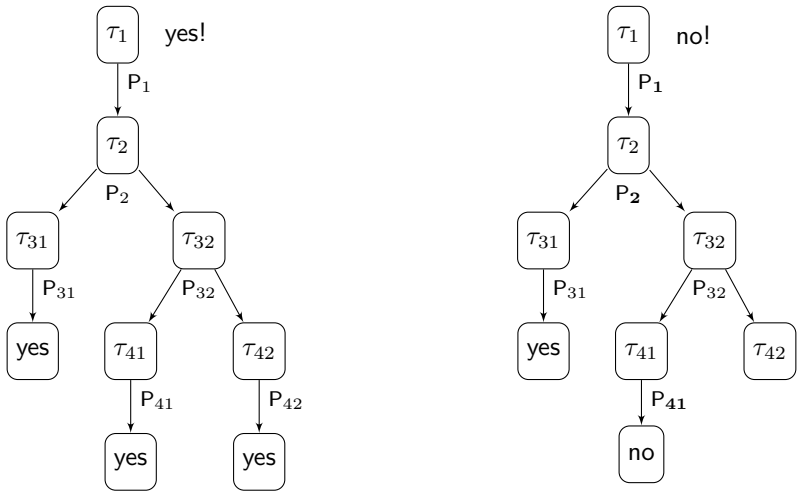
Processors are used in a *divide and conquer* scheme to incrementally simplify the *initial* CTRS problem τ_I, possibly decomposing it into (a tree of) smaller problems which are independently treated in the same way.

Definition 28 (CTRS Proof Tree). *Let* τ_I *be a CTRS problem. A* CTRS Proof tree $\mathcal{T}$ *(CTRSP-tree for short) for* τ_I *is a tree whose nodes are labeled with CTRS problems; the leaves may also be labeled with either "*yes*" or "*no*". The root of* $\mathcal{T}$ *is labeled with* τ_I. *For every inner node* n *with label* τ, *there is a processor* P *such that* $\tau \in \mathsf{Dom}(\mathsf{P})$ *and:*

1. *If* $\mathsf{P}(\tau) = \mathsf{no}$, *then* n *has just one child* n' *with label "*no*".*
2. *If* $\mathsf{P}(\tau) = \emptyset$, *then* n *has just one child* n' *with label "*yes*".*
3. *If* $\mathsf{P}(\tau) = \{\tau_1, \ldots, \tau_k\}$ *with* $k > 0$, *then* n *has exactly* k *children* $\mathsf{n}_1, \ldots, \mathsf{n}_k$ *with labels* $\tau_1, \ldots, \tau_k$, *respectively.*

Theorem 29 (2D DP Framework). *Let* τ *be a CTRS problem and* $\mathcal{T}$ *be a CTRSP-tree for* τ. *Then,*

1. *If all leaves in* $\mathcal{T}$ *are labeled with "*yes*" and all involved processors are* sound *for the CTRS problems they are applied to, then* τ *is* finite.



All involved processors are *sound* $\mathsf{P}_1, \mathsf{P}_2, \mathsf{P}_{32}$, and P_{41} are *complete*

Figure 3: Proving finiteness/infiniteness of CTRS problems in the 2D DP Framework

2. *If $\mathcal{T}$ has a leaf with label "*no*" and all processors from τ to the leaf are* complete *for the CTRS problems they are applied to, then τ is* infinite.

Figure 3 illustrates Theorem 29. The leftmost CTRSP-tree shows a proof of finiteness of τ_1; note that all leaves are labeled with '**yes**' and all involved processors are *sound*. The rightmost CTRSP-tree shows a proof of infiniteness. As soon as a leaf is labeled with '**no**', we *stop* the development of the tree and check whether all processors involved in the *path* to the leaf are *complete*; note that other leaves could be labeled with '**yes**' due to the specific generation strategy for the tree. And other nodes (e.g., τ_{42}) could remain *unexplored*.

Term. prop.	**Initial CTRS problems**	**Requirements on $\mathcal{R}$**	**F/I**
Term.	$(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, \mathsf{a})$	preserves terminating substitutions	F
Nonterm.	$(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, f),\ f \in \mathsf{F}$	None	I
V-Term.	$(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, \mathsf{a})$	deterministic 3-CTRS	F
Non-V-Term.	$(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, f),\ f \in \mathsf{F}$	None	I
Op. term.	$(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, f),\ f \in \mathsf{F}$ $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, f'),\ f' \in \mathsf{F}$	deterministic 3-CTRS	F F
Op. nonterm.	$(\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, f),\ f \in \mathsf{F}$	None	I
Op. nonterm.	$(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, f),\ f \in \mathsf{F}$	None	I

Table 1: CTRS problems for proving termination properties

Following Theorem 16, Table 1 shows the *initial CTRS problems* to be used with Theorem 29 to prove termination properties, including the requirements

on $\mathcal{R}$ which must be satisfied, and the condition (F)inite or (I)nfinite we need to prove for each CTRS problem to conclude the desired property.

By Theorem 8(3) and [24, Proposition 78] V-termination can be proved as finiteness of $(\mathsf{DP}_V(\mathcal{R}) \cup \mathsf{DP}_{VH}(\mathcal{R}), \emptyset, \mathcal{R}, \mathsf{a})$. And, by [24, Theorem 81], *operational termination* of $\mathcal{R}$ can be (dis)proved as (in)finiteness of $(\mathsf{DP}_H(\mathcal{R}) \cup \mathsf{DP}_V(\mathcal{R}), \emptyset, \mathcal{R}, f)$, with $f \in \mathsf{F}$. Thus, the following could be added to Table 1:

Term. prop.	**Initial CTRS problems**	**Requirements on $\mathcal{R}$**	**F/I**
V-term.	$(\mathsf{DP}_V(\mathcal{R}) \cup \mathsf{DP}_{VH}(\mathcal{R}), \emptyset, \mathcal{R}, \mathsf{a})$	deterministic 3-CTRS	F
Op. term.	$(\mathsf{DP}_H(\mathcal{R}) \cup \mathsf{DP}_V(\mathcal{R}), \emptyset, \mathcal{R}, f)$, $f \in \mathsf{F}$	deterministic 3-CTRS	F
Op. nonterm.	$(\mathsf{DP}_H(\mathcal{R}) \cup \mathsf{DP}_V(\mathcal{R}), \emptyset, \mathcal{R}, f)$, $f \in \mathsf{F}$	None	I

However, non-V-termination cannot be proved by just moving $\mathsf{DP}_{VH}(\mathcal{R})$ to $\mathsf{DP}_V(\mathcal{R})$ and then proving infiniteness of $(\mathsf{DP}_V(\mathcal{R}) \cup \mathsf{DP}_{VH}(\mathcal{R}), \emptyset, \mathcal{R}, \mathsf{a})$.

Example 30. *Consider the following CTRS $\mathcal{R}$:*

$$\mathsf{b} \to \mathsf{c} \quad (28) \qquad \mathsf{a} \to \mathsf{d} \Leftarrow \mathsf{b} \to \mathsf{c} \quad (31)$$
$$\mathsf{c} \to \mathsf{b} \quad (29) \qquad \mathsf{b} \to \mathsf{d} \Leftarrow \mathsf{e} \to \mathsf{f} \quad (32)$$
$$\mathsf{e} \to \mathsf{f} \quad (30) \qquad \mathsf{c} \to \mathsf{d} \Leftarrow \mathsf{e} \to \mathsf{f} \quad (33)$$

together with

$$\mathsf{DP}_H(\mathcal{R}) : \mathsf{B} \to \mathsf{C} \quad (34) \qquad \mathsf{DP}_V(\mathcal{R}) : \mathsf{A} \to \mathsf{B} \quad (36)$$
$$\mathsf{C} \to \mathsf{B} \quad (35) \qquad \mathsf{B} \to \mathsf{E} \quad (37)$$
$$\mathsf{C} \to \mathsf{E} \quad (38)$$

Since $\mathsf{DP}_{VH}(\mathcal{R}) = \mathsf{DP}_H(\mathcal{R})$, there is an infinite $(\mathsf{DP}_V(\mathcal{R}) \cup \mathsf{DP}_{VH}(\mathcal{R}), \emptyset, \mathcal{R})$-O-chain $\mathsf{B} \to_{\mathsf{DP}_{VH}(\mathcal{R})} \mathsf{C} \to_{\mathsf{DP}_{VH}(\mathcal{R})} \mathsf{B} \to \cdots$ However, $\mathcal{R}$ is V-terminating (see Example 61), i.e., there is no infinite $(\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R})$-O-chain.

Remark 31 (Which kind of CTRS problems?). *From the previous discussion, the following question arises: do we really need CTRS problems with* three *CTRSs $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{R}$ instead of just $\mathcal{P}$ and $\mathcal{R}$? Some reasons for our choice are:*

1. *With CTRS problems like $(\mathcal{P}, \mathcal{R}, f)$, we do* not *capture non-V-termination (Example 30).*
2. *Processors are more accurate due to a more precise characterization of the role of the different kind of rules in the investigated termination property.*
3. *Several processors take* advantage *of the distinction between $\mathcal{P}$ and $\mathcal{Q}$.*

6. The 2D DP Framework in practice: the Open 2D DP Framework

The 2D DP Framework can be used to prove *several* termination properties. Depending on the targeted property, we may start with CTRS problems $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ where $f = \mathsf{a}$ or $f = \mathsf{m}$ (see Table 1). In practice, we often want to *analyze* the termination behavior of a *given* CTRS *without focusing any target* property. For this purpose, we introduce an *Open 2D DP Framework* where

the proofs are driven by the use of processors (viewed as CTRS problem transformers) rather than by (specific kinds of) CTRS problems. We proceed as follows:

1. We *generalize* the notion of CTRS problem into a notion of *open* CTRS problem, where the *flag* component f is left *unspecified* and treated now as a *variable*. We define *open* processors to work with open CTRS problems.
2. We define the notion of *plugging scheme* to capture the *requirements* on the flags in CTRS problems that guarantee *soundness* and *completeness*.
3. We use open processors to build an *open* CTRS proof tree, where processors transform open CTRS problems and plugging schemes are used to track their use in proofs of finiteness/infiniteness.

6.1. Open CTRS problems and processors

In the following, V is a set of *flag variables* $f, f', \ldots$ which are intended to range over F; the terms of signature F are $\mathcal{T}(\mathsf{F},\mathsf{V}) = \mathsf{V} \cup \mathsf{F}$.

Definition 32 (Open CTRS problem). *An open CTRS problem (OCTRS problem for short) is a tuple $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$, where $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ are CTRSs, and $\varphi \in \mathcal{T}(\mathsf{F},\mathsf{V})$.*

The usual notions of *substitution*, *ground substitution*, *instantiation by a substitution*, etc., apply naturally to these special variables and constant symbols.

Definition 33 (Flag substitution). *A substitution $\varsigma : \mathsf{V} \to \mathcal{T}(\mathsf{F},\mathsf{V})$ is called a* flag substitution. *We denote as ς_{a} (resp. ς_{m}) the* constant *(and obviously ground) flag substitution that replaces every flag variable by a (resp. m).*

Definition 34 (Instance of an open CTRS problem). *Let $\varsigma : \mathsf{V} \to \mathcal{T}(\mathsf{F},\mathsf{V})$ be a flag substitution. The* instance *by ς (or ς-instance) of an OCTRS problem $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ is the (possibly open) CTRS problem $\varsigma(\check{\tau}) = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varsigma(\varphi))$.*

The instantiation of an open CTRS problem (Definition 32) by a *ground* flag substitution ς yields a CTRS problem (Definition 14).

Definition 35 (Open CTRS processor). *An* open *CTRS processor P is a partial function from OCTRS problems $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ into sets of OCTRS problems; alternatively, it can return "no". If $\mathsf{P}(\check{\tau}) \neq \emptyset$, then, for all $\check{\tau}' = (\mathcal{P}', \mathcal{Q}', \mathcal{R}', \varphi') \in \mathsf{P}(\check{\tau})$, we have $\varphi' \in \{\mathsf{a}, \mathsf{m}, \varphi\}$ (no new flag variables are introduced). The set of open CTRS problems for which P is defined is $\mathsf{Dom}^{\check{}}(\mathsf{P})$.*

Processors like P_{Fin} and P_{Inf}, whose definition over CTRS problems $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ does *not* use the value of f, can be seen as *open* CTRS processors.

Definition 36 (Soundness and completeness). *Let P be an open processor, $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi) \in \mathsf{Dom}^{\check{}}(\mathsf{P})$, and $f \in \mathsf{F}$ be such that $\varsigma_f(\varphi) = f$. P is said to be:*

- *$(\check{\tau}, f)$-sound iff $\varsigma_f(\check{\tau})$ is finite whenever $\mathsf{P}(\check{\tau}) \neq \mathsf{no}$ and for all $\check{\tau}' \in \mathsf{P}(\check{\tau})$, $\varsigma_f(\check{\tau}')$ is finite.*

- $(\check{\tau}, f)$-complete *iff* $\varsigma_f(\check{\tau})$ *is infinite whenever* $\mathsf{P}(\check{\tau}) = \mathsf{no}$ *or there is* $\check{\tau}' \in \mathsf{P}(\check{\tau})$ *such that* $\varsigma_f(\check{\tau}')$ *is infinite.*

Accordingly:

1. *Given* $f \in \mathsf{F}$*, we say that* P *is* f-sound *(resp.* f-complete*) if it is* $(\check{\tau}, f)$-*sound (*$(\check{\tau}, f)$-*complete) for all* $\check{\tau} \in \mathsf{Dom}^{\check{}}(\mathsf{P})$.
2. *Given* $\check{\tau} \in \mathsf{Dom}^{\check{}}(\mathsf{P})$*, we say that* P *is* $\check{\tau}$-sound *(resp.* $\check{\tau}$-complete*) if it is* $(\check{\tau}, f)$-*sound (*$(\check{\tau}, f)$-*complete) for all* $f \in \mathsf{F}$.
3. P *is* sound *(resp.* complete*) if it is* f-*sound (*f-*complete) for all* $f \in \mathsf{F}$.

Theorem 23 proves P_{Fin} (viewed as an *open* processor) both sound and complete. It also proves P_{Inf} sound and a-*complete* (and m-complete in some specific cases).

6.2. Plugging schemes

Soundness and completeness of processors for CTRS problems $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ may require specific values for f. The role that flag variables play during the application of an open processor to an open CTRS problem is made *explicit* by means of *plugging schemes*.

Definition 37 (Plugging scheme). *A* plugging scheme *is a pair* $\psi = \langle \varphi_s, \varphi_c \rangle$*, where* $\varphi_s, \varphi_c \in \mathcal{T}(\mathsf{F}^\bullet, \mathsf{V}) = \mathsf{F}^\bullet \cup \mathsf{V}$ *for* $\mathsf{F}^\bullet = \mathsf{F} \cup \{\bullet\}$ *with '*$\bullet$*' a new constant symbol. We call* φ_s *and* φ_c *the* soundness *and* completeness *components of the plugging scheme, respectively.*

The symbol $\bullet$ is used to *exclude* a given (sound or complete) use of P. In the following, when using plugging schemes, we will assume that they do not share any flag variable with any other plugging scheme or open CTRS problem. Renamed flag variables can be used if necessary.

Definition 38. *We say that an open processor* P follows *a plugging scheme* $\psi = \langle \varphi_s, \varphi_c \rangle$ *with* $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi) \in \mathsf{Dom}^{\check{}}(\mathsf{P})$ *iff for all* $f \in \mathsf{F}$*,*

1. *If* $\varsigma_f(\varphi) = \varsigma_f(\varphi_s)$*, then* P *is* $(\check{\tau}, f)$*-sound; and*
2. *If* $\varsigma_f(\varphi) = \varsigma_f(\varphi_c)$*, then* P *is* $(\check{\tau}, f)$*-complete.*

If P *follows* ψ *with all CTRS problems* $\check{\tau} \in \mathsf{Dom}^{\check{}}(\mathsf{P})$*, we say that* P *follows* ψ.

Sound and complete processors follow *every* plugging scheme. For instance, P_{Fin} is sound and complete (Theorem 23) and follows $\psi_{\mathsf{P}_{Fin}} = \langle f, f \rangle$. Also, the plugging scheme $\langle \bullet, \bullet \rangle$ is trivially followed by *every* processor P.

Example 39. *By Theorem 23,* P_{Inf} *follows* $\psi = \langle f, \mathsf{a} \rangle$*, as* P_{Inf} *is sound and* a*-complete. Also,* P_{Inf} *follows* $\psi' = \langle \bullet, \mathsf{a} \rangle$*, since the presence of* $\bullet$ *in the soundness component trivially makes condition 1 in Definition 38 true. Indeed,* ψ' *is 'more precise' than* ψ*:* P_{Inf} *will* never *be properly used in any proof of finiteness of CTRS problems as it always returns "*no*". The use of '*$\bullet$*'* disallows *calls to* P_{Inf} *in proofs of* finiteness*. Finally, note that* P_{Inf} *follows* $\psi_{\mathsf{P}_{Fin}}$ *with* $\check{\tau}$ *only if* $\check{\tau}$ *satisfies certain conditions (see Theorem 23): for* $\check{\tau}' = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m})$ *with* $\mathcal{P}$*,* $\mathcal{Q}$*, and* $\mathcal{R}$ *as in Example 24, we have* $\mathsf{m} = \varsigma_{\mathsf{m}}(f)$*. However,* P_{Inf} *is* not τ'-complete*. Thus, we may also need* $\langle \bullet, f \rangle$ *(better than* $\psi_{\mathsf{P}_{Fin}}$*) for* P_{Inf}.

Example 39 shows that a processor P may follow *several* plugging schemes; actually, as we will see below, this can be necessary for some processors to deal with different CTRS problems. In general, we will *choose* a *set* $\Pi(\mathsf{P})$ of plugging schemes associated to a given (open) processor P. This set is obtained from the soundness and completeness results for P.

Example 40. *According to the discussion above, we let* $\Pi(\mathsf{P}_{Fin}) = \{\langle f, f\rangle\}$ *and* $\Pi(\mathsf{P}_{Inf}) = \{\langle \bullet, \mathsf{a}\rangle, \langle \bullet, f\rangle\}$.

The next section explains the use of plugging schemes in proofs of finiteness and infiniteness of CTRS problems in the Open 2D DP Framework.

6.3. Open CTRS proof trees

In this section we adapt the CTRSP-trees to deal with *open* CTRS problems. The nodes are now labeled with *open* CTRS problems *and plugging schemes*.

Definition 41 (Open CTRS Proof Tree). *Let* $\check{\tau}_I$ *be an OCTRS problem. An* open CTRS Proof tree $\check{\mathcal{T}}$ *(OCTRSP-tree) for* $\check{\tau}_I$ *is a tree whose nodes are labeled with an OCTRS problem and a plugging scheme; the leaves may also be labeled with either* "yes" *or* "no"*. The root of* $\check{\mathcal{T}}$ *is labeled with* $\check{\tau}_I$*. For every inner node* n *with labels* $\check{\tau}$ *and* ψ*, there is an open processor* P *such that (i)* $\check{\tau} \in \mathsf{Dom}^{\vee}(\mathsf{P})$*, (ii)* P *follows* ψ *with* $\check{\tau}$*, and:*

1. *If* $\mathsf{P}(\check{\tau}) = \mathsf{no}$*, then* n *has just one child* n' *with label* "no"*.*
2. *If* $\mathsf{P}(\check{\tau}) = \emptyset$*, then* n *has just one child* n' *with label* "yes"*.*
3. *If* $\mathsf{P}(\check{\tau}) = \{\check{\tau}_1, \ldots, \check{\tau}_k\}$ *with* $k > 0$*, then* n *has exactly* k *children* $\mathsf{n}_1, \ldots, \mathsf{n}_k$ *with labels* $\check{\tau}_1, \ldots, \check{\tau}_k$*, respectively.*

The labeling of a node with labels $\check{\tau}$ and $\psi = \langle \varphi_s, \varphi_c\rangle$ is displayed as follows:

$$\boxed{\varphi_s \,[\,\check{\tau}\,]\, \varphi_c}$$

Figure 4 shows an open CTRSP-tree with leaves holding labels L_{31}, L_{41}, and L_{42} which can be either yes or no. Proofs of termination properties of CTRSs in the *Open* 2D DP Framework only consider two *initial* open CTRS problems:

$$\check{\tau}^H = (\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, f) \quad \text{and} \quad \check{\tau}^V = (\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, f) \qquad (39)$$

where $f \in \mathsf{V}$, each with a single open CTRSP-tree. At the end, flag substitutions are obtained describing the *possible* instantiations of f. The interplay between the different soundness and completeness plugs and the labels in the OCTRS problems is handled by means of two sets of *equations*.

Definition 42 (Soundness and completeness equations). *Given an open CTRSP-tree* $\check{\mathcal{T}}$*,*

1. *the set* $\mathcal{E}_s(\check{\mathcal{T}})$ *of* soundness equations *for* $\check{\mathcal{T}}$ *consists of an equation* $\varphi =^? \varphi_s$ *for each inner node* n *labeled with* $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ *and* $\langle \varphi_s, \varphi_c\rangle$*.*

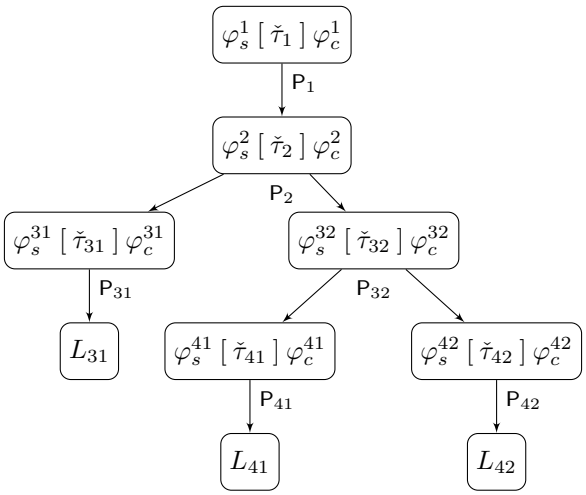


Figure 4: An open CTRSP-tree

2. *the set* $\mathcal{E}_c^{\mathsf{L}}(\check{\mathcal{T}})$ *of* completeness equations *for the* path Γ *in* $\check{\mathcal{T}}$ *leading from the root of* $\check{\mathcal{T}}$ *to a leaf* L *consists of an equation* $\varphi =^? \varphi_c$ *for each node* n *in* Γ *labeled with* $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ *and* $\langle \varphi_s, \varphi_c \rangle$.

For instance, for the OCTRSP-tree $\mathcal{T}$ in Figure 4, we have:

$$\begin{array}{rcl} \mathcal{E}_s(\check{\mathcal{T}}) & = & \{\ \ \varphi_1 =^? \varphi_s^1,\ \varphi_2 =^? \varphi_s^2,\ \varphi_{31} =^? \varphi_s^{31},\\ & & \quad\ \varphi_{32} =^? \varphi_s^{32},\ \varphi_{41} =^? \varphi_s^{41},\ \varphi_{42} =^? \varphi_s^{42}\ \ \} \\ \mathcal{E}_c^{L_{41}}(\check{\mathcal{T}}) & = & \{\ \ \varphi_1 =^? \varphi_c^1,\ \varphi_2 =^? \varphi_c^2, \varphi_{32} =^? \varphi_c^{32},\ \varphi_{41} =^? \varphi_c^{41}\ \ \} \end{array}$$

Note that, since open processors do not introduce new flag variables (see Definition 35), besides the variables introduced by the plugging schemes in an OCTRSP-tree $\check{\mathcal{T}}$ the corresponding soundness and completeness equations may contain at most one additional flag variable (introduced by the open CTRS problem $\check{\tau}_I$ which labels the root of $\check{\mathcal{T}}$).

Theorem 43 (Open 2D DP Framework). *Let* $\check{\tau}_I = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ *be an OCTRS problem and* $\check{\mathcal{T}}_I$ *be an open CTRSP-tree for* $\check{\tau}_I$. *Then,*

1. *If all leaves in* $\check{\mathcal{T}}_I$ *are labeled with "*yes*" and there is a ground flag substitution* ς *that* unifies $\mathcal{E}_s(\check{\mathcal{T}}_I)$ *and* $\varsigma(\varphi) \neq \bullet$, *then* $\varsigma(\check{\tau}_I)$ *is* finite.
2. *If there is a leaf* n_∞ *in* $\check{\mathcal{T}}_I$ *with label "*no*" and there is a ground flag substitution* ς *that* unifies $\mathcal{E}_c^{\mathsf{n}_\infty}(\check{\mathcal{T}}_I)$ *and* $\varsigma(\varphi) \neq \bullet$, *then* $\varsigma(\check{\tau}_I)$ *is* infinite.

According to the possible instantiations of f in $\check{\tau}^H$ and $\check{\tau}^V$, Table 2 summarizes the conclusions of the analyses.

Termination prop.	$\check{\tau}^H$ / $\check{\tau}^V$	f **instantiation**	**Requirements on** $\mathcal{R}$	**F/I**
Termination	$\check{\tau}^H$	$f \mapsto \mathsf{a}$	preserves terminating substitutions	F
Nontermination	$\check{\tau}^H$	$f \mapsto \mathsf{m}$ or $f \mapsto \mathsf{a}$	None	I
V-Termination	$\check{\tau}^V$	$f \mapsto \mathsf{a}$	deterministic 3-CTRS	F
Non-V-Termination	$\check{\tau}^V$	$f \mapsto \mathsf{m}$ or $f \mapsto \mathsf{a}$	None	I
Op. termination	$\check{\tau}^H$	$f \mapsto \mathsf{m}$ or $f \mapsto \mathsf{a}$	deterministic 3-CTRS	F
	$\check{\tau}^V$	$f \mapsto \mathsf{m}$ or $f \mapsto \mathsf{a}$		F
Op. nontermination	$\check{\tau}^H$	$f \mapsto \mathsf{m}$ or $f \mapsto \mathsf{a}$	None	I
Op. nontermination	$\check{\tau}^V$	$f \mapsto \mathsf{m}$ or $f \mapsto \mathsf{a}$	None	I

Table 2: Open CTRS problems for proving termination properties

7. Processors for the (Open) 2D DP Framework

In this section we introduce several *processors* for their use in proofs of termination properties of CTRSs within the (Open) 2D DP Framework and illustrate their application with several examples. In particular:

1. the *SCC processor* P_{SCC} which permits the use of graph techniques to *decompose* CTRS problems (Section 7.1),
2. the *subterm processor*, $\mathsf{P}_{\triangleright}$, which removes pairs from $\mathcal{P}$ and $\mathcal{Q}$ without paying attention to the structure of rules in $\mathcal{R}$ (Section 7.2),
3. the *Removal Triple Processor* P_{RT} which uses well-founded relations to *simplify* termination problems (Section 7.3),
4. P_{IR} and P_{RIR}, which (re)move infeasible rules from the component $\mathcal{R}$ of an OCTRS problem (Section 7.4),
5. P_{NR}, which transforms the right-hand sides of the rules in $\mathcal{P}$ by using *narrowing* with $\mathcal{R}$; and P_{NQ}, which *narrows* with $\mathcal{Q}$ instead (Section 7.5).

First, we introduce the following processors that just *fix* the label of an open CTRS problem. Often, we will silently use them to introduce more flexibility in the use of processors as part of an OCTRSP-tree.

Definition 44. *Let* $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ *be an OCTRS problem. Processors* P_{arb} *and* P_{min} *are given by:*

$$\mathsf{P}_{arb}(\check{\tau}) = \{(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{a})\} \quad \textit{and} \quad \mathsf{P}_{min}(\check{\tau}) = \{(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m})\}$$

Theorem 45. P_{arb} *is sound and* a*-complete.* P_{min} *is* m*-sound and complete. Therefore,* $\Pi(\mathsf{P}_{arb}) = \{\langle f, \mathsf{a}\rangle\}$ *and* $\Pi(\mathsf{P}_{min}) = \{\langle \mathsf{m}, f\rangle\}$.

Processors P_{arb} and P_{min} are often silently used in practice to be able to use a-sound (m-complete) processors after using m-sound (a-complete) processors in a branch of an OCTRSP-tree. Also the *feasibility assumption* (Remark 13) is silently taken into account in the definition of our processors, regarding $\mathcal{P}$ and $\mathcal{Q}$ (which are always assumed $\mathcal{R}$-feasible). As an example, though, we make explicit reference to *feasible* rules in the development of P_{SCC} in the next section.

7.1. The SCC processor

In this section we provide a notion of graph that is able to represent all infinite (*minimal*) chains of (dependency) pairs as given in Definition 7. In the following, given CTRSs $\mathcal{P}$ and $\mathcal{R}$, by $\mathcal{P}_F$ we denote the subset of $\mathcal{R}$-*feasible* rules of $\mathcal{P}$ (see Section 4).

Definition 46 (CTRS Graph of Pairs). *Let $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ be CTRSs. The* CTRS-graph $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R})$ *has $\mathcal{P}_F$ as the set of nodes. There is an arc from a node α to a node α' if α,α' is a $(\mathcal{P},\mathcal{Q},\mathcal{R})$-O-chain for some substitution σ.*

By a graph for an OCTRS problem $\check{\tau} = (\mathcal{P},\mathcal{Q},\mathcal{R},\varphi)$ (written $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R},\varphi)$ or just $\mathsf{G}(\check{\tau})$) we mean the graph $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R})$. In general, $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R})$ is *not* computable: neither $\mathcal{R}$-feasibility of pairs nor reachability conditions $\sigma(v)(\to_{\mathcal{R}}^* \cup \overset{\Lambda}{\to}_{\mathcal{Q}})^*\sigma(u')$ between nodes $u \to v \Leftarrow c, u' \to v' \Leftarrow c'$ of the graph to connect them by means of an *arc* are decidable. Therefore, we need *approximations*.

Definition 47 (Approximation of $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R})$). *Let $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ be CTRSs. A graph $\mathsf{AG}(\mathcal{P},\mathcal{Q},\mathcal{R})$ is an (over)*approximation *of $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R})$ iff the nodes and arcs in $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R})$ are included in $\mathsf{AG}(\mathcal{P},\mathcal{Q},\mathcal{R})$.*

The following simpler graphs are also used in our development.[5]

Definition 48 (Simple CTRS Graph). *Let $\mathcal{Q}$ and $\mathcal{R}$ be CTRSs. The* simple *CTRS-graph $\mathsf{SG}(\mathcal{Q},\mathcal{R})$ has $\mathcal{Q}$ as the set of nodes. There is an arc from $u \to v \Leftarrow c \in \mathcal{Q}$ to $u' \to v' \Leftarrow c' \in \mathcal{Q}$ iff there is a substitution σ such that $\sigma(v) \to_{\mathcal{R}}^* \sigma(u')$ whenever $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ for all $s \to t \in c$.*

Note that, whenever $\mathcal{Q} = \emptyset$, we have $\mathsf{G}(\mathcal{P},\mathcal{Q},\mathcal{R}) = \mathsf{SG}(\mathcal{P},\mathcal{R})$. In the following, we identify a *cycle* in a (directed) graph G as a *sequence* $n_1,\ldots,n_m$ of nodes in G such that, for all i, $1 \le i < m$ there is an arc from n_i to n_{i+1}; there is also an arc from n_m to n_1. As usual in the literature of the field (see [12, Section 2.2], for instance), we write cycles as *sets* $\{n_1,\ldots,n_m\}$ whose presentation does not necessarily display the nodes in the appropriate sequence to define a cycle.

Given $\check{\tau} = (\mathcal{P},\mathcal{Q},\mathcal{R},\varphi)$, the SCC processor P_{SCC} decomposes $\check{\tau}$ according to an approximation $\mathsf{AG}(\check{\tau})$ of $\mathsf{G}(\check{\tau})$ into a possibly empty set $\mathsf{P}_{SCC}(\check{\tau})$ of CTRS problems, where the set of pairs $\mathcal{P}'$ in each $\check{\tau}' = (\mathcal{P}',\mathcal{Q}',\mathcal{R},\varphi) \in \mathsf{P}_{SCC}(\check{\tau})$ contains the pairs which are nodes of a maximal cycle (called a *strongly connected component* (SCC)) in $\mathsf{AG}(\check{\tau})$. Such SCCs are identified as subsets $\mathcal{P}' \subseteq \mathcal{P}_F$ of the nodes in the graph. Then, we take into account that only a (possibly empty) subset $\mathcal{Q}'$ of rules in $\mathcal{Q}_F$ will be *useful* to define the arcs between nodes in $\mathcal{P}'$.

[5] It is not difficult to see that our simple graphs are essentially the same as the usual dependency graphs for TRSs (see [12, Section 2.2] for instance) if we use conditional rewrite rules instead of unconditional rules.

Definition 49. *Let $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ be CTRSs. We let $\mathcal{Q}_{\mathcal{P},\mathcal{R}}$ be the smallest set of rules in $\mathcal{Q}_F$ such that for all $\alpha, \alpha' \in \mathcal{P}_F$, and paths from α to α' in $\mathsf{SG}(\mathcal{Q}_F \cup \{\alpha, \alpha'\}, \mathcal{R})$ the nodes (i.e., rules) in the path belonging to $\mathcal{Q}_F$ are in $\mathcal{Q}_{\mathcal{P},\mathcal{R}}$.*

Note that $\mathcal{Q}_{\mathcal{P},\mathcal{R}}$ can be easily overapproximated by using approximations of $\mathsf{SG}(\mathcal{Q}_F \cup \{\alpha, \alpha'\}, \mathcal{R})$ (see Section 7.1.1 below).

Definition 50. *Let $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ be CTRSs. The approximation $\overline{\mathcal{Q}}_{\mathcal{P},\mathcal{R}}$ of $\mathcal{Q}_{\mathcal{P},\mathcal{R}}$ is the union of all nodes (and therefore rules) from an approximation of $\mathcal{Q}_F$ occurring in the paths of the graphs $\mathsf{ASG}(\mathcal{Q}_F \cup \{\alpha, \alpha'\}, \mathcal{R})$ approximating $\mathsf{SG}(\mathcal{Q}_F \cup \{\alpha, \alpha'\}, \mathcal{R})$ that connect α and α' for all $\alpha, \alpha' \in \mathcal{P}$.*

Definition 51 (SCC processor). *Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ be an OCTRS problem, and $\mathsf{AG}(\check{\tau})$ be an approximation of $\mathsf{G}(\check{\tau})$. Then, P_{SCC} is given by*

$$\mathsf{P}_{SCC}(\check{\tau}) = \{(\mathcal{P}', \overline{\mathcal{Q}}_{\mathcal{P}',\mathcal{R}}, \mathcal{R}, \varphi) \mid \mathcal{P}' \text{ are the nodes of an SCC in } \mathsf{AG}(\check{\tau})\}$$

Theorem 52 (Soundness and completeness of P_{SCC}). *P_{SCC} is sound and complete. Thus, P_{SCC} follows $\Pi(\mathsf{P}_{SCC}) = \{\langle f, f\rangle\}$.*

With P_{SCC} we can *separately* work with the strongly connected components of $\mathsf{AG}(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$, disregarding other parts of the graph. In the next section we investigate a *computable* approximation that we use in the following.

7.1.1. Approximating the CTRS graph of pairs

In practice, rather than computing $\mathcal{Q}_F$ and $\mathcal{P}_F$ in approximations of $\mathsf{G}(\check{\tau})$ and $\mathsf{SG}(\check{\tau})$, we use the techniques referred in Section 4 and the references therein to obtain $\mathcal{Q}_I$ and $\mathcal{P}_I$ (which can be empty!) and let $\mathcal{P}_F = \mathcal{P} - \mathcal{P}_I$ and $\mathcal{Q}_F = \mathcal{Q} - \mathcal{Q}_I$ as the corresponding approximations. Thus, in the worst case, we will approximate $\mathcal{P}_F$ and $\mathcal{Q}_F$ as $\mathcal{P}$ and $\mathcal{Q}$, respectively.

With regard to the approximation of the arcs, following [11], we approximate $\mathsf{G}(\mathcal{P}, \mathcal{Q}, \mathcal{R})$ with $\textsc{tcap}_{\mathcal{R}}$ given as follows:

$$\textsc{tcap}_{\mathcal{R}}(x) = y \text{ if } x \text{ is a variable, and}$$

$$\textsc{tcap}_{\mathcal{R}}(f(t_1, \ldots, t_k)) = \begin{cases} f([t_1], \ldots, [t_k]) & \text{if } f([t_1], \ldots, [t_k]) \text{ does not unify} \\ & \text{with } \ell \text{ for any } \ell \to r \Leftarrow c \text{ in } \mathcal{R} \\ y & \text{otherwise} \end{cases}$$

where y is a variable that has not yet been used (i.e., it is new in every invocation of $\textsc{tcap}_{\mathcal{R}}$), and given a term s, $[s] = \textsc{tcap}_{\mathcal{R}}(s)$. We assume that ℓ shares no variable with $f([t_1], \ldots, [t_k])$. Note that, for all terms t, $\textsc{tcap}_{\mathcal{R}}(t)$ is a *linear term* and there is a substitution σ_t such that $t = \sigma_t(\textsc{tcap}_{\mathcal{R}}(t))$ and $\sigma_t(x) = x$ for all $x \notin \mathcal{V}ar(\textsc{tcap}_{\mathcal{R}}(t))$. With $\textsc{tcap}_{\mathcal{R}}$ we approximate reachability problems (with $\mathcal{R}$) by means of *unification*.

Proposition 53. *Let $\mathcal{R}$ be a CTRS, t, u be terms, and σ be a substitution. If $\sigma(t) \to_{\mathcal{R}}^* \sigma(u)$, then $\textsc{tcap}_{\mathcal{R}}(t)$ and u unify.*

According to Proposition 53, given terms $t, u \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and a substitution σ, the reachability of $\sigma(u)$ from $\sigma(t)$ by rewriting can be *approximated* as unification of $\mathrm{TCAP}_{\mathcal{R}}(t)$ and u. As a corollary, we have the following.

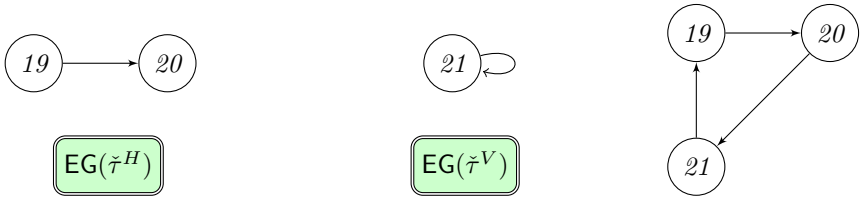
Corollary 54. *Let $\mathcal{Q}$ and $\mathcal{R}$ be CTRSs, u, v be terms, and σ be a substitution. If $\sigma(v)(\to_{\mathcal{R}} \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^*\sigma(u)$, then $\mathrm{TCAP}_{\mathcal{Q}\cup\mathcal{R}}(v)$ and u unify.*

This result immediately leads to an approximation of $\mathsf{G}(\mathcal{P}, \mathcal{Q}, \mathcal{R})$: just draw an arc from $u \to v \Leftarrow c$ to $u' \to v' \Leftarrow c'$ if $\mathrm{TCAP}_{\mathcal{Q}\cup\mathcal{R}}(v)$ and u' *unify*, provided that the conditional part c is $\mathcal{R}$-feasible. However, we investigate a more precise approximation based on the following.

Definition 55 (Estimated Simple CTRS Graph). *Let $\mathcal{Q}$ and $\mathcal{R}$ be CTRSs. The* estimated *simple CTRS-graph $\mathsf{ESG}(\mathcal{Q}, \mathcal{R})$ has an approximation of $\mathcal{Q}_F$ as the set of nodes. There is an arc from $u \to v \Leftarrow c \in \mathcal{Q}$ to $u' \to v' \Leftarrow c' \in \mathcal{Q}$ iff $\mathrm{TCAP}_{\mathcal{R}}(v)$ and u' unify.*

Definition 56 (Estimated CTRS Graph). *Let $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ be CTRSs. The* estimated *CTRS-graph $\mathsf{EG}(\mathcal{P}, \mathcal{Q}, \mathcal{R})$ has an approximation of $\mathcal{P}_F$ as the set of nodes. There is an arc from $\alpha : u \to v \Leftarrow c$ to $\alpha' : u' \to v' \Leftarrow c'$ iff there is a path from α to α' in $\mathsf{ESG}(\mathcal{Q} \cup \{\alpha, \alpha'\}, \mathcal{R})$.*

Example 57 (Estimated CTRS Graphs for $\mathcal{R}$ in Example 3). *Consider $\mathcal{R}$ in Example 3, with $\mathsf{DP}_H(\mathcal{R}) = \{(19), (20)\} = \{\mathsf{G}(\mathsf{a}) \to \mathsf{B}, \mathsf{B} \to \mathsf{F}(\mathsf{a})\}$ as in Example 6. The estimated graph $\mathsf{EG}(\check{\tau}^H)$ is shown below. Now, consider $\mathsf{DP}_V(\mathcal{R}) = \{(21)\} = \{\mathsf{F}(x) \to \mathsf{G}(x)\}$ also in Example 6 and $\mathsf{DP}_{VH}(\mathcal{R}) = \mathsf{DP}_H(\mathcal{R})$. Due to the path from (21) to itself in $\mathsf{ESG}(\mathsf{DP}_{VH}(\mathcal{R}) \cup \{(21)\}, \mathcal{R})$ displayed below, there is an arc from (21) to itself in $\mathsf{EG}(\check{\tau}^V)$:*



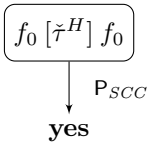
Estimated graphs for $\check{\tau}^H$ and $\check{\tau}^V$ $\mathsf{ESG}(\mathsf{DP}_{VH}(\mathcal{R}) \cup \{(21)\}, \mathcal{R})$

The following result is obvious from Proposition 53 and Corollary 54.

Theorem 58. *Let $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ be CTRSs. $\mathsf{EG}(\mathcal{P}, \mathcal{Q}, \mathcal{R})$ is an approximation of $\mathsf{G}(\mathcal{P}, \mathcal{Q}, \mathcal{R})$.*

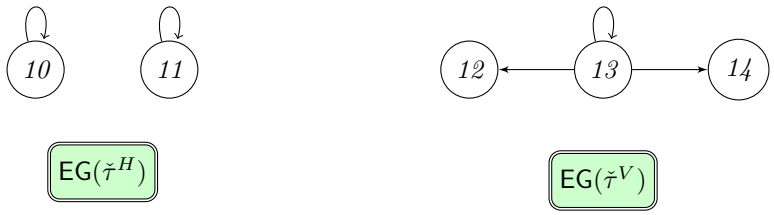
In the following, we use P_{SCC} with $\mathsf{EG}(\mathcal{P}, \mathcal{Q}, \mathcal{R})$.

Example 59 (Termination of $\mathcal{R}$ in Example 3). *For $\mathcal{R}$ in Example 3, with $\mathsf{EG}(\check{\tau}^H)$ in Example 57, we have $\mathsf{P}_{SCC}(\check{\tau}^H) = \emptyset$. Thus, the open CTRSP-tree $\check{\mathcal{T}}^H$ for $\check{\tau}^H$ is:*

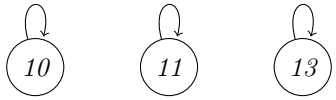


Note the renaming of flag variable f in the plugging scheme $\langle f, f\rangle$ for P_{SCC} (see Theorem 52) into f_0 to avoid 'collisions' with the flag variable used in $\check{\tau}^H$. Now, $\mathcal{E}_s(\check{\mathcal{T}}^H) = \{f = f_0\}$ is unified by the ground flag substitution ς_{a}. Thus, $\varsigma_{\mathsf{a}}(\check{\tau}^H) = (\mathsf{DP}_H(\mathcal{R}), \emptyset, \mathcal{R}, \mathsf{a})$ is proved finite. *Since $\mathcal{R}$ is a 2-CTRS (and preserves terminating substitutions), this proves $\mathcal{R}$* terminating *(see Table 2).*

Example 60. *For $\mathcal{R}$, $\mathsf{DP}_H(\mathcal{R})$, $\mathsf{DP}_V(\mathcal{R})$ in Example 2, with $\mathsf{DP}_{VH}(\mathcal{R}) = \mathsf{DP}_H(\mathcal{R})$ (see Example 5), we have $\check{\tau}^H = (\{(10), (11)\}, \emptyset, \mathcal{R}, f)$ and also $\check{\tau}^V = (\{(12), (13), (14)\}, \{(10), (11)\}, \mathcal{R}, f)$; $\mathsf{EG}(\check{\tau}^H)$ and $\mathsf{EG}(\check{\tau}^V)$ are:*

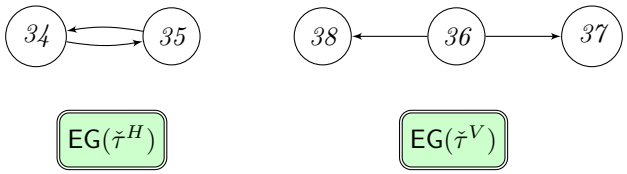


We have $\mathsf{P}_{SCC}(\check{\tau}^H) = \{\check{\tau}^H_{11}, \check{\tau}^H_{12}\}$, where $\check{\tau}^H_{11} = (\{(10)\}, \emptyset, \mathcal{R}, f)$ and $\check{\tau}^H_{12} = (\{(11)\}, \emptyset, \mathcal{R}, f)$. Also, $\mathsf{P}_{SCC}(\check{\tau}^V) = \{\check{\tau}^V_1\}$, where $\check{\tau}^V_1 = (\{(13)\}, \emptyset, \mathcal{R}, f)$. Note that the second component of $\check{\tau}^V_1$ (the $\mathcal{Q}$-component) becomes empty *due to the definition of P_{SCC}: since $\mathsf{ESG}(\{(10), (11)\} \cup \{(13)\}, \mathcal{R})$ is*

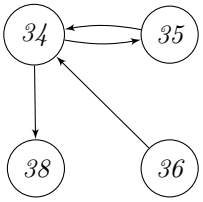


no path starting from (13) *involves nodes from $\{(10), (11)\}$ (Definition 50).*

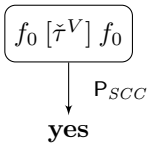
Example 61 (V-Termination of $\mathcal{R}$ in Example 30). *Consider $\mathcal{R}$, $\mathsf{DP}_H(\mathcal{R})$, $\mathsf{DP}_V(\mathcal{R})$, and $\mathsf{DP}_{VH}(\mathcal{R}) = \mathsf{DP}_H(\mathcal{R})$ in Example 30. We have:*



The arc from (36) to (38) is due to a path *in $\mathsf{ESG}(\mathsf{DP}_{VH}(\mathcal{R}) \cup \{(36), (38)\}, \mathcal{R})$ from (36) to (38):*



We have $\mathsf{P}_{SCC}(\check{\tau}^V) = \emptyset$ *and the corresponding OCTRSP-tree is:*



Now, $\mathcal{E}_s(\check{\mathcal{T}}^V) = \{f = f_0\}$ *is unified by the ground flag substitution* ς_{a}*. Thus,* $\varsigma_{\mathsf{a}}(\check{\tau}^V) = (\mathsf{DP}_V(\mathcal{R}), \mathsf{DP}_{VH}(\mathcal{R}), \mathcal{R}, \mathsf{a})$ *is* finite*. This proves* V-termination *of* $\mathcal{R}$*. Although* ς_{m} *is also a (ground) unifier for* $\mathcal{E}_s(\check{\mathcal{T}}^V)$ *no conclusion about any termination property of* $\mathcal{R}$ *in Example 30 follows in this way (see Table 2).*

Example 62 (Operational termination of $\mathcal{R}$ in Example 10). *For* $\mathcal{R}$ *in Example 10:*

$$\mathsf{DP}_H(\mathcal{R}):\ \mathsf{A} \ \to\ \mathsf{A} \Leftarrow \mathsf{b} \to x, \mathsf{c} \to x \tag{40}$$

$$\mathsf{DP}_V(\mathcal{R}):\ \mathsf{A} \ \to\ \mathsf{B} \tag{41}$$

$$\mathsf{A} \ \to\ \mathsf{C} \Leftarrow \mathsf{b} \to x \tag{42}$$

and $\mathsf{DP}_{VH}(\mathcal{R}) = \emptyset$*. Consider* $\check{\tau}^H = (\{(40)\}, \emptyset, \mathcal{R}, f)$*. Since* (40) *is* $\mathcal{R}$*-infeasible (use [21]) there is no node in* $\mathsf{G}(\check{\tau}^H)$ *and* $\mathsf{P}_{SCC}(\check{\tau}^H) = \emptyset$*.*

For $\check{\tau}^V = (\{(41), (42)\}, \emptyset, \mathcal{R}, f)$*, since* (41) *and* (42) *define no cycle, we have that* $\mathsf{P}_{SCC}(\check{\tau}^V) = \emptyset$*. The OCTRSP-trees*

prove operational termination *of* $\mathcal{R}$ *(see Table 2).*

Since P_{SCC} *coincides* with P_{Fin} on OCTRS problems $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ such that $\mathcal{P} = \emptyset$, P_{Fin} is not really necessary in an implementation of the (open) 2D DP Framework that already implements P_{SCC}. We used P_{Fin} to illustrate the definition and use of our framework. We will not use it anymore.

7.2. Subterm processor

In this section we generalize the *subterm processor* for TRSs [14]. Let $\mathcal{S}$ be a CTRS. The set of *root symbols* associated to $\mathcal{S}$ is:

$$\mathit{Root}(\mathcal{S}) \ = \ \{\mathit{root}(\ell) \mid \ell \to r \Leftarrow c \in \mathcal{S}\} \cup \{\mathit{root}(r) \mid \ell \to r \Leftarrow c \in \mathcal{S}, r \notin \mathcal{X}\}$$

Definition 63 (Simple projection). *Let $\mathcal{S}$ be a CTRS. A* simple projection *for $\mathcal{S}$ is a mapping $\pi : Root(\mathcal{S}) \to \mathbb{N}$ such that $\pi(f) \in \{1, \ldots, ar(f)\}$. The mapping that assigns a subterm $\pi(t) = t|_{\pi(f)}$ to each term t with $root(t) \in Root(\mathcal{S})$ is also denoted by π; we also let $\pi(x) = x$ if $x \in \mathcal{X}$.*

Given a simple projection π for a CTRS $\mathcal{S}$ and a CTRS $\mathcal{R}$, we let

$$\pi_{\mathcal{R}}(\mathcal{S}) = \{\pi(\ell) \to \pi(r) \mid \ell \to r \Leftarrow c \in \mathcal{S}\}.$$

The conditions of the rules are *dismissed* (but remember Remark 13). Given a CTRS problem $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$, the subterm processor *removes* from $\mathcal{P}$ and $\mathcal{Q}$ those rules $u \to v \Leftarrow c$ whose left-hand side u contains an immediate subterm $\pi(u)$ which is a *strict superterm* of an immediate subterm $\pi(v)$ of v (i.e., $\pi(u) \rhd \pi(v)$).

Definition 64 (Subterm processor). *Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ be an OCTRS problem, π be a simple projection for $\mathcal{P} \cup \mathcal{Q}$, and $\alpha : u \to v \Leftarrow c \in \mathcal{P} \cup \mathcal{Q}$. Then, $\mathsf{P}_{\rhd}$ is given by $\mathsf{P}_{\rhd}(\check{\tau}) = \{(\mathcal{P}[\emptyset]_\alpha, \mathcal{Q}[\emptyset]_\alpha, \mathcal{R}, \varphi)\}$, if $\pi_{\mathcal{R}}(\mathcal{P} \cup \mathcal{Q}) \subseteq \trianglerighteq$ and $\pi(u) \rhd \pi(v)$.*

We call a CTRS $\mathcal{R}$ *collapsing* if there is a rule $\ell \to r \Leftarrow c \in \mathcal{R}$ where r is a variable. Recall that $\mathcal{D}_{\mathcal{R}}$ is the set of defined symbols in $\mathcal{R}$.

Theorem 65 (Soundness and completeness of $\mathsf{P}_{\rhd}$). *$\mathsf{P}_{\rhd}$ is complete and $(\check{\tau}, \mathsf{m})$-sound for all $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi) \in \mathsf{Dom}^{\smile}(\mathsf{P}_{\rhd})$ such that $\mathcal{P} \cup \mathcal{Q}$ is not collapsing and $(Root(\mathcal{P}) \cup Root(\mathcal{Q})) \cap \mathcal{D}_{\mathcal{R}} = \emptyset$. Therefore, $\Pi(\mathsf{P}_{\rhd}) = \{\langle \mathsf{m}, f\rangle, \langle \bullet, f\rangle\}$.*

The plugging scheme $\langle \bullet, f\rangle$ should be used with OCTRS problems which do *not* make $\mathsf{P}_{\rhd}$ sound (for instance with $\mathcal{P} \cup \mathcal{Q}$ collapsing). Of course, such 'risky' uses of the processor could be *avoided* in an implementation of the Open 2D DP Framework. Then only $\langle \mathsf{m}, f\rangle$ would be used to build OCTRSP-trees.

Example 66 (Finiteness of $\check{\tau}^H$ for $\mathcal{R}$ in Example 2). *Consider the open CTRS problems in Example 60:*

$$\check{\tau}^H_{11} = (\{(10)\}, \emptyset, \mathcal{R}, f) \qquad \check{\tau}^H_{12} = (\{(11)\}, \emptyset, \mathcal{R}, f)$$

For $\check{\tau}^H_{11}$, where (10) is $\mathsf{LESS}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{LESS}(x, y)$, with the simple projection *π given by $\pi(\mathsf{LESS}) = 1$, we have:*

$$\pi(\mathsf{LESS}(\mathsf{s}(x), \mathsf{s}(y))) = \mathsf{s}(x) \rhd x = \pi(\mathsf{LESS}(x, y))$$

i.e., $\mathsf{P}_{\rhd}(\check{\tau}^H_{11}) = \{\check{\tau}^H_{21}\}$ with $\check{\tau}^H_{21} = (\emptyset, \emptyset, \mathcal{R}, f)$, which is trivially finite. *For $\check{\tau}^H_{12}$, where (11) is $\mathsf{MONUS}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{MONUS}(x, y)$, with $\pi(\mathsf{MONUS}) = 1$:*

$$\pi(\mathsf{MONUS}(\mathsf{s}(x), \mathsf{s}(y))) = \mathsf{cons}(x, xs) \rhd xs = \pi(\mathsf{MONUS}(x, y))$$

i.e., $\mathsf{P}_{\rhd}(\check{\tau}^H_{12}) = \{\check{\tau}^H_{22}\}$ with $\check{\tau}^H_{22} = (\emptyset, \emptyset, \mathcal{R}, f)$, also trivially finite. *The OCTRSP-tree is shown in Figure 5. We have (repeated equations are removed):*

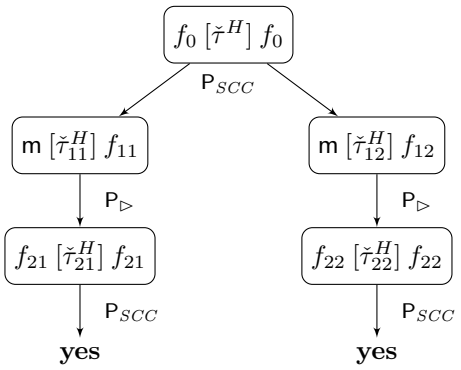


Figure 5: Finiteness of $\check{\tau}^H$ for $\mathcal{R}$ in Example 2 using the Open 2D DP Framework

$$\mathcal{E}_s(\check{\mathcal{T}}^H) \quad = \quad \{f = f_0, f = \mathsf{m}, f = f_{21}, f = f_{22}\}$$

which is unified by the ground substitution ς_{m} only. *We can conclude finiteness of* $\varsigma_{\mathsf{m}}(\check{\tau}^H)$ *now. However, we cannot conclude termination (according to Table 1) because finiteness of* $\varsigma_{\mathsf{a}}(\check{\tau}^H)$ *is required for that. In Example 73, though, we prove* $\mathcal{R}$ *operationally terminating and, consequently, terminating as well.*

Note that $\mathsf{P}_{\triangleright}$ cannot *be used to remove (13), i.e.,*

$$\mathsf{QUOTREM}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{QUOTREM}(\mathsf{monus}(x, y), \mathsf{s}(y))$$

from $\check{\tau}^V_1 = (\{(13)\}, \emptyset, \mathcal{R}, f)$ *because no simple projection on its left-hand side yields a strict superterm of the projection of the right-hand side.*

Note that $\mathsf{P}_{\triangleright}$ may *fail* to be sound when applied to an *arbitrary* $\check{\tau} \in \mathsf{Dom}^{\check{}}(\mathsf{P}_{\triangleright})$.

Example 67. *Let* $\mathcal{P} = \{\mathsf{F}(\mathsf{g}(\mathsf{a})) \to \mathsf{F}(\mathsf{a})\}$ *and* $\mathcal{R} = \{\mathsf{a} \to \mathsf{g}(\mathsf{a})\}$. *Then, for* $\check{\tau} = (\mathcal{P}, \emptyset, \mathcal{R}, \mathsf{a}) \in \mathsf{Dom}^{\check{}}(\mathsf{P}_{\triangleright})$, *there is an infinite* $(\mathcal{P}, \emptyset, \mathcal{R})$*-O-chain:*

$$\underline{\mathsf{F}(\mathsf{g}(\mathsf{a}))} \to_{\mathcal{P}} \mathsf{F}(\underline{\mathsf{a}}) \to_{\mathcal{R}} \underline{\mathsf{F}(\mathsf{g}(\mathsf{a}))} \to_{\mathcal{P}} \cdots \tag{43}$$

However, since $\pi(\mathsf{F}(\mathsf{g}(\mathsf{a}))) = \mathsf{g}(\mathsf{a}) \triangleright \mathsf{a} = \pi(\mathsf{F}(\mathsf{a}))$ *for* $\pi(\mathsf{F}) = 1$, *we have* $\mathsf{P}_{\triangleright}(\check{\tau}) = \{(\emptyset, \emptyset, \mathcal{R}, \mathsf{a})\}$. *We would wrongly conclude finiteness of* $\check{\tau}$. *Note that* (43) *is* not *minimal because* $\mathsf{F}(\mathsf{a})$ *is* not *(operationally) terminating.*

7.3. Use of well-founded relations

The absence of infinite $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chains can be ensured by finding appropriate relations that are compatible with the rules in $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$. In the following we provide precise definitions to achieve this.

Definition 68 (Removal triple). *A* removal triple $(\gtrsim, \succeq, \sqsupset)$ *consists of relations* $\gtrsim, \succeq, \sqsupset$ *on terms such that (i)* $\sqsupset$ *is well-founded, (ii)* $\gtrsim \circ \sqsupset \subseteq \sqsupset$, *and (iii)* $\succeq \circ \sqsupset \subseteq \sqsupset$.

Definition 69. *Let $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ be CTRSs. A removal triple $(\gtrsim, \succeq, \sqsupset)$ is* compatible *with $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$, if for all terms s, t,*

1. *if $s \to_{\mathcal{R}} t$, then $s \gtrsim t$ and*
2. *if $s \xrightarrow{\Lambda}_{\mathcal{P}\cup\mathcal{Q},\mathcal{R}} t$, then $s \bowtie t$ holds for some $\bowtie \in \{\gtrsim, \succeq, \sqsupset\}$.*

Removal triples are used to simplify CTRS problems $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ by removing rules from $\mathcal{P}$ and $\mathcal{Q}$. Here, $\gtrsim$ simulates $\to_{\mathcal{R}}$-steps within a connection $\sigma(v)(\to_{\mathcal{R}}^* \cup \xrightarrow{\Lambda}_{\mathcal{Q}})^*\sigma(u')$. Similarly, $\succeq$ simulates $\xrightarrow{\Lambda}_{\mathcal{Q}}$-steps.

Definition 70 (Removal triple processor). *Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ be an OC-TRS problem and $(\gtrsim, \succeq, \sqsupset)$ be a removal triple which is compatible with $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$. Let $\alpha : u \to v \Leftarrow c \in \mathcal{P} \cup \mathcal{Q}$. Then, P_{RT} is given by $\mathsf{P}_{RT}(\check{\tau}) = \{(\mathcal{P}[\emptyset]_\alpha, \mathcal{Q}[\emptyset]_\alpha, \mathcal{R}, \varphi)\}$ iff $\sigma(u) \sqsupset \sigma(v)$ whenever $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ for all substitutions σ and all $s \to t \in c$.*

Theorem 71 (Soundness and completeness of P_{RT}). *P_{RT} is sound and complete. Therefore, $\Pi(\mathsf{P}_{RT}) = \{\langle f, f\rangle\}$.*

Example 72. *With regard to $\mathcal{R}$ in Example 2, for $\check{\tau}_1^V = (\{(13)\}, \emptyset, \mathcal{R}, f)$ in Example 60, we apply P_{RT} using the removal triple $(\geq, \geq, >)$ induced by the polynomial interpretation over the naturals*

$$\begin{array}{rclrclrcl}
[\mathsf{false}] &=& 0 & [\mathsf{true}] &=& 0 & [\mathsf{0}] &=& 0\\
[\mathsf{s}](x) &=& x+1 & [\mathsf{less}](x) &=& 0 & [\mathsf{monus}](x,y) &=& x\\
[\mathsf{pair}](x,y) &=& 0 & [\mathsf{quotrem}](x,y) &=& 0 & [\mathsf{QUOTREM}](x,y) &=& x
\end{array}$$

where, as usual, $s \geq t$ if $[s] \geq [t]$ and $s > t$ if $[s] > [t]$. We have:

$$\begin{array}{rclcl}
[\mathsf{less}(x,\mathsf{0})] &=& 0 &\geq& 0 = [\mathsf{false}]\\
[\mathsf{less}(\mathsf{0},\mathsf{s}(x))] &=& 0 &\geq& 0 = [\mathsf{true}]\\
[\mathsf{less}(\mathsf{s}(x),\mathsf{s}(y))] &=& 0 &\geq& 0 = [\mathsf{less}(x,y)]\\
[\mathsf{monus}(\mathsf{0},\mathsf{s}(y))] &=& 0 &\geq& 0 = [\mathsf{0}]\\
[\mathsf{monus}(x,\mathsf{0})] &=& x &\geq& x = [x]\\
[\mathsf{monus}(\mathsf{s}(x),\mathsf{s}(y))] &=& x+1 &\geq& x = [\mathsf{monus}(x,y)]\\
[\mathsf{quotrem}(\mathsf{0},\mathsf{s}(y))] &=& 0 &\geq& 0 = [\mathsf{pair}(\mathsf{0},\mathsf{0})]\\
[\mathsf{quotrem}(\mathsf{s}(x),\mathsf{s}(y))] &=& 0 &\geq& 0 = [\mathsf{pair}(\mathsf{0},\mathsf{s}(x))]\\
[\mathsf{quotrem}(\mathsf{s}(x),\mathsf{s}(y))] &=& 0 &\geq& 0 = [\mathsf{pair}(\mathsf{s}(q),r)]
\end{array}$$

$$[\mathsf{QUOTREM}(\mathsf{s}(x),\mathsf{s}(y))] = x+1 > x = [\mathsf{QUOTREM}(\mathsf{monus}(x,y),\mathsf{s}(y))]$$

Since $\geq$ is monotonic, stable, reflexive and transitive, the inequalities prove $\to_{\mathcal{R}}^* \subseteq \geq$ (we do not need to pay attention to the conditional part of the rules). The strict inequality shows that $u_{13} > v_{13}$. Since $>$ is stable, this proves $\sigma(u_{13}) > \sigma(v_{13})$ for all substitutions σ (disregarding the condition in (13)).

Example 73 (Operational termination of $\mathcal{R}$ in Example 2). *For $\mathcal{R}$ in Example 2, Example 66 proves $\varsigma_{\mathsf{m}}(\check{\tau}^H)$ finite (Figure 5). Figure 6 shows $\check{\mathcal{T}}^V$ for*

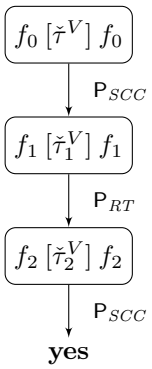


Figure 6: Finiteness of $\check{\tau}^V$ for $\mathcal{R}$ in Example 2 in the Open 2D DP Framework

$\check{\tau}^V$ in Example 60, according to the application of different processors as discussed in Examples 66 and 72. Since $\mathcal{E}_s(\check{\mathcal{T}}^V) = \{f = f_0, f = f_1, f = f_2\}$ is unified by ς_{a} and also by ς_{m}, $\varsigma_{\mathsf{m}}(\check{\tau}^V)$ is also finite, and then $\mathcal{R}$ is operationally terminating.

In Example 72 no real use of the information in the conditional part of the rules is made. In other cases, this is crucial for a successful application of P_{RT}.

Example 74. *Consider the following CTRS $\mathcal{R}$:*

$$\mathsf{g}(\mathsf{a}) \to \mathsf{f}(\mathsf{b}) \quad (44) \qquad\qquad \mathsf{f}(x) \to \mathsf{g}(y) \Leftarrow \mathsf{h}(x) \to y \quad (46)$$
$$\mathsf{h}(\mathsf{a}) \to \mathsf{b} \quad (45)$$

We have:

$$\mathsf{DP}_H(\mathcal{R}): \quad \mathsf{G}(\mathsf{a}) \to \mathsf{F}(\mathsf{b}) \quad (47)$$
$$\mathsf{F}(x) \to \mathsf{G}(y) \Leftarrow \mathsf{h}(x) \to y \quad (48)$$
$$\mathsf{DP}_V(\mathcal{R}): \quad \mathsf{F}(x) \to \mathsf{H}(x) \quad (49)$$

and $\mathsf{DP}_{VH}(\mathcal{R}) = \emptyset$. Since $\mathsf{EG}(\check{\tau}^V)$ has no cycle, we only need to consider $\check{\tau}^H$ which can be proved finite using P_{RT}. Consider the removal triple $(\geq, \geq, >)$ induced by the following polynomial interpretation:

$$[\mathsf{a}] = 2 \quad [\mathsf{f}](x) = 2x + 2 \quad [\mathsf{g}](x) = x + 2 \quad [\mathsf{h}](x) = 2x$$
$$[\mathsf{b}] = 0 \quad [\mathsf{F}](x) = 2x + 2 \quad [\mathsf{G}](x) = x + 1 \quad [\mathsf{H}](x) = 0$$

In order to apply the processor, we have to prove the following (Definition 69):

1. *for all terms s, t, if $s \to_{\mathcal{R}} t$, then $s \geq t$. This can be proved if the following sentence $(\forall s, t)\ s \to t \Rightarrow s \geq t$ (where, by abuse, $\geq$ is considered a new* predicate symbol*) is satisfied by the structure $\mathcal{A}$ which interprets the symbols in the signature as above and $\to^*$ and $\to$ are interpreted as $\geq$, see [20, Section 11.1].*

2. *for all terms* s, t*, if* $s \xrightarrow{\Lambda}_{\mathcal{P}\cup\mathcal{Q},\mathcal{R}} t$*, then* $s \bowtie t$ *holds for some* $\bowtie \in \{\geq, >\}$*. In this particular case, we can avoid this and go directly to*
3. *prove 'decreasingness' of the two dependency pairs in* $\check{\tau}^H$ *(Definition 70) by proving the following sentences*

$$\mathsf{G}(\mathsf{a}) > \mathsf{F}(\mathsf{b}) \tag{50}$$

$$(\forall x, y)\ \mathsf{h}(x) \to^* y \Rightarrow \mathsf{F}(x) > \mathsf{G}(y) \tag{51}$$

both satisfiable by $\mathcal{A}$*. Actually, since* $[\mathsf{G}(\mathsf{a})] = 2 + 1 = 3$ *and* $[\mathsf{F}(\mathsf{b})] = 2 \cdot 0 + 2 = 2$*, this corresponds to checking the following*

$$3 > 2 \tag{52}$$

$$(\forall x, y \in \mathbb{N})\ 2x \geq y \Rightarrow 2x + 2 > y + 1 \tag{53}$$

Note that this time we use the fact that $2x \geq y$ *holds, in the antecedent of (53), to* conclude *that* $2x + 2 > y + 1$ *holds.*

Therefore, we can remove both pairs in $\check{\tau}^H$ *to obtain a trivial CTRS problem.*

7.4. Processors exploiting infeasibility of rules

One could think of rules in $\mathcal{R}_I$ as somehow *useless* and therefore just *removable* from any CTRS problem $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$. However, as illustrated in Example 12, completeness of processors that remove rules from $\mathcal{R}$ in the returned CTRS problems *cannot* be guaranteed for *minimal* CTRS problems due to the lack of preservation of operational termination of $\mathcal{R}$ under addition of rules, even if such rules are infeasible. For this reason, we introduce two processors to deal with infeasible rules: the first one, P_{IR}, rather than removing infeasible rules from $\mathcal{R}$, *moves* them from $\mathcal{R}_F$ to $\mathcal{R}_I$.

Definition 75 (Infeasible rules processor). *Let* $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ *be an OCTRS problem and* $\alpha : \ell \to r \Leftarrow c \in \mathcal{R}_F$*.* P_{IR} *is given by*

$$\mathsf{P}_{IR}(\mathcal{P}, \mathcal{Q}, \mathcal{R}_F \uplus \mathcal{R}_I, \varphi) = \{(\mathcal{P}, \mathcal{Q}, \mathcal{R}_F[\emptyset]_\alpha \uplus (\mathcal{R}_I \cup \{\alpha\}), \varphi)\}$$

iff α *is* $\mathcal{R}_F$*-infeasible.*

Although P_{IR} does *not* formally change the input CTRS problem, it makes the distinction between infeasible and 'other' rules from $\mathcal{R}$ explicit. Other processors (e.g., P_{SCC}) may benefit from such a distinction introduced by P_{IR}.

Theorem 76. P_{IR} *is sound and complete. Therefore,* $\Pi(\mathsf{P}_{IR}) = \{\langle f, f\rangle\}$*.*

In some cases, definitely dropping infeasible rules from $\mathcal{R}$ can be useful. Thus, we provide the following.

Definition 77 (Removing infeasible rules processor). *Let* $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ *be an OCTRS problem and* $\varphi \in \mathcal{T}(\mathsf{F}, \mathsf{V})$*.* P_{RIR} *is given by*

$$\mathsf{P}_{RIR}(\mathcal{P}, \mathcal{Q}, \mathcal{R}_F \uplus \mathcal{R}_I, \varphi) = \{(\mathcal{P}, \mathcal{Q}, \mathcal{R}_F, \varphi)\}$$

iff $\mathcal{R}_I \neq \emptyset$*.*

Theorem 78. P_{RIR} *is sound and* a*-complete. Therefore,* $\Pi(\mathsf{P}_{RIR}) = \{\langle f, \mathsf{a}\rangle\}$*.*

7.5. Narrowing the right-hand sides of rules

In the DP approach for TRSs $\mathcal{R}$, connections between dependency pairs $\alpha : u \to v$ and $\alpha' : u' \to v'$ by rewriting, i.e., the existence of a substitution σ such that $\sigma(v) \to^*_{\mathcal{R}} \sigma(u')$ has been investigated by using narrowing [1, 12]. If there is a connection between α and α' involving rewritings with $\mathcal{R}$, then after narrowing v into *all* its possible narrowings $v_1, \ldots, v_n$, the connection will be exhibited by some of the v_i in a more specific way. The good point is that other narrowings will eventually become *unable* to establish any connection. This may lead to their *removal* by other processors (e.g., P_{SCC}), thus leading to a more precise analysis.

The connection between rules $\alpha : u \to v \Leftarrow c$ and $\alpha' : u' \to v' \Leftarrow c' \in \mathcal{P}$ within a $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain is a reachability problem $\sigma(v)(\to_{\mathcal{R}} \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^*\sigma(u')$, which can also be investigated using narrowing. In our setting, we introduce the following notion of narrowing for CTRSs.

Definition 79 (Narrowing with CTRSs). *Let $\mathcal{R}$ be a CTRS. A term s* narrows *to a term t (written $s \rightsquigarrow_{\mathcal{R},\theta,p} t$ or just $s \rightsquigarrow_{\mathcal{R},\theta} t$ or even $s \rightsquigarrow t$), iff there are a nonvariable position $p \in \mathcal{P}os_{\mathcal{F}}(s)$, a* renamed *rule $\ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n$ in $\mathcal{R}$, substitutions $\theta_0, \ldots, \theta_n, \tau_1, \ldots, \tau_n$, and terms $t'_1, \ldots, t'_n$ such that:*

1. *$s|_p =^?_{\theta_0} \ell$,*
2. *for all i, $1 \le i \le n$, $\eta_{i-1}(s_i) \rightsquigarrow^*_{\mathcal{R},\theta_i} t'_i$ and $t'_i =^?_{\tau_i} \theta_i(\eta_{i-1}(t_i))$, where $\eta_0 = \theta_0$ and for all $i > 0$, $\eta_i = \tau_i \circ \theta_i \circ \eta_{i-1}$, and*
3. *$t = \theta(s[r]_p)$, where $\theta = \eta_n$.*

Here (and in the following) we write $u \rightsquigarrow^*_{\mathcal{R},\beta} v$ for terms u, v and substitution β iff there are terms $u_1, \ldots, u_{m+1}$ and substitutions $\beta_1, \ldots, \beta_m$ for some $m \ge 0$ such that

$$u = u_1 \rightsquigarrow_{\mathcal{R},\beta_1} u_2 \rightsquigarrow_{\mathcal{R},\beta_2} \cdots \rightsquigarrow_{\mathcal{R},\beta_m} u_{m+1} = v$$

and $\beta = \beta_{m-1} \circ \cdots \circ \beta_1$ (or $\beta = \varepsilon$ if $m = 0$).

In contrast to *unconditional* narrowing, *several* narrowing steps can be issued from a given term, position, and rule. This is due to different ways to satisfy the reachability tests in the conditional part of the rule. Also, the substitution θ in $s \rightsquigarrow_{\mathcal{R},\theta,p} t$ is a unifier of $s|_p$ and ℓ but it does *not* need to be an *mgu*. In fact, there can be *delayed* instantiations of variables in $s|_p$ which become instantiated during the reachability tests for the conditional part of the rules.

Example 80. *Consider the following CTRS $\mathcal{R}$:*

$$\begin{aligned}
\mathsf{c} &\to \mathsf{h}(\mathsf{a}) && (54) & \qquad \mathsf{f}(\mathsf{h}(x)) \to y &\Leftarrow \mathsf{g}(\mathsf{h}(x)) \to y && (56)\\
\mathsf{g}(\mathsf{h}(\mathsf{a})) &\to \mathsf{b} && (55) & \qquad \mathsf{g}(\mathsf{b}) \to \mathsf{g}(\mathsf{f}(x)) &\Leftarrow \mathsf{c} \to x && (57)
\end{aligned}$$

Consider $s = \mathsf{f}(x)$ and a renaming of rule (56):

$$\mathsf{f}(\mathsf{h}(x')) \to y' \Leftarrow \mathsf{g}(\mathsf{h}(x')) \to y' \qquad (58)$$

We narrow s using (58). Note that s and $\ell' = \mathsf{f}(\mathsf{h}(x'))$ unify with mgu *$\theta_0 = \{x \mapsto \mathsf{h}(x')\}$. Now, the instance $\theta_0(s') = \mathsf{g}(\mathsf{h}(x'))$ of the left-hand side $s' = \mathsf{g}(\mathsf{h}(x'))$ of the condition in (58) can be narrowed in two ways:*

1. *No narrowing step is issued on* $\theta_0(s')$ *(i.e.,* $\theta_1 = \varepsilon$*, the empty substitution). Since* y' *and* s' *unify with* mgu $\tau_1 = \{y' \mapsto \mathsf{g}(\mathsf{h}(x'))\}$ *we finally obtain:*

$$\mathsf{f}(x) \leadsto_{\mathcal{R},\theta} \mathsf{g}(\mathsf{h}(x'))$$

where $\theta = \eta_1 = \tau_1 \circ \theta_1 \circ \eta_0 = \tau_1 \circ \theta_1 \circ \theta_0 = \{x \mapsto \mathsf{h}(x'), y' \mapsto \mathsf{g}(\mathsf{h}(x'))\}$.
2. *An* unconditional *narrowing step is issued on* $\theta_0(s')$ *using rule (55), i.e.:*

$$\mathsf{g}(\mathsf{h}(x')) \leadsto_{\mathcal{R},\theta_1} \mathsf{b}$$

with $\theta_1' = \{x' \mapsto \mathsf{a}\}$*. Since* y' *and* b *unify with* mgu $\tau_1' = \{y' \mapsto \mathsf{b}\}$ *we get:*

$$\mathsf{f}(x) \leadsto_{\mathcal{R},\theta'} \mathsf{b}$$

where $\theta' = \eta_1' = \tau_1' \circ \theta_1' \circ \eta_0' = \tau_1' \circ \theta_1' \circ \theta_0 = \{x \mapsto \mathsf{h}(\mathsf{a}), x' \mapsto \mathsf{a}, y' \mapsto \mathsf{b}\}$.

No further narrowing step is possible on s*.*

Furthermore, given a term, a position within this term, and a conditional rule the set of one-step narrowings using this position and rule can be *infinite*.

Example 81. *Consider the following (operationally terminating) 1-CTRS* $\mathcal{R}$*:*

$$\begin{aligned} \mathsf{f}(\mathsf{f}(x)) &\to \mathsf{g}(\mathsf{f}(x)) && (59)\\ \mathsf{g}(x) &\to \mathsf{a} && (60)\\ \mathsf{k}(x) &\to x \Leftarrow x \to \mathsf{a} && (61) \end{aligned}$$

Note that $\mathsf{f}(y)$ *can be narrowed as follows:*

$$\underline{\mathsf{f}(y)} \leadsto_{(59),\{y\mapsto\mathsf{f}(x')\},\Lambda} \mathsf{g}(\underline{\mathsf{f}(x')}) \leadsto_{(59),\{x'\mapsto\mathsf{f}(x'')\},1} \mathsf{g}(\mathsf{g}(\underline{\mathsf{f}(x'')})) \leadsto \cdots \qquad (62)$$

Thus, a term like $\mathsf{k}(\mathsf{f}(y))$ *has infinitely many one-step narrowings with (61) corresponding to extracting an initial finite sequence from (62) and then applying (60) at the root of the last term* $\mathsf{g}(\cdots\mathsf{g}(\mathsf{f}(x^{(n)}))\cdots)$ *in the sequence:*

$$\begin{array}{lll} \underline{\mathsf{k}(\mathsf{f}(y))} & \leadsto_{(61),\{y\mapsto\mathsf{f}(x')\},\Lambda} & \mathsf{f}(\mathsf{f}(x'))\\ \underline{\mathsf{k}(\mathsf{f}(y))} & \leadsto_{(61),\{y\mapsto\mathsf{f}(\mathsf{f}(x''))\},\Lambda} & \mathsf{f}(\mathsf{f}(\mathsf{f}(x'')))\\ & \vdots & \end{array}$$

Actually, this may also happen if one-step rewritings (rather than narrowings) are considered in proper (finite) 3-CTRSs: there are terms with infinitely many one-step reducts.

Example 82. *Consider the following (nonterminating) 3-CTRS* $\mathcal{R}$*:*

$$\begin{aligned} \mathsf{a} &\to \mathsf{c}(\mathsf{a}) && (63)\\ \mathsf{b} &\to x \Leftarrow \mathsf{a} \to x && (64) \end{aligned}$$

Note that $\mathsf{b} \to_{\mathcal{R}} \mathsf{c}^n(\mathsf{a})$ *for all* $n \geq 0$*.*

Given a CTRS $\mathcal{S}$, a non-variable term t is a *narrowing redex* (or a *narrex*, for short) of a rule $\ell \to r \Leftarrow c \in \mathcal{S}$ if t and ℓ unify with *mgu* θ (we assume $\mathcal{V}ar(t) \cap \mathcal{V}ar(\ell) = \emptyset$), and $\theta(c)$ is $\mathcal{S}$-feasible. We let $NRules(\mathcal{S}, s)$ be the set of rules $\alpha : \ell \to r \Leftarrow c \in \mathcal{S}$ such that a nonvariable subterm t of s is a *narrex* of α. Then, $N_1(\mathcal{S}, s)$ represents the set of one-step $\mathcal{S}$-narrowings issued from s:

$$N_1(\mathcal{S}, s) = \{(t, \theta{\downarrow_{\mathcal{V}ar(s)}}) \mid s \rightsquigarrow_{\ell \to r \Leftarrow c, \theta} t, \ell \to r \Leftarrow c \in NRules(\mathcal{S}, s)\} \tag{65}$$

where $\theta{\downarrow_{\mathcal{V}ar(s)}}$ is a substitution defined by $\theta{\downarrow_{\mathcal{V}ar(s)}}(x) = \theta(x)$ if $x \in \mathcal{V}ar(s)$ and $\theta{\downarrow_{\mathcal{V}ar(s)}}(x) = x$ otherwise.

Example 83. *For $\mathcal{R}$ in Example 80, $NRules(\mathcal{R}, \mathsf{f}(x))$ consists of rule (56) only, and $N_1(\mathcal{R}, \mathsf{f}(x)) = \{(\mathsf{g}(\mathsf{h}(x')), \{x \mapsto \mathsf{h}(x')\})\ ,\ (\mathsf{b}, \{x \mapsto \mathsf{h}(\mathsf{a})\})\}$.*

For $\mathcal{R}$ in Example 82, $NRules(\mathcal{R}, \mathsf{b})$ consists of rule (64) only, but $N_1(\mathcal{R}, \mathsf{b}) = \{(\mathsf{a}, \varepsilon)\ ,\ (\mathsf{c}(\mathsf{a}), \varepsilon), \ldots\}$ is infinite.

If $\alpha : u \to v \Leftarrow c$ is a conditional rule,

$$\overline{\mathcal{N}}(\mathcal{S}, \alpha) = \{\theta(u) \to w \Leftarrow \theta(c) \mid (w, \theta) \in N_1(\mathcal{S}, v)\} \tag{66}$$

We call $\overline{\mathcal{N}}(\mathcal{S}, \alpha)$ the set of *one-step narrowings* of α with $\mathcal{S}$. Note that (66) can be an *infinite* set (if $N_1(\mathcal{S}, v)$ is infinite).

Example 84. *For $\mathcal{R}$ in Example 80, we have:*

$$\mathsf{DP}_H(\mathcal{R}): \qquad \mathsf{G}(\mathsf{b}) \to \mathsf{G}(\mathsf{f}(x)) \Leftarrow \mathsf{c} \to x \tag{67}$$

$$\mathsf{G}(\mathsf{b}) \to \mathsf{F}(x) \Leftarrow \mathsf{c} \to x \tag{68}$$

$\overline{\mathcal{N}}(\mathcal{R}, (67))$ consists of the following rules (see Example 83):

$$\mathsf{G}(\mathsf{b}) \to \mathsf{G}(\mathsf{g}(\mathsf{h}(x'))) \Leftarrow \mathsf{c} \to \mathsf{h}(x') \tag{69}$$

$$\mathsf{G}(\mathsf{b}) \to \mathsf{G}(\mathsf{b}) \Leftarrow \mathsf{c} \to \mathsf{h}(\mathsf{a}) \tag{70}$$

We define the following processor.

Definition 85 (Narrowing Processor with $\mathcal{R}$). *Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ be an OCTRS problem, $\alpha : u \to v \Leftarrow c \in \mathcal{P}$, and $\mathcal{N}$ be a finite subset of $\overline{\mathcal{N}}(\mathcal{R}, \alpha)$. Then,*

$$\mathsf{P}_{NR}(\check{\tau}) = \{(\mathcal{P}[\mathcal{N}]_\alpha, \mathcal{Q}, \mathcal{R}, \varphi)\}$$

Clearly, $\overline{\mathcal{N}}(\mathcal{R}, \alpha)$ is finite iff $N_1(\mathcal{R}, v)$ is finite; this guarantees that every subset $\mathcal{N}$ of $\overline{\mathcal{N}}(\mathcal{R}, \alpha)$ in Definition 85 can be used to apply P_{NR}. Our next result provides a sufficient condition to guarantee finiteness of $N_1(\mathcal{R}, v)$. First, we need the following definition, which uses $RULES_\Lambda$ as defined in Section 3.3.

Definition 86. [25, Definition 11] *Let $\mathcal{R}$ be a CTRS and t be a term. Let $RULES(\mathcal{R}, t) = \bigcup_{s \in \mathcal{D}_\trianglerighteq(\mathcal{R}, t)} RULES_\Lambda(\mathcal{R}, s)$. The set of* usable rules *of $\mathcal{R}$ for t is*

$$\mathcal{U}(\mathcal{R}, t) = RULES(\mathcal{R}, t) \cup \bigcup_{\ell \to r \Leftarrow c \in RULES(\mathcal{R}, t)} \Big(\mathcal{U}(\mathcal{R}^\bullet, r) \cup \bigcup_{s_i \to t_i \in c} \mathcal{U}(\mathcal{R}^\bullet, s_i)\Big)$$

where $\mathcal{R}^\bullet = \mathcal{R} - RULES(\mathcal{R}, t)$.

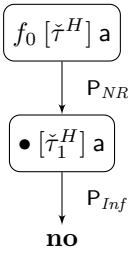


Figure 7: Nontermination of $\mathcal{R}$ in Example 80 in the Open 2D DP Framework

The following simple result summarizes our previous discussion about finiteness of $N_1(\mathcal{R}, t)$.

Proposition 87. *Let $\mathcal{R}$ be a finite CTRS and t be a term. Then, $N_1(\mathcal{R}, t)$ is finite if one of the following conditions hold:*

1. *$NRules(\mathcal{R}, t)$ is a TRS,*
2. *t is ground and $\mathcal{R}$ is a 2-CTRS,*
3. *t is ground and $\mathcal{U}(\mathcal{R}, t)$ is a terminating and deterministic 3-CTRS.*

If $\mathcal{N}$ in Definition 85 is such that $\mathcal{N} = \overline{\mathcal{N}}(\mathcal{R}, \alpha)$, then all possible rewriting steps issued by instantiations of rule α and rewriting steps on the corresponding instance of v are covered. This is crucial to prove *soundness* of P_{NR}. In contrast, *completeness* is not affected if some narrowings from $\overline{\mathcal{N}}(\mathcal{R}, \alpha)$ are *missing* in $\mathcal{N}$.

Theorem 88 (Soundness and completeness of P_{NR}). *Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ and $\alpha : u \to v \Leftarrow c \in \mathcal{P}$ be as in Definition 85. P_{NR} is a-complete. P_{NR} is sound for $\check{\tau}$ if (i) $\mathcal{N} = \overline{\mathcal{N}}(\mathcal{R}, \alpha)$, (ii) for all $u' \to v' \Leftarrow c' \in \mathcal{P} \cup \mathcal{Q}$ (with renamed variables), either v and u' do not unify or v and u' unify with* mgu *θ and $\theta(c)$ is $\mathcal{R}$-infeasible, and (iii) either v is ground and $\mathcal{R}$ is a deterministic 3-CTRS or $NRules(\mathcal{R}, v)$ is a TRS and v is linear. Therefore, $\Pi(\mathsf{P}_{NR}) = \{\langle f, \mathsf{a}\rangle, \langle \bullet, \mathsf{a}\rangle\}$.*

As for the DP Framework for TRSs, P_{NR} is not, in general, m-complete [12, Example 32].

Example 89 (Nontermination of $\mathcal{R}$ in Example 80). *Consider the CTRS $\mathcal{R}$ in Example 80 with $\check{\tau}^H = (\{(67), (68)\}, \emptyset, \mathcal{R}, f)$ and $\overline{\mathcal{N}}(\mathcal{R}, (67))$ as in Example 84. Hence, $\mathsf{P}_{NR}(\check{\tau}^H) = \{\check{\tau}_1^H\}$ where $\check{\tau}_1^H = (\{(69), (70), (68)\}, \emptyset, \mathcal{R}, f)$. Now, we can use P_{Inf} to prove $\check{\tau}_1^H$ infinite due to (70) (see Figure 7).*

Our next processor uses $\mathcal{Q}$ to narrow the right-hand sides v of pairs $u \to v \Leftarrow c \in \mathcal{P}$ at the *root*. First, given a CTRS $\mathcal{S}$, we let $NRules^{\Lambda}(\mathcal{S}, t)$ be the set of rules $\alpha : \ell \to r \Leftarrow c \in \mathcal{S}$ such that t is a narrex of α and $N_1^{\Lambda}(\mathcal{S}, s)$ represents the set of one-step $\mathcal{S}$-narrowings issued from s *at the root position*:

$$N_1^{\Lambda}(\mathcal{S}, s) = \{(t, \theta{\downarrow_{\mathcal{V}ar(s)}}) \mid s \overset{\Lambda}{\leadsto}_{\ell \to r \Leftarrow c, \theta} t, \ell \to r \Leftarrow c \in NRules^{\Lambda}(\mathcal{S}, s)\} \quad (71)$$

Finally, if $\alpha : u \to v \Leftarrow c$ is a conditional rule,

$$\overline{\mathcal{N}}_\Lambda(\mathcal{S},\alpha) = \{\theta(u) \to w \Leftarrow \theta(c) \mid (w,\theta) \in N_1^\Lambda(\mathcal{S},v)\} \tag{72}$$

Definition 90 (Narrowing Processor with $\mathcal{Q}$). *Let $\check{\tau} = (\mathcal{P},\mathcal{Q},\mathcal{R},\varphi)$ be an OCTRS problem, $\alpha : u \to v \Leftarrow c \in \mathcal{P}$, and $\mathcal{N}_\Lambda$ be a finite subset of $\overline{\mathcal{N}}_\Lambda(\mathcal{Q},\alpha)$. Then, P_{NQ} is given by*

$$\mathsf{P}_{NQ}(\check{\tau}) = \{(\mathcal{P}[\mathcal{N}_\Lambda]_\alpha,\mathcal{Q},\mathcal{R},\mathsf{a})\}.$$

P_{NQ} changes φ into a because minimality of $(\mathcal{P}[\mathcal{N}_\Lambda]_\alpha,\mathcal{Q},\mathcal{R})$-O-chains A' obtained from a minimal $(\mathcal{P},\mathcal{Q},\mathcal{R})$-O-chains cannot, in general, be guaranteed. This is because operational termination of a term (with respect to $\mathcal{R}$) cannot, in general, be guaranteed after rewritings with $\mathcal{Q}$.

Theorem 91 (Soundness and completeness of P_{NQ}). *P_{NQ} is a-complete. P_{NQ} is sound if $\mathcal{N}_\Lambda = \overline{\mathcal{N}}_\Lambda(\mathcal{Q},\alpha)$, $NRules(\mathcal{R},v) = \emptyset$, for all $u' \to v' \Leftarrow c' \in \mathcal{P}$ (with renamed variables), v and u' do not unify or v and u' unify with* mgu *θ and $\theta(c)$ is $\mathcal{R}$-infeasible, and either v is ground and $\mathcal{R}$ is a deterministic 3-CTRS or $NRules^\Lambda(\mathcal{Q},v)$ is a TRS and v is linear. Therefore, $\Pi(\mathsf{P}_{NQ}) = \{\langle f,\mathsf{a}\rangle,\langle\bullet,\mathsf{a}\rangle\}$.*

Example 92 (Non-V-termination of $\mathcal{R}$ in Example 3). *We apply P_{NQ} to $\check{\tau}^V = (\mathsf{DP}_V(\mathcal{R}),\mathsf{DP}_{VH}(\mathcal{R}),\mathcal{R},f) = (\{(21)\},\{(19),(20)\},\mathcal{R},f)$ in Example 57 (for $\mathsf{DP}_V(\mathcal{R})$ and $\mathsf{DP}_{VH}(\mathcal{R})$ in Example 6) to get $\mathsf{P}_{NQ}(\check{\tau}^V) = \{\check{\tau}_1^V\}$ for $\check{\tau}_1^V = (\{(73)\},\{(19),(20)\},\mathcal{R},\mathsf{a})$ with*

$$\mathsf{F}(\mathsf{a}) \quad \to \quad \mathsf{B} \tag{73}$$

And yet $\mathsf{P}_{NQ}(\check{\tau}_1^V) = \{\check{\tau}_2^V\}$ where $\check{\tau}_2^V = (\{(74)\},\{(19),(20)\},\mathcal{R},\mathsf{a})$ with

$$\mathsf{F}(\mathsf{a}) \quad \to \quad \mathsf{F}(\mathsf{a}) \tag{74}$$

This is an infinite CTRS problem, as can be proved with P_{Inf}. Thus, $\check{\tau}^V$ is infinite (Figure 8) and $\mathcal{R}$ non-V-terminating and hence operationally nonterminating (although terminating*, as proved in Example 59!).*

8. Related work

Our proposal extends the DP Framework for TRSs [10, 12] to CTRSs. However, we differ in several *foundational* aspects which deserve further discussion.

Termination problems. Our definition of CTRS problem is the natural extension of the definition of *DP problem* for TRSs [10, 12] given the theory in [24]. The use we make of such CTRS problems to characterize the different termination properties of CTRSs we are interested in is also similar to what is done in the DP Framework for TRSs. However, the notion of *open* CTRS problem and the corresponding *Open Framework* to deal with the different kinds of termination problems seem to be new in the literature.

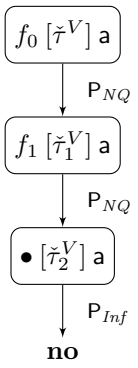


Figure 8: Non-V-termination of $\mathcal{R}$ in Example 3 in the Open 2D DP Framework

Infiniteness of CTRS problems. Our definition of infinite CTRS problem does *not* follow the one in [10, 12] (see Example 17 and the discussion above). In [12, footnote 2], the *asymmetrical* definition of finiteness/infiniteness of DP problems is justified: assuming $\mathcal{R}$ nonterminating is "required for the *completeness* of" most processors for "*dependency pair transformations*". But, actually, this assumption is used to guarantee the preservation of *minimality* of chains. In [12], the main focus is "to show that there is no infinite *minimal* chain" (for proving termination) or, otherwise, proving that there is an infinite *minimal* chain (for proving nontermination). However, for proving nontermination of TRSs, showing the existence of an *arbitrary chain* of DP pairs suffices[6]. We substantiate this fact by providing different results showing that minimality can be advantageously used to prove operational termination of CTRSs (because we consider 'fewer' O-chains), but being more general when proving operational nontermination, where *any arbitrary* infinite O-chain witnesses the property (so finding an example is 'easier'). As a consequence we do *not* impose any specific requirement on chains before analyzing the CTRS. Instead, our *Open 2D DP Framework* leaves the involved processors to impose their requirements.

Soundness/completeness. In [12], an important reason for dealing with minimal chains *only* is that some processors (e.g., the subterm processor) are *sound* under minimality assumptions only. Actually, in the DP Framework, *only sound processors are allowed* in any step of the proof (see [12, Corollary 5]). But soundness is only necessary to prove (operational) termination of (C)TRSs, whereas *completeness* is the crucial property to prove (operational) nontermination. Thus, we do *not* require soundness of all processors involved in a proof (see Theorem 29). In our approach, *plugging schemes* are used to separately manage the description of processors as transformations of (open) CTRS problems and the

[6] see [1, Theorem 6], where the proof clearly shows that from any infinite chain of DPs one can obtain an infinite rewrite sequence that proves nontermination of the TRS.

assignment of a sound/complete behavior within an OCTRSP-tree.

8.1. Transformation techniques

To the best of our knowledge, existing tools for proving operational termination of CTRSs currently use transformation techniques. Transformation $\mathcal{U}$ [28, Definition 7.2.48] is often used for this purpose. This transformation is not complete, though, and may *fail* to prove some examples (see [24, Section 8.1.2]).

Example 93. *Consider the following CTRS $\mathcal{R}$ [9, page 46]:*

$$\mathsf{a} \to \mathsf{b} \tag{75}$$
$$\mathsf{f}(\mathsf{a}) \to \mathsf{b} \tag{76}$$
$$\mathsf{g}(x) \to \mathsf{g}(\mathsf{a}) \Leftarrow \mathsf{f}(x) \to x \tag{77}$$

As noticed by Giesl and Arts [9], the transformed TRS $\mathcal{U}(\mathcal{R})$:

$$\begin{array}{rcl@{\qquad}rcl} \mathsf{a} & \to & \mathsf{b} & \mathsf{g}(x) & \to & \mathsf{U}(\mathsf{f}(x), x) \\ \mathsf{f}(\mathsf{a}) & \to & \mathsf{b} & \mathsf{U}(x, x) & \to & \mathsf{g}(\mathsf{a}) \end{array}$$

is not terminating: $\underline{\mathsf{g}(\mathsf{a})} \to \mathsf{U}(\underline{\mathsf{f}(\mathsf{a})}, \mathsf{a}) \to \mathsf{U}(\mathsf{b}, \underline{\mathsf{a}}) \to \underline{\mathsf{U}(\mathsf{b}, \mathsf{b})} \to \mathsf{g}(\mathsf{a}) \to \cdots$ We can give a simple proof of operational termination of $\mathcal{R}$: $\mathsf{DP}_H(\mathcal{R})$ and $\mathsf{DP}_V(\mathcal{R})$ are:

$$\mathsf{DP}_H(\mathcal{R}) \qquad \mathsf{G}(x) \to \mathsf{G}(\mathsf{a}) \Leftarrow \mathsf{f}(x) \to x \tag{78}$$
$$\mathsf{G}(x) \to \mathsf{A} \Leftarrow \mathsf{f}(x) \to x \tag{79}$$
$$\mathsf{DP}_V(\mathcal{R}) \qquad \mathsf{G}(x) \to \mathsf{F}(x) \tag{80}$$

and $\mathsf{DP}_{VH}(\mathcal{R}) = \emptyset$. The two pairs in $\mathsf{DP}_H(\mathcal{R})$ are $\mathcal{R}$-infeasible, see [19, Example 9]. Hence $\mathsf{P}_{SCC}(\check{\tau}^H) = \emptyset$. And, since there is no cycle in $\mathsf{EG}(\check{\tau}^V)$, we also have $\mathsf{P}_{SCC}(\check{\tau}^V) = \emptyset$, i.e., $\mathcal{R}$ is operationally terminating.

Also, due to its incompleteness, disproving operational termination is *not possible* with this transformation, in sharp contrast to our approach.

9. Conclusion

We briefly explain the new contributions of this paper.

1. In Section 5 we have generalized the Dependency Pair Framework for TRSs to deal with CTRSs and the different termination properties for CTRSs investigated in [24]. With the same notion of CTRS problem and using the same processors, we can prove and disprove not only operational termination of CTRSs, but also termination and V-termination of CTRSs.
2. We define an *Open* Dependency Pair Framework, where we do *not* assume any specific kind of (initial) CTRS problems (Section 6). As far as we know, the proposal of such an *open* framework is new and could be used in other adaptations of the DP Framework to other variants of rewriting where similar situations occur (in particular, in the DP Framework itself).

3. We have introduced several processors to be used within the 2D DP Framework: some of them (e.g., P_{Inf}, P_{SCC}, $\mathsf{P}_{\triangleright}$, P_{RT}, and P_{NR}) adapt existing processors from the DP Framework for TRSs. In other cases (e.g., P_{IR}, P_{RIR}, and P_{NQ}), they exploit specific features of CTRSs (e.g., the conditions in the rule) or peculiarities of the 2D DP Framework.
4. We have shown by means of several examples that we can actually prove different termination properties that 'coexist' in the same CTRS. For instance, the CTRS in Example 3 has been proved *terminating* in Example 59 and *non-V-terminating* in Example 92.
5. We have proved operational termination of CTRSs which could not be proved operationally terminating with existing techniques (in particular, by means of the usual tranformations, see Example 93). Moreover, tools based on these (incomplete) transformations are *not* able to disprove operational termination of CTRSs (as we do in Examples 89 and 92).

The 2D DP Framework has been implemented as part of the tool MU-TERM. We have participated in the 2014, 2015, 2016, and 2017 editions of the *International Termination Competition* where we obtained the first position among the tools in the *TRS Conditional* subcategory, see

`http://zenon.dsic.upv.es/muterm/?page_id=82`

for a summary. However, although the processors introduced in this first part of the paper have been proved poweful enough to deal with CTRSs which cannot be handled by using the usual transformation techniques[7] (e.g., $\mathcal{R}$ in Example 3 cannot be proved operationally nonterminating by using the usual transformation $\mathcal{U}$ but it is proved operationally nonterminating in Example 92; similarly for $\mathcal{R}$ in Example 93), more processors are necessary to obtain the good results exhibited in practice. The second part of this paper [27] develops such processors and also provides details about the implementation of the 2D DP Framework, including statistics regarding the use of the different processors and benchmarks comparing MU-TERM and other termination tools.

Acknowledgments. We thank the anonymous referees for many remarks and suggestions that led to improve the paper.

References

[1] T. Arts and J. Giesl. Termination of Term Rewriting Using Dependency Pairs. *Theoretical Computer Science*, 236(1–2):133–178, 2000.

[2] B. Alarcón, R. Gutiérrez, S. Lucas, R. Navarro-Marset. Proving Termination Properties with MU-TERM. In M. Johnson and D. Pavlovic, editors, *Proc. of the 13th International Conference on Algebraic Methodology and Software Technology, AMAST'10*, LNCS 6486:201-208, 2011.

[7] Sound and complete transformations for proving operational termination of some classes of CTRSs have been reported, see, e.g., [18]. Yet, their practical application is underexplored.

[3] J.A. Bergstra and J.W. Klop. Conditional Rewrite Rules: Confluence and Termination. *Journal of Computer and System Sciences* 32:323-362, 1986.

[4] P. Borovanský, C. Kirchner, H. Kirchner, P.-E. Moreau, and C. Ringeissen. An Overview of ELAN. In C. Kirchner and H. Kirchner, editors, *Proc. of 2nd International Workshop on Rewriting Logic and its Applications, WRLA'98*, Electronic Notes in Theoretical Computer Science, 15(1998):1-16, 1998.

[5] M.G.J. van den Brand, J. Heering, P. Klint, and P.A. Olivier. Compiling language definitions: The ASF+SDF compiler. *ACM Transactions on Programming Languages and Systems*, 24(4):334–368, 2002.

[6] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. All About Maude – A High-Performance Logical Framework. LNCS 4350, Springer-Verlag, 2007.

[7] N. Dershowitz and M. Okada. A rationale for conditional equational programming. *Theoretical Computer Science* 75:111-138, 1990.

[8] K. Futatsugi and R. Diaconescu. *CafeOBJ Report*. World Scientific, AMAST Series in Computing, Volume 6, 1998.

[9] J. Giesl and T. Arts. Verification of Erlang Processes by Dependency Pairs. *Applicable Algebra in Engineering, Communication and Computing* 12:39-72, 2001.

[10] J. Giesl, R. Thiemann, and P. Schneider-Kamp. The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs. In F. Baader and A. Voronkov, editors, *Proc. of XI International Conference on Logic for Programming Artificial Intelligence and Reasoning, LPAR'04*, LNAI 3452:301–331, Springer-Verlag, Berlin, 2004.

[11] J. Giesl, R. Thiemann, and P. Schneider-Kamp. Proving and Disproving Termination of Higher-Order Functions. In B. Gramlich, editor, *Proc. of 5th International Workshop on Frontiers of Combining Systems, FroCoS'05*, LNAI 3717:216-231, Springer-Verlag, Berlin, 2005.

[12] J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and Improving Dependency Pairs. *Journal of Automatic Reasoning*, 37(3):155–203, 2006.

[13] J.A. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ. In J. Goguen and G. Malcolm, editors, *Software Engineering with OBJ: algebraic specification in action*, Kluwer, 2000.

[14] N. Hirokawa and A. Middeldorp. Dependency Pairs Revisited. In V. van Oostrom, editor, *Proc. of XV International Conference on Rewriting Techniques and Applications, RTA'04*, LNCS 3091:249–268, Springer-Verlag, Berlin, 2004.

[15] P. Hudak, S.J. Peyton-Jones, and P. Wadler. Report on the Functional Programming Language Haskell: a non–strict, purely functional language. *Sigplan Notices*, 27(5):1-164, 1992.

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

[17] D. Kapur, D. Musser, P. Narendran, and J. Stillman. Semi-unification. *Theoretical Computer Science* 81:169-187, 1991.

[18] C. Kop, A. Middeldorp, and T. Sternagel. Complexity of Conditional Term Rewriting. *Logical Methods in Computer Science* 13(1), February 2017.

[19] S. Lucas. Analysis of Rewriting-Based Systems as First-Order Theories. In F. Fioravanti and J.P. Gallagher, editors, *Revised Selected papers from the 27th International Symposium on Logic-Based Program Synthesis and Transformation, LOPSTR 2017*, LNCS volume 10855, to appear, 2018.

[20] S. Lucas and R. Gutiérrez. Automatic Synthesis of Logical Models for Order-Sorted First-Order Theories. *Journal of Automated Reasoning*, 60(4):465–501, 2018.

[21] S. Lucas and R. Gutiérrez. Use of logical models for proving infeasibility in term rewriting. *Information Processing Letters*, 136C:90-95, 2018.

[22] S. Lucas, C. Marché, and J. Meseguer. Operational termination of conditional term rewriting systems. *Information Processing Letters* 95:446–453, 2005.

[23] S. Lucas and J. Meseguer. 2D Dependency Pairs for Proving Operational Termination of CTRSs. In S. Escobar, editor, *Proc. of the 10th International Workshop on Rewriting Logic and its Applications, WRLA'14*, LNCS 8663:195-212, 2014.

[24] S. Lucas and J. Meseguer. Dependency pairs for proving termination properties of conditional term rewriting systems. *Journal of Logical and Algebraic Methods in Programming*, 86:236-268, 2017.

[25] S. Lucas and J. Meseguer. Normal forms and normal theories in conditional rewriting. *Journal of Logical and Algebraic Methods in Programming*, 85(1):67-97, 2016.

[26] S. Lucas, J. Meseguer, and R. Gutiérrez. Extending the 2D DP Framework for Conditional Term Rewriting Systems. In M. Proietti and H. Seki, editors, Selected papers of the *24th International Symposium on Logic-Based Program Synthesis and Transformation LOPSTR'14*, LNCS 8981:113-130, 2015.

[27] S. Lucas, J. Meseguer, and R. Gutiérrez. The 2D DP Framework for Conditional Term Rewriting Systems. Part II: advanced processors. In preparation, 2018.

[28] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer-Verlag, Apr. 2002.

[29] R.M. Smullyan. Theory of Formal Systems. Princeton University Press, 1961.

[30] T. Sternagel and A. Middeldorp. Conditional Confluence (System Description). In G. Dowek, editor, *Proc. of Joint International Conference on Rewriting and Typed Lambda Calculi, RTA-TLCA'14*, LNCS 8560:456-465, 2014.

[31] T. Sternagel and A. Middeldorp. Infeasible Conditional Critical Pairs. In A. Tiwari and T. Aoto, editors, *Proc. of the 4th International Workshop on Confluence, IWC'15*, pages 13–18, 2014.

Appendix A. Proofs of Theorems

Theorem 23. P_{Fin} *is sound and complete.* P_{Inf} *is sound.* P_{Inf} *is sound and* a*-complete.* P_{Inf} *is* τ*-complete if* $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{m})$ *and* $u \to v \Leftarrow c$ *in Definition 20 is such that* v *is ground and contains no symbol from* $\mathcal{D}_{\mathcal{R}}$*.*

Proof. Let $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f)$ be a CTRS problem with $\mathcal{P} = \emptyset$. There is no (minimal or arbitrary) $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain due to the emptiness of $\mathcal{P}$. Thus, soundness of P_{Fin} is obvious for all $f \in \mathsf{F}$. Completeness is also trivial.

Soundness of P_{Inf} is obvious. Regarding completeness, if $\mathsf{P}_{Inf}(\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) = \mathsf{no}$, then we can define an infinite $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain A as follows. The sequence of pairs is $(u_i \to v_i \Leftarrow c_i)_{i \geq 1}$, where $u_i \to v_i \Leftarrow c_i$ are renamed versions of $\alpha : u \to v \Leftarrow c$. If $x \in \mathcal{V}ar(\alpha)$, then x_i is the corresponding variable of $\alpha_i : u_i \to v_i \Leftarrow c_i$, where $x_i \neq y_i$ if $x, y \in \mathcal{V}ar(\alpha)$ are different ($x \neq y$) and for all $i \neq j$, $\mathcal{V}ar(\alpha_i) \cap \mathcal{V}ar(\alpha_j) = \emptyset$. Let $\rho_i : \mathcal{V}ar(\alpha_i) \to \mathcal{V}ar(\alpha)$ be the corresponding renaming of variables of α_i into variables of α. For all $i \geq 1$, we have $\rho_i(u_i) = u$ and $\rho_i(v_i) = v$. For all variables x, we let σ be given by $\sigma(x_1) = \eta(\rho_1(x_1)) = \eta(x)$ and, for all $i \geq 1$, $\sigma(x_{i+1}) = \theta(\sigma(x_i))$. Note that, by definition of σ, for all $i \geq 1$, $\sigma(u_{i+1}) = \theta(\sigma(u_i))$ and $\sigma(v_{i+1}) = \theta(\sigma(v_i))$. We prove by induction that, for all $i \geq 1$, $\sigma(v_i) = \sigma(\rho_i^{-1}(v)) = \sigma(\rho_{i+1}^{-1}(u)) = \sigma(u_{i+1})$ and for all $s \to t \in c$, $\sigma(\rho_i^{-1}(s)) \to_{\mathcal{R}}^* \sigma(\rho_i^{-1}(t))$. For the base case, we have $\sigma(v_1) = \eta(\rho_1(v_1)) = \eta(v) = \theta(\eta(u))$ and $\sigma(u_2) = \theta(\sigma(u_1)) = \theta(\eta(\rho_1(u_1))) = \theta(\eta(u))$. Therefore, $\sigma(v_1) = \sigma(u_2)$. We also have $\sigma(\rho_1^{-1}(s)) = \eta(\rho_1(\rho^{-1}(s))) = \eta(s) \to_{\mathcal{R}}^* \eta(t) = \sigma(\rho_1^{-1}(t))$. For the induction step, by definition $\sigma(v_{i+1}) = \theta(\sigma(v_i))$, and by the I.H., $\sigma(v_i) = \sigma(u_{i+1})$. Therefore, $\sigma(v_{i+1}) = \theta(\sigma(u_{i+1})) = \sigma(u_{i+2})$. And also, $\sigma(\rho_{i+1}^{-1}(s)) = \theta(\sigma(\rho_i^{-1}(s)))$. By the I.H., $\sigma(\rho_i^{-1}(s)) \to_{\mathcal{R}}^* \sigma(\rho_i^{-1}(t))$. Thus, by stability of rewriting, $\theta(\sigma(\rho_i^{-1}(s))) \to_{\mathcal{R}}^* \theta(\sigma(\rho_i^{-1}(t)))$, i.e., $\sigma(\rho_{i+1}^{-1}(s)) \to_{\mathcal{R}}^* \sigma(\rho_{i+1}^{-1}(t))$ as required. Hence, A with σ defines an infinite $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain. If, additionally, v is ground and contains no symbol from $\mathcal{D}_{\mathcal{R}}$, then it is obviously operationally terminating with respect to $\mathcal{R}$. Hence, A is minimal. □

Proposition 25. *Let* P *be a processor such that, for all CTRS problems* $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) \in \mathsf{Dom}(\mathsf{P})$, $\mathsf{P}(\tau) \neq \mathsf{no}$ *and if* $\tau' = (\mathcal{P}', \mathcal{Q}', \mathcal{R}', f') \in \mathsf{P}(\tau)$*, then* $\mathcal{P}' \subseteq \mathcal{P}$, $\mathcal{Q}' \subseteq \mathcal{Q}$, $\mathcal{R}' \subseteq \mathcal{R}$ *and (i)* $f = \mathsf{a}$ *or (ii)* $f' = f = \mathsf{m}$ *and* $\mathcal{R}' = \mathcal{R}$*. Then,* P *is complete.*

Proof. Since $\mathcal{P}' \subseteq \mathcal{P}$, $\mathcal{Q}' \subseteq \mathcal{Q}$, and $\mathcal{R}' \subseteq \mathcal{R}$, every infinite (minimal or arbitrary) $(\mathcal{P}', \mathcal{Q}', \mathcal{R}')$-O-chain A' is an infinite $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain. If $f = \mathsf{a}$ (case *(i)*), then τ is infinite if τ' is infinite. If $f = f' = \mathsf{m}$ and $\mathcal{R}' = \mathcal{R}$ (case *(ii)*), then if A' is minimal, then A' viewed as a $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain is also minimal, i.e., τ is infinite if τ' is infinite. Hence P is complete. □

Theorem 43. *Let* $\check{\tau}_I = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ *be an open CTRS problem and* $\check{\mathcal{T}}_I$ *be an open CTRSP-tree for* $\check{\tau}_I$*. Then,*

1. *If all leaves in* $\check{\mathcal{T}}_I$ *are labeled with "*yes*" and there is a ground flag substitution* ς *that* unifies $\mathcal{E}_s(\check{\mathcal{T}}_I)$ *and* $\varsigma(\varphi) \neq \bullet$*, then* $\varsigma(\check{\tau}_I)$ *is* finite*.*

2. *If there is a leaf* n_∞ *in* $\check{\mathcal{T}}_I$ *with label "*no*" and there is a ground flag substitution* ς *that* unifies $\mathcal{E}_c^{\mathsf{n}\infty}(\check{\mathcal{T}}_I)$ *and* $\varsigma(\varphi) \neq \bullet$*, then* $\varsigma(\check{\tau}_I)$ *is* infinite.

PROOF. Let n_I be the root node of $\check{\mathcal{T}}_I$ with labels $\check{\tau}_I$ and $\psi_I = \langle \varphi_s^I, \varphi_c^I \rangle$. There is an open processor P such that $\check{\tau}_I \in \mathsf{Dom}\check{}(\mathsf{P})$ and P follows ψ_I with $\check{\tau}_I$.

1. We prove $\varsigma(\check{\tau}_I)$ finite by induction on the number $N > 1$ of nodes in $\check{\mathcal{T}}_I$.
   (a) If $N = 2$, then $\check{\mathcal{T}}_I$ consists of two nodes: n_I and n', which is a leaf with label "yes", and $\mathcal{E}_s(\check{\mathcal{T}}_I) = \{\varphi =^? \varphi_s^I\}$. Since $\varsigma(\varphi) = \varsigma(\varphi_s^I) \neq \bullet$, we can assume $\varsigma = \varsigma_f$ for some $f \in \mathsf{F}$. Since P follows ψ_I with $\check{\tau}_I$, P is $(\check{\tau}_I, f)$-*sound*. By Definition 41, $\mathsf{P}(\check{\tau}_I) = \emptyset$. By Definition 36, $\varsigma(\check{\tau}_I)$ is finite.
   (b) If $N > 2$, then, $\mathsf{P}(\check{\tau}_I) = \{\check{\tau}_1, \ldots, \check{\tau}_m\}$ for some $m > 0$, so that n_I has m children $\mathsf{n}_1, \ldots, \mathsf{n}_m$ which are the roots of OCTRSP-trees $\check{\mathcal{T}}_i$ for $1 \leq i \leq m$. For each i, $1 \leq i \leq m$, let $\check{\tau}_i = (\mathcal{P}_i, \mathcal{Q}_i, \mathcal{R}_i, \varphi_i)$ and $\psi_i = \langle \varphi_s^i, \varphi_c^i \rangle$ be the open CTRS problem and plugging scheme labeling n_i. Since $\mathcal{E}_s(\check{\mathcal{T}}_I) = \{\varphi =^? \varphi_s^I\} \cup \bigcup_{i=1}^m \mathcal{E}_s(\check{\mathcal{T}}_i)$, for all i, $1 \leq i \leq m$, ς is a ground unifier of $\mathcal{E}_s(\check{\mathcal{T}}_i)$. By Definition 35, for all $1 \leq i \leq m$, $\varphi_i \in \mathsf{F} \cup \{\varphi\}$; thus, $\varsigma(\varphi_i) \neq \bullet$. By the induction hypothesis, for all i, $1 \leq i \leq m$, $\varsigma(\check{\tau}_i)$ is finite. Since ς is a ground unifier of $\mathcal{E}_s(\check{\mathcal{T}}_I)$ and $\varsigma(\varphi) \neq \bullet$, we have $\varsigma_f(\varphi) = \varsigma_f(\varphi_s^I)$ for some $f \in \mathsf{F}$. Since P follows ψ_I with $\check{\tau}_I$, this means that P is $(\check{\tau}_I, f)$-sound. Thus, $\varsigma_f(\check{\tau}_I)$, i.e., $\varsigma(\check{\tau}_I)$ is finite.
2. Let Γ be a path from n_I to a node n_∞ with label "no". We prove by induction on the number $N > 1$ of nodes in Γ that $\varsigma(\check{\tau}_I)$ is infinite.
   (a) If $N = 2$, then n_I is the predecessor of n_∞ in Γ. And $\mathcal{E}_c(\check{\mathcal{T}}) = \{\varphi =^? \varphi_{ic}^I\}$, i.e., $\varsigma_f(\varphi) = \varsigma_f(\varphi_c^I)$ for some $f \in \mathsf{F}$. Since P follows ψ_I with $\check{\tau}_I$, P is $(\check{\tau}_I, f)$-complete. Hence, since $\mathsf{P}(\check{\tau}_I) =$ "no", $\varsigma(\check{\tau}_I)$ is infinite.
   (b) If $N > 2$, then let n be the successor of n_I in Γ, with labels $\check{\tau} = (\mathcal{P}', \mathcal{Q}', \mathcal{R}', \varphi')$ and $\psi = \langle \varphi_s, \varphi_c \rangle$. We have $\check{\tau} \in \mathsf{P}(\check{\tau}_I)$ and by Definition 35, $\varphi' \in \mathsf{F} \cup \{\varphi\}$; thus, $\varsigma(\varphi') \neq \bullet$. Since $\mathcal{E}_c^{\mathsf{n}\infty}(\check{\mathcal{T}}_I) = \{\varphi_I =^? \varphi_c^I\} \cup \mathcal{E}_c^{\mathsf{n}\infty}(\check{\mathcal{T}})$, ς is a unifier of $\mathcal{E}_c^{\mathsf{n}\infty}(\check{\mathcal{T}})$ and by the induction hypothesis, $\varsigma(\check{\tau})$ is infinite. Since ς is a ground unifier of $\mathcal{E}_c^{\mathsf{n}\infty}(\check{\mathcal{T}}_I)$ and $\varsigma(\varphi) \neq \bullet$, we have $\varsigma_f(\varphi) = \varsigma_f(\varphi_s^I)$ for some $f \in \mathsf{F}$. Since P follows ψ_I with $\check{\tau}$, P is $(\check{\tau}, f)$-complete and $\varsigma_f(\check{\tau}_I)$, and hence $\varsigma(\check{\tau}_I)$, is infinite.□

Theorem 45 P_{arb} *is sound and* a*-complete.* P_{min} *is* m*-sound and complete.*

PROOF. As for P_{arb}, if $(\mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathsf{a})$ is finite, then for all f such that $\varsigma_f(\varphi) = f$, since $f \leq \mathsf{a}$, $\varsigma_f(\check{\tau}) = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varsigma_f(\varphi))$ is finite, i.e., P_{arb} is sound. The proof of a-completeness is immediate. The proofs for P_{min} are analogous. □

Theorem 52. P_{SCC} *is sound and complete.*

PROOF. Completeness follows by Corollary 27. We prove soundness by contradiction. If there is $\tau = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, f) = \varsigma_f(\check{\tau})$ for some $f \in \mathsf{F}$ such that, for all $\tau' \in \mathsf{P}_{SCC}(\check{\tau})$, $\tau' = \varsigma_f(\check{\tau}) = (\mathcal{P}', \mathcal{Q}', \mathcal{R}, f)$ is finite but τ is not finite, then there

is an infinite (minimal) $(\mathcal{P},\mathcal{Q},\mathcal{R})$-O-chain A. Since $\mathcal{P}$ contains a finite number of pairs, there is $\mathcal{P}'' \subseteq \mathcal{P}$ and a tail B of A, which is an infinite (minimal) $(\mathcal{P}'',\mathcal{Q}'',\mathcal{R})$-O-chain where all pairs in $\mathcal{P}''$ are infinitely often used and $\mathcal{Q}'' \subseteq \mathcal{Q}$ is the subset of pairs in $\mathcal{Q}$ which are used in B. According to Definitions 46 and 47, $\mathcal{P}''$ is a cycle in $\mathsf{AG}(\check{\tau})$ included in some SCC with nodes in $\mathcal{P}'$, i.e., $\mathcal{P}'' \subseteq \mathcal{P}'$. And according to Definition 49, $\mathcal{Q}'' \subseteq \mathcal{Q}'$. Thus, B is an infinite (minimal) $(\mathcal{P}',\mathcal{Q}',\mathcal{R})$-O-chain and τ' is infinite; a contradiction. □

Proposition 53. *Let $\mathcal{R}$ be a CTRS, t,u be terms, and σ be a substitution. If $\sigma(t) \to_{\mathcal{R}}^* \sigma(u)$, then $\textsc{tcap}_{\mathcal{R}}(t)$ and u unify.*

Proof. In the following, we let $s = \textsc{tcap}_{\mathcal{R}}(t)$. By definition of $\textsc{tcap}_{\mathcal{R}}$, we can assume s linear and $\mathcal{V}ar(s) \cap \mathcal{V}ar(u) = \emptyset$. Also, $t = \sigma_t(s)$ for some substitution σ_t. We proceed by induction on the length m of the sequence from $\sigma(t)$ to $\sigma(u)$. If $m = 0$, then $\sigma(t) = \sigma(\sigma_t(s)) = \sigma(u)$. Since $\mathcal{V}ar(s) \cap \mathcal{V}ar(u) = \emptyset$, s and u unify. If $m > 0$, then $\sigma(t) = \sigma(\sigma_t(s)) \to t' \to^* \sigma(u)$. Let $p \in \mathcal{P}os(\sigma(t))$ be the position where the rewrite step $\sigma(t) \to t'$ with rule $\ell \to r \Leftarrow c$ is performed. By definition of $\textsc{tcap}_{\mathcal{R}}$, $s = s[z]_q$ for some fresh variable z and position q such that $q \leq p$. Therefore, $t' = \sigma(t)[\sigma(r)]_p = \sigma(\sigma_t(s))[\sigma(r)]_p = \sigma(\sigma'_t(s))$ where $\sigma'_t(z) = t'|_q$ and $\sigma'_t(x) = \sigma_t(x)$ for any other variable $x \neq z$. Thus, $\sigma(\sigma_t(s)) \to \sigma(\sigma'_t(s))$ and $\sigma(\sigma'_t(s)) \to^* \sigma(u)$ in $m-1$ steps. By the induction hypothesis, $\textsc{tcap}_{\mathcal{R}}(\sigma'_t(s))$ and u unify with *mgu* θ. Since $\sigma_t(s)$ and $\sigma'_t(s)$ may differ below position q only, we can assume that $\textsc{tcap}_{\mathcal{R}}(\sigma'_t(s)) = s[s']_q$ for some term s' sharing no variable with $s[]_q$ or u. Thus, $\textsc{tcap}_{\mathcal{R}}(t) = s$ and u unify with *mgu* θ' where $\theta'(z) = \theta(s')$ and $\theta'(x) = \theta(x)$ for any other variable $x \neq z$. □

Theorem 65. *$\mathsf{P}_{\rhd}$ is complete and $(\check{\tau},\mathsf{m})$-sound for all $\check{\tau} = (\mathcal{P},\mathcal{Q},\mathcal{R},\varphi) \in \mathsf{Dom}^{\check{}}(\mathsf{P}_{\rhd})$ such that $\mathcal{P} \cup \mathcal{Q}$ is not collapsing and $(Root(\mathcal{P}) \cup Root(\mathcal{Q})) \cap \mathcal{D}_{\mathcal{R}} = \emptyset$.*

Proof. Completeness follows by Corollary 27. For soundness, we proceed by contradiction. If there is an infinite minimal $(\mathcal{P},\mathcal{Q},\mathcal{R})$-chain A but there is no infinite minimal $(\mathcal{P}[\emptyset]_\alpha,\mathcal{Q}[\emptyset]_\alpha,\mathcal{R})$-chain, since $\mathcal{P}$ and $\mathcal{Q}$ are finite, there are $\mathcal{P}' \subseteq \mathcal{P}$ and $\mathcal{Q}' \subseteq \mathcal{Q}$ such that A has a tail B that is an infinite minimal $(\mathcal{P}',\mathcal{Q}',\mathcal{R})$-chain where all pairs in $\mathcal{P}'$ and $\mathcal{Q}'$ are infinitely often used:

$$\sigma(u^1) \xrightarrow{\Lambda}_{\mathcal{P}',\mathcal{R}} \sigma(v^1)(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}{}^{=}_{\mathcal{Q}',\mathcal{R}})^*\sigma(u^2) \xrightarrow{\Lambda}_{\mathcal{P}',\mathcal{R}} \cdots$$

for some substitution σ. Note that, for all $i \geq 1$, $root(u^i) \in Root(\mathcal{P})$. Since $root(v^i) \notin \mathcal{X}$, we have that $root(v^i) \in Root(\mathcal{P})$. Thus, we can apply π to $\sigma(u^i)$ and $\sigma(v^i)$ for all $i \geq 1$. Since $\sigma(v^i)(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}{}^{=}_{\mathcal{Q},\mathcal{R}})^*\sigma(u^{i+1})$ for all $i \geq 1$ and $(Root(\mathcal{P}) \cup Root(\mathcal{Q})) \cap \mathcal{D}_{\mathcal{R}} = \emptyset$, we can write $\sigma(v^i)(\xrightarrow{>\Lambda}{}^*_{\mathcal{R}} \circ \xrightarrow{\Lambda}{}^{=}_{\mathcal{Q}',\mathcal{R}})^*\sigma(u^{i+1})$ because rewritings with $\mathcal{R}$ do not change $root(v^i) = root(\sigma(v^i))$ and we have $\pi(u') \trianglerighteq \pi(v')$ for all $u' \to v' \Leftarrow c' \in \mathcal{Q}'$. Hence, for all $i \geq 1$, $\pi(\sigma(v^i))(\to_{\mathcal{R}}^* \circ \trianglerighteq)^*\pi(\sigma(u^{i+1}))$. Finally, since $\pi(u^i) \trianglerighteq \pi(v^i)$ for all $i \geq 1$, by stability of $\trianglerighteq$, we have $\pi(\sigma(u^i)) = \sigma(\pi(u^i)) \trianglerighteq \sigma(\pi(v^i)) = \pi(\sigma(v^i))$ for all $i \geq 1$. Note that $u \to v \Leftarrow c \notin \mathcal{P}' \cup \mathcal{Q}'$. Otherwise, we get a contradiction: since A is minimal, we can assume that $\sigma(v^1)$ is operationally terminating (w.r.t. $\mathcal{R}$). Since

$\pi(\sigma(v^i))(\to^*_{\mathcal{R}} \circ \trianglerighteq)^* \pi(\sigma(u^{i+1}))$ and $\pi(\sigma(u^i)) \trianglerighteq \pi(\sigma(v^i))$ for all $i \geq 1$, the sequence B is transformed into an infinite $\to_{\mathcal{R}} \cup \trianglerighteq$-sequence

$$\pi(\sigma(v^1))(\to^*_{\mathcal{R}} \circ \trianglerighteq)^* \pi(\sigma(u^2)) \trianglerighteq \pi(\sigma(v^2))(\to^*_{\mathcal{R}} \circ \trianglerighteq)^* \pi(\sigma(u^3)) \trianglerighteq \pi(\sigma(v^2))(\to^*_{\mathcal{R}} \circ \trianglerighteq)^* \cdots$$

containing infinitely many $\triangleright$-steps, due to $\pi(u) \triangleright \pi(v)$ for $u \to v \Leftarrow c$ which occurs infinitely often in B. Since $\triangleright$ is well-founded, the infinite sequence must also contain infinitely many $\to_{\mathcal{R}}$-steps. By making repeated use of the fact that $\triangleright \circ \to_{\mathcal{R}} \subseteq \to_{\mathcal{R}} \circ \triangleright$, we obtain an infinite $\to_{\mathcal{R}}$-sequence starting from $\pi(\sigma(v^1))$. Thus, $\pi(\sigma(v^1))$ is not operationally terminating with respect to $\mathcal{R}$, leading to a contradiction. Since $\mathcal{P}' \subseteq \mathcal{P}[\emptyset]_\alpha$ and $\mathcal{Q}' \subseteq \mathcal{Q}[\emptyset]_\alpha$, B is an infinite minimal $(\mathcal{P}[\emptyset]_\alpha, \mathcal{Q}[\emptyset]_\alpha, \mathcal{R})$-chain. This contradicts our initial argument. □

Theorem 71. P_{RT} *is sound and complete.*

Proof. Completeness follows by Corollary 27. Regarding soundness, we proceed by contradiction. Assume that A is an infinite (minimal) $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-chain, but there is no infinite (minimal) $(\mathcal{P}[\emptyset]_\alpha, \mathcal{Q}[\emptyset]_\alpha, \mathcal{R})$-chain. By finiteness of $\mathcal{P}$ and $\mathcal{Q}$, there are $\mathcal{P}' \subseteq \mathcal{P}$ and $\mathcal{Q}' \subseteq \mathcal{Q}$ such that A has a tail B

$$\sigma(u^1) \xrightarrow{\Lambda}_{\mathcal{P}',\mathcal{R}} \sigma(v^1)(\to^*_{\mathcal{R}} \circ \xrightarrow{\Lambda}^{=}_{\mathcal{Q}',\mathcal{R}})^* \sigma(u^2) \xrightarrow{\Lambda}_{\mathcal{P}',\mathcal{R}} \cdots$$

for some substitution σ, where all pairs in $\mathcal{P}'$ and $\mathcal{Q}'$ are infinitely often used. Although $\mathcal{Q}'$ could be *empty* (if no pair in $\mathcal{Q}'$ is used to *connect* pairs in $\mathcal{P}'$) $\mathcal{P}'$ is not empty. By Definition 69.2, for all $i \geq 1$ and $u^i \to v^i \Leftarrow c^i \in \mathcal{P}'$,

$$\sigma(u^i) \quad (\gtrsim \cup \succeq \cup \sqsupseteq) \quad \sigma(v^i) \tag{A.1}$$

Note that $\alpha \notin \mathcal{P}' \cup \mathcal{Q}'$. Otherwise, we get a contradiction as follows. First, note that, without loss of generality, we can assume that $u^1 \to v^1 \Leftarrow c^1$ in B is (a renamed version of) α. Therefore, $\sigma(u^1) \sqsupseteq \sigma(v^1)$, by hypothesis. Since $\sigma(v^i)(\to^*_{\mathcal{R}} \circ \xrightarrow{\Lambda}^{=}_{\mathcal{Q}',\mathcal{R}})^* \sigma(u^{i+1})$, there are pairs $\overline{u}^k_i \to \overline{v}^k_i \Leftarrow \bigwedge_{j=1}^{n_{ik}} \overline{u}^k_{ij} \to \overline{v}^k_{ij} \in \mathcal{Q}'$ for k, $1 \leq k \leq \kappa_i$ ($\kappa_i = 0$ indicates that no pair in $\mathcal{Q}'$ is necessary to connect $\sigma(v^i)$ and $\sigma(u^{i+1})$; if $\mathcal{Q}' = \emptyset$, then $\kappa_i = 0$ for all $i \geq 0$) such that $\sigma(\overline{u}^k_{ij}) \to^*_{\mathcal{R}} \sigma(\overline{v}^k_{ij})$ for all j, $1 \leq j \leq n_{ik}$ and k, $1 \leq k \leq \kappa_i$, and

$$\sigma(v^i) \to^*_{\mathcal{R}} \sigma(\overline{u}^1_i) \xrightarrow{\Lambda}_{\mathcal{Q},\mathcal{R}} \sigma(\overline{v}^1_i) \to^*_{\mathcal{R}} \cdots \to^*_{\mathcal{R}} \sigma(\overline{u}^{\kappa_i}_i) \xrightarrow{\Lambda}_{\mathcal{Q},\mathcal{R}} \sigma(\overline{v}^{\kappa_i}_i) \to^*_{\mathcal{R}} \sigma(u^{i+1})$$

for $i \geq 1$. By Definition 69.1, for all k, $1 \leq k < \kappa_i$ we have

$$\sigma(v^i) \gtrsim^* \sigma(\overline{u}^1_i) \quad \text{and} \quad \sigma(\overline{v}^k_i) \gtrsim^* \sigma(\overline{u}^{k+1}_i) \tag{A.2}$$

and $\sigma(\overline{v}^{\kappa_i}_i) \gtrsim^* \sigma(u^{i+1})$ (or $\sigma(v^i) \gtrsim^* \sigma(u_{i+1})$ if $\kappa_i = 0$). Also

$$\sigma(u^k_i) \quad (\gtrsim \cup \succeq \cup \sqsupseteq) \quad \sigma(v^k_i) \tag{A.3}$$

for all k, $1 \leq k < \kappa_i$. From $(A.1)$ and $(A.3)$ we conclude that $\sigma(u^i)\,(\gtrsim \cup \succeq \cup \sqsupseteq)^+\, \sigma(u^{i+1})$. Since α occurs infinitely often in B, there is an infinite set $\mathcal{J} \subseteq \mathbb{N}$

$$\text{(NRf)}\quad \frac{}{x \rightsquigarrow^*_{\epsilon} x} \qquad \text{(NRl)}\quad \frac{x =^?_{\theta_0} \ell \quad [\eta_{i-1}(s_i) \rightsquigarrow^*_{\theta_i} t'_i \quad t'_i =^?_{\tau_i} \theta_i(\eta_{i-1}(t_i)]_{i=1}^n}{x \rightsquigarrow^1_{\eta_n} r}$$
for all $\ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n \in \mathcal{R}$;
x a *non-variable* term;
$\eta_0 = \theta_0$, for all $1 \leq i \leq n$, $\eta_i = \tau_i \circ \theta_i \circ \eta_{i-1}$

$$\text{(NT)}\quad \frac{x \rightsquigarrow_{\theta} y \qquad y \rightsquigarrow^*_{\theta'} z}{x \rightsquigarrow^*_{\theta' \circ \theta} z} \qquad \text{(NC)}\quad \frac{x_i \rightsquigarrow^1_{\theta} y_i}{f(x_1, \ldots, x_i, \ldots, x_k) \rightsquigarrow^1_{\theta} f(x_1, \ldots, y_i, \ldots, x_k)}$$
for all $f \in \mathcal{F}$ and $1 \leq i \leq k = arity(f)$

$$\text{(Nr)}\quad \frac{x \rightsquigarrow^1_{\theta} y}{x \rightsquigarrow_{\theta} \theta(y)}$$

Figure A.9: An inference system for conditional narrowing

such that $\sigma(u^j) \sqsupset \sigma(u^{j+1})$ for all $j \in \mathcal{J}$ (note that $1 \in \mathcal{J}$). And we have $\sigma(u^i)$ $(\gtrsim \cup \succeq \cup \sqsupset)$ $\sigma(u^{i+1})$ for all other $u^i \to v^i \Leftarrow c^i \in \mathcal{P}'$ with $i \in \mathbb{N} - \mathcal{J}$. Thus, by using the compatibility conditions of the removal triple, we obtain an infinite decreasing $\sqsupset$-sequence that contradicts the well-foundedness of $\sqsupset$.

Therefore, $\mathcal{P}' \subseteq \mathcal{P}[\emptyset]_\alpha$ and $\mathcal{Q}' \subseteq \mathcal{Q}[\emptyset]_\alpha$, which means that B is an infinite (minimal) $(\mathcal{P}[\emptyset]_\alpha, \mathcal{Q}[\emptyset]_\alpha, \mathcal{R})$-O-chain, thus leading to a contradiction. □

Theorem 76. P_{IR} *is sound and complete.*

Proof. Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}_F \uplus \mathcal{R}_I, \varphi) \in \mathsf{Dom}^{\check{}}(\mathsf{P}_{IR})$, $\alpha \in \mathcal{R}_F$ be $\mathcal{R}_F$-infeasible, $\check{\tau}' = (\mathcal{P}, \mathcal{Q}, \mathcal{R}_F[\emptyset]_\alpha \uplus (\mathcal{R}_I \cup \{\alpha\}), \varphi)$ be such that $\mathsf{P}_{IR}(\check{\tau}) = \{\check{\tau}'\}$, and $f \in \mathsf{F}$ such that $\varsigma_f(\varphi) = f$. Regarding *soundness*, since α *cannot* be used for issuing any rewriting step (with $\mathcal{P}$, $\mathcal{Q}$, or $\mathcal{R}$), every (minimal) $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain with substitution σ is a (minimal) $(\mathcal{P}, \mathcal{Q}, \mathcal{R}_F[\emptyset]_\alpha \uplus (\mathcal{R}_I \cup \{\alpha\}))$-O-chain with substitution σ. Thus, P_{IR} is sound. Completeness is similar. □

Theorem 78. P_{RIR} *is sound and* a*-complete.*

Proof. Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}_F \uplus \mathcal{R}_I, \varphi) \in \mathsf{Dom}^{\check{}}(\mathsf{P}_{RIR})$, and $\check{\tau}' = (\mathcal{P}, \mathcal{Q}, \mathcal{R}_F, \varphi)$ be such that $\mathsf{P}_{RIR}(\check{\tau}) = \{\check{\tau}'\}$, and $f \in \mathsf{F}$ such that $\varsigma_f(\varphi) = f$. *Soundness* is obvious, as every (minimal) $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain with substitution σ is a (minimal) $(\mathcal{P}, \mathcal{Q}, \mathcal{R}_F)$-O-chain with substitution σ (by definition, only rules from $\mathcal{R}_F$ are actually used in the $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain, and operationally terminating terms with respect to $\mathcal{R}$ are also operationally terminating with respect to $\mathcal{R}_F \subseteq \mathcal{R}$). Thus, P_{RIR} is sound. Regarding *completeness*, since $\mathcal{R}_F \subseteq \mathcal{R}$, every $(\mathcal{P}, \mathcal{Q}, \mathcal{R}_F)$-chain is also a $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-chain. Thus, since $\varsigma_{\mathsf{a}}(\varphi) = \mathsf{a}$, infiniteness of $\varsigma_{\mathsf{a}}(\check{\tau}')$ implies that of $\varsigma_{\mathsf{a}}(\check{\tau})$, i.e., P_{RIR} is a-complete. □

It is not difficult to see that the one-step and many-steps narrowing relations described in Definition 79 can be obtained as the ones computed by using the inference system in Figure A.9. However, in contrast to the inference system in Figure 1, only 'administrative' variables $x_1, \ldots, x_n, y_1, \ldots, y_n, x, y, z$ can be instantiated for the application of these inference rules to a goal. In particular, no variable in the rules of the CTRS (see the inference rule (NRl)) can be instantiated unless a substitution η_i, θ_i, τ_i is used. We have the following:

Proposition 94. *Let $\mathcal{R}$ be a CTRS and s,t be terms. If $s \rightsquigarrow_{\mathcal{R},\theta} t$, then $\theta(s) \to_{\mathcal{R}} t$. If $s \rightsquigarrow^*_{\mathcal{R},\theta} t$, then $\theta(s) \to^*_{\mathcal{R}} t$.*

Proof. For the goal $s \rightsquigarrow_\theta t$, there must be an application of (Nr); hence, there must be a term t' such that $t = \theta(t')$ and there is a proof of $s \rightsquigarrow^1_\theta t'$. We claim that, for all terms s and t, $s \rightsquigarrow^1_\theta t$ implies $\theta(s) \to_{\mathcal{R}} \theta(t)$, which is consistent with the use of (Nr): if $s \rightsquigarrow^1_\theta t'$ holds (and hence $\theta(s) \to_{\mathcal{R}} \theta(t)$ would hold), then $s \rightsquigarrow_\theta \theta(t')$ holds by the application of (Nr) and, since $t = \theta(t')$, we have $\theta(s) \to^*_{\mathcal{R}} t$ as required.

We proceed by multiple induction on the well-formed proof trees for $s \rightsquigarrow^1_\theta t'$ and $s \rightsquigarrow^*_\theta t$ where s is a nonvariable term and the inference system in Figure A.9 is used. The base cases are: (i) $s \rightsquigarrow^1_\theta t'$ is proved by a direct application of (NRl) with an *unconditional* rule $\ell \to r$. Then, θ is the *mgu* θ of s and ℓ, i.e., $\theta(s) = \theta(\ell)$, and $t' = r$. Thus, we obtain $\theta(s) \to_{\mathcal{R}} \theta(t') = t$ by a single application of (Repl) in Figure 1 to prove the goal $\theta(s) \to t$ with rule $\ell \to r$ and using substitution θ. (ii) $s \rightsquigarrow^*_\epsilon s$ which immediately implies $\epsilon(s) = s \to^* s$ by using (Refl) in the Inference System in Figure 1.

For the induction case, for the goal $s \rightsquigarrow^*_\theta t$, the only possibility is a single application of (NT). Then, $s \rightsquigarrow_{\theta_1} t'$ for some term t' and substitution θ_1, and $t' \rightsquigarrow^*_{\theta_2} t$ for some substitution θ_2 such that $\theta = \theta_2 \circ \theta_1$. By the induction hypothesis, $\theta_1(s) \to t'$ and $\theta_2(t') \to^* t$. By stability of conditional rewriting $\theta_2(\theta_1(s)) \to \theta_2(t')$ and by using (Tran) in Figure 1 we obtain $\theta_2(\theta_1(s)) \to^* t$, i.e., $\theta(s) \to^* t$ as required.

For the goal $s \rightsquigarrow^1_\theta t'$, we have two cases:

1. (NC) is applied. Then $s = f(s_1, \ldots, s_k)$ and $t' = f(t'_1, \ldots, t'_k)$ for some function symbol f and there is i, $1 \le i \le ar(f)$ such that $s_j = t'_j$ for all $1 \le j \le k$, $i \ne j$ and there is a proof of $s_i \rightsquigarrow^1_\theta t'_i$. By the I.H., $\theta(s_i) \to_{\mathcal{R}} \theta(t'_i)$, i.e., there is a closed proof tree for $\theta(s_i) \to \theta(t'_i)$. Now, we can use (Cong) to prove the goal $\theta(s) \to \theta(t')$, and therefore conclude $\theta(s) \to_{\mathcal{R}} \theta(t') = t$ as desired.
2. (NRl) is applied with a conditional rule $\alpha : \ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n$ to obtain a proof of $s \rightsquigarrow^1_\theta t'$, where $\theta = \eta_n$ for η_n as in the rule. Therefore, s and ℓ unify with *mgu* θ_0, i.e., $\theta_0(s) = \theta_0(\ell)$, $t' = r$, and for all $1 \le i \le n$ we have proofs of $\eta_{i-1}(s_i) \rightsquigarrow^*_{\theta_i} t'_i$ and $t'_i =^?_{\tau_i} \theta_i(\eta_{i-1}(t_i))$ (i.e., $\tau_i(t'_i) = \tau_i(\theta_i(\eta_{i-1}(t_i)))$). By the I.H., for all $1 \le i \le n$ we have $\theta_i(\eta_{i-1}(s_i)) \to^*_{\mathcal{R}} t'_i$ and by stability of rewriting, $\tau_i(\theta_i(\eta_{i-1}(s_i))) \to^*_{\mathcal{R}} \tau_i(\theta_i(\eta_{i-1}(t_i)))$. By definition of η_n which coincides with θ, $\theta(s) = \theta(\ell)$ and for all i, $1 \le i \le n$, $\theta(s_i) \to^*_{\mathcal{R}} \theta(t_i)$. Thus, the application of (Repl) with rule α and substitution θ leads to a proof of $\theta(s) \to \theta(r)$. Since, $\theta(r) = \theta(t') = t$, we have $\theta(s) \to_{\mathcal{R}} t$ as desired.

□

Theorem 88. *Let $\check{\tau} = (\mathcal{P}, \mathcal{Q}, \mathcal{R}, \varphi)$ and $\alpha : u \to v \Leftarrow c \in \mathcal{P}$ be as in Definition 85. P_{NR} is a-complete. P_{NR} is sound for $\check{\tau}$ if (i) $\mathcal{N} = \overline{\mathcal{N}}(\mathcal{R}, \alpha)$, (ii) for all $u' \to v' \Leftarrow c' \in \mathcal{P} \cup \mathcal{Q}$ (with renamed variables), either v and u' do not unify or*

v and u' unify with mgu θ *and $\theta(c)$ is $\mathcal{R}$-infeasible, and (iii) either v is ground and $\mathcal{R}$ is a deterministic 3-CTRS, or NRules$(\mathcal{R}, v)$ is a TRS and v is linear.*

PROOF. In the following, we let $\mathcal{P}' = \mathcal{P}[\mathcal{N}]_\alpha$ for readability. With regard to *soundness*, we show that any *arbitrary* $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain A where appropriately renamed versions of $\alpha : u \to v \Leftarrow c$ are used, written

$$\ldots, u^1 \to v^1 \Leftarrow c^1, u \to v \Leftarrow c, u^2 \to v^2 \Leftarrow c^2, \ldots \tag{A.4}$$

(where only the first occurrence of a use of α is displayed and by abuse it is denoted here without any explicit mark concerning the renaming of α) with substitution σ can be transformed into a $(\mathcal{P}', \mathcal{Q}, \mathcal{R})$-O-chain A'

$$\ldots, u^1 \to v^1 \Leftarrow c^1, \theta(u) \to w \Leftarrow \theta(c), u^2 \to v^2 \Leftarrow c^2, \ldots \tag{A.5}$$

for some narrowing w of v, i.e., $(w, \theta) \in N_1(\mathcal{R}, v)$, using an unconditional rule $\ell \to r \in \mathit{NRules}(\mathcal{R}, v)$. Pairs like $u \to v \Leftarrow c$ and $u^2 \to v^2 \Leftarrow c^2$ are connected in (A.4) as follows:

$$\sigma(v) = w_1(\to_\mathcal{R}^* \circ \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})\, w_2(\to_\mathcal{R}^* \circ \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}}) \cdots (\to_\mathcal{R}^* \circ \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})\, w_m \to_\mathcal{R}^* \sigma(u^2)$$

for some $m \geq 0$ and terms $w_1, \ldots, w_m$. Accordingly, we have

$$\sigma(v) \to_\mathcal{R}^* \sigma(u') \tag{A.6}$$

for some $\alpha' : u' \to v' \Leftarrow c' \in \mathcal{P} \cup \mathcal{Q}$: if $m = 0$, then α' is $u^2 \to v^2 \Leftarrow c^2 \in \mathcal{P}$; if $m > 0$, then $\alpha' \in \mathcal{Q}$. Furthermore, we assume σ to be such that the restriction of σ to variables in α makes the length of sequence (A.6) *minimal*. Note that the sequence (A.6) from $\sigma(v)$ to $\sigma(u')$ cannot be empty. Otherwise, v and u' unify, contradicting our initial assumptions. Thus, we can write (A.6) as:

$$\sigma(v) \to_\mathcal{R}^* \delta(v) \overset{p}{\to}_\mathcal{R} \delta(v[r]_p) \to_\mathcal{R}^* \sigma(u') \tag{A.7}$$

where $p \in \mathcal{P}os_\mathcal{F}(v)$ is a nonvariable position of v, δ is a substitution satisfying that, for all $x \in \mathcal{V}ar(v)$, $\sigma(x) \to_\mathcal{R}^* \delta(x)$, and there is a rule $\ell \to r \Leftarrow d \in \mathit{NRules}(\mathcal{R}, v)$ and a substitution ρ such that $\delta(v|_p) = \rho(\ell)$, $\delta(x) = \rho(x)$ for all $x \in \mathcal{V}ar(r)$; $\rho(s') \to_\mathcal{R}^* \rho(t')$ for all $s' \to t' \in d$. Otherwise, by *linearity* of v, we would have $\sigma(v) \to_\mathcal{R}^* \sigma'(v) = \sigma(u')$ for some substitution σ' such that for all $x \in \mathcal{V}ar(v)$, $\sigma(x) \to^* \sigma'(x)$. This means that v and u' unify, contradicting again our initial assumptions. Furthermore, we must have $\sigma(v) = \delta(v)$, i.e., the rewrite step at position p should be *the first one*. Otherwise, we could replace $\sigma(x) = \delta(x)$ for x in the variables of α. Hence $\delta(v^1) = \sigma(v^1)(\to_\mathcal{R}^* \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^* \sigma(u) \to_\mathcal{R}^* \delta(u)$ (by monotonicity of $\to^*$), i.e., $\delta(v^1)(\to_\mathcal{R}^* \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^* \delta(u)$. And $\delta(v) \to_\mathcal{R}^* \delta(u') = \sigma(u')$ in a shorter sequence, leading to a contradiction with our assumption for σ. Therefore, since we can assume that variables in ℓ are fresh, we can extend σ to behave like ρ on variables in ℓ; thus, (A.7) can be written as follows:

$$\sigma(v) = \sigma(v)[\sigma(v|_p)]_p = \sigma(v)[\sigma(\ell)]_p \overset{p}{\to}_\mathcal{R} \sigma(v)[\sigma(r)]_p \to_\mathcal{R}^* \sigma(u') \tag{A.8}$$

Thus, $\sigma(\ell) = \sigma(v|_p)$, i.e, ℓ and $v|_p$ unify with mgu θ and there is a substitution γ satisfying $\sigma(x) = \gamma(\theta(x))$ for all variables $x \in \mathcal{V}ar(\ell)$. Furthermore, v narrows to $\theta'(v)[\theta'(r)]_p = w$ with substitution θ' which extends θ for the variables not in ℓ; this is a consequence of our assumptions:

1. If v is not ground, we assume $\ell \to r \Leftarrow d$ to be an unconditional rule $\ell \to r$ because $NRules(\mathcal{R}, v)$ is a TRS. Hence, nothing else is necessary to justify the narrowing step $v \leadsto_{\theta'} \theta'(v[r]_p) = \theta'(v)[\theta'(r)]_p = w$ where $\theta' = \theta$.
2. If v is ground, then $\sigma(x)$ is ground for all variables $x \in \mathcal{V}ar(\ell)$ and we actually have $\sigma(x) = \theta(x) = \theta'(x)$ for all such variables. By determinism of the rules in $\mathcal{R}$, which is a 3-CTRS, during the rewriting step in (A.8) terms s in conditions $s \to t$ in d are instantiated to ground terms before any evaluation with $\to^*_{\mathcal{R}}$. Actually, for all such conditions we have $\sigma(s) \to^*_{\mathcal{R}} \sigma(t)$ with $\sigma(s)$ and $\sigma(t)$ being ground terms and σ is a ground substitution (for the non-trivial bindings). Therefore, if we let $\theta' = \sigma$, we have $\theta'(s) \leadsto^*_{\theta'} \theta'(t)$ in all these cases. Hence, we also have the desired narrowing step[8] $v \leadsto_{\theta'} v[\theta'(r)]_p = \theta'(v)[\theta'(r)]_p = w$.

Again, we can extend σ to behave like γ on the variables of $\theta'(u)$ and w. Therefore, we have $\sigma(v^1)(\to^*_{\mathcal{R}} \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^*\sigma(u) = \gamma(\theta'(u)) = \sigma(\theta'(u))$, and $\sigma(s) = \gamma(\theta'(s)) \to^* \gamma(\theta'(t)) = \sigma(t)$ for all $s \to t \in c$, and

$$\begin{array}{rcll} \sigma(w) = \gamma(w) & = & \gamma(\theta'(v))[\gamma(\theta'(r))]_p \\ & = & \sigma(v)[\sigma(r)]_p \\ & = & \sigma(v)[\rho(r)]_p \\ & (\to^*_{\mathcal{R}} \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^* & \sigma(u^2) \end{array}$$

Thus, A' is a $(\mathcal{P}', \mathcal{Q}, \mathcal{R})$-chain. Now, with regard to minimality, since termination of $\sigma(v)$ implies termination of q, if A is a minimal $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-O-chain, then A' is also a minimal $(\mathcal{P}', \mathcal{Q}, \mathcal{R})$-O-chain.

With regard to *completeness*, let A' be an infinite $(\mathcal{P}', \mathcal{Q}, \mathcal{R})$-chain

$$\ldots, u^1 \to v^1 \Leftarrow c^1, \theta(u) \to w \Leftarrow \theta(c), u^2 \to v^2, \ldots$$

where w is obtained from v by some narrowing step with $\alpha : \ell \to r \Leftarrow d \in NRules(\mathcal{R}, v)$ with substitution θ (which, of course, can be different for each renamed version of α which is used in A'). We prove that A, i.e.,

$$\ldots, u^1 \to v^1 \Leftarrow c^1, u \to v \Leftarrow c, u^2 \to v^2, \ldots$$

(where the occurrences of $u \to v \Leftarrow c$ are appropriately renamed) is an infinite $(\mathcal{P}, \mathcal{Q}, \mathcal{R})$-chain. There is a substitution σ such that $\sigma(v^1)(\to^*_{\mathcal{R}} \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^*\sigma(\theta(u))$, $\sigma(\theta(s)) \to^* \sigma(\theta(t))$ for all $s \to t \in c$, $\sigma(\theta(s')) \to^* \sigma(\theta(t'))$ for all $s' \to t' \in d$, and $\sigma(w)(\to^*_{\mathcal{R}} \cup \overset{\Lambda}{\to}_{\mathcal{Q},\mathcal{R}})^*\sigma(u_2)$. Since the variables in the pairs are

[8] The following 3-CTRS $\mathcal{R} = \{\mathsf{b} \to \mathsf{a}, \mathsf{a} \to x \Leftarrow x \to \mathsf{a}, x \to \mathsf{b}\}$ shows that this does not hold without determinism of $\mathcal{R}$: we have $\mathsf{a} \to \mathsf{b}$ with $\sigma(x) = \mathsf{b}$, but a is *not* narrowable.

pairwise disjoint, we may extend σ to behave like $\sigma(\theta(x))$ on $x \in \mathcal{V}ar(u)$. Then $\sigma(u) = \sigma(\theta(u))$, $\sigma(v_1)(\to_{\mathcal{R}}^* \cup \xrightarrow{\Lambda}_{\mathcal{Q},\mathcal{R}})^*\sigma(u)$, $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ for all $s \to t \in c$, and $\sigma(s') \to_{\mathcal{R}} \sigma(t')$ for all $s' \to t' \in d$. By Proposition 94, we have $\theta(v) \to_{\mathcal{R}} w$. This implies that $\sigma(\theta(v)) \to_{\mathcal{R}} \sigma(w)$, and since $\sigma(v) = \sigma(\theta(v))$, we get $\sigma(v) \to_{\mathcal{R}} \sigma(w)(\to_{\mathcal{R}}^* \cup \xrightarrow{\Lambda}_{\mathcal{Q},\mathcal{R}})^*\sigma(u_2)$. □

Theorem 91. P_{NQ} *is* a*-complete.* P_{NQ} *is sound if* $\mathcal{N}_\Lambda(\mathcal{Q},\alpha) = \overline{\mathcal{N}}_\Lambda(\mathcal{Q},\alpha)$, $NRules(\mathcal{R}, v) = \emptyset$*, for all* $u' \to v' \Leftarrow c' \in \mathcal{P}$ *(with renamed variables),* v *and* u' *do not unify or* v *and* u' *unify with* mgu θ *and* $\theta(c)$ *is* $\mathcal{R}$*-infeasible, and either* v *is ground and* $\mathcal{R}$ *is a deterministic* 3*-CTRS or* $NRules^{\Lambda}(\mathcal{Q}, v)$ *is a TRS and* v *is linear.*

Proof. Let $\mathcal{P}' = \mathcal{P}[\mathcal{N}_\Lambda(\mathcal{Q},\alpha)]_\alpha$. For soundness, as in Theorem 88 we proceed by contradiction and prove that for any infinite $(\mathcal{P},\mathcal{Q},\mathcal{R})$-chain A, where appropriately renamed versions of $\alpha : u \to v \Leftarrow c$ are used, written:

$$\ldots, u^1 \to v^1 \Leftarrow c^1, u \to v \Leftarrow c, u^2 \to v^2 \Leftarrow c^2, \ldots,$$

there is an infinite $(\mathcal{P}',\mathcal{Q},\mathcal{R})$-chain A':

$$\ldots, u^1 \to v^1 \Leftarrow c^1, \theta(u) \to w \Leftarrow \theta(c), u^2 \to v^2 \Leftarrow c^2, \ldots,$$

where $\theta(u) \to w \Leftarrow \theta(c) \in \mathcal{P}'$. Since A is a $(\mathcal{P},\mathcal{Q},\mathcal{R})$-chain, there is a substitution σ such that $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ for all $s \to t \in c$, and $\sigma(v)(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}_{\overline{\mathcal{Q}}}^{=})^*\sigma(u^2)$. The length of the sequence $\sigma(v)(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}_{\overline{\mathcal{Q}}}^{=})^*\sigma(u^2)$ is greater than zero, because v and u^2 do not unify, that is, $\sigma(v) \neq \sigma(u^2)$. Hence, $\sigma(v)(\to_{\mathcal{R}} \cup \xrightarrow{\Lambda}_{\overline{\mathcal{Q}}}^{=})q(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}_{\overline{\mathcal{Q}}}^{=})^*\sigma(u^2)$ for some term q. Furthermore, since $NRules(\mathcal{R}, v) = \emptyset$, we have $\sigma(v) \xrightarrow{\Lambda}_{\mathcal{Q}} q(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}_{\overline{\mathcal{Q}}}^{=})^*\sigma(u^2)$, i.e., $\sigma(v) = \sigma(u')$ for some $u' \to v' \Leftarrow c \in \mathcal{Q}$. Hence, v is a narrex of $u' \to v' \Leftarrow c \in NRules^{\Lambda}(\mathcal{Q}, v)$ and if v is not ground we can assume that such a rule is unconditional. Furthermore, u' and v unify and there is a mgu θ and a substitution γ satisfying $\sigma(x) = \gamma(\theta(x))$ for all variables x. Therefore, $\sigma(v^1)(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}_{\overline{\mathcal{Q}}}^{=})^*\sigma(u) = \gamma(\theta(u)) = \sigma(\theta(u))$, $\sigma(s) = \gamma(\theta(s)) \to_{\mathcal{R}}^* \gamma(\theta(t)) = \sigma(t)$ for all $s \to t \in c$, and $w = \sigma(v')(\to_{\mathcal{R}}^* \circ \xrightarrow{\Lambda}_{\overline{\mathcal{Q}}}^{=})^*\sigma(u^2)$. Hence, A' is a $(\mathcal{P}',\mathcal{Q},\mathcal{R})$-chain. Completeness is similar to P_{NR}. □