Confluence of Conditional Rewriting in Logic Form

Raúl Gutiérrez
Universidad Politécnica de Madrid, Spain

Salvador Lucas
DSIC & VRAIN, Universitat Politècnica de València, Spain

Miguel Vítores
VRAIN, Universitat Politècnica de València, Spain

Abstract

We characterize conditional rewriting as satisfiability in a Herbrand-like model of terms where variables are also included as fresh constant symbols extending the original signature. Confluence of conditional rewriting and joinability of conditional critical pairs is characterized similarly. Joinability of critical pairs is then translated into combinations of *(in)feasibility* problems which can be efficiently handled by a number of automatic tools. This permits a more efficient use of standard results for proving confluence of conditional term rewriting systems, most of them relying on auxiliary proofs of joinability of conditional critical pairs, perhaps with additional syntactical and (operational) termination requirements on the system. Our approach has been implemented in a new system: CONFident. Its ability to (dis)prove confluence of conditional term rewriting systems is witnessed by means of some benchmarks comparing our tool with existing tools for similar purposes.

2012 ACM Subject Classification Theory of computation → Automated reasoning; Theory of computation → Logic and verification; Theory of computation → Equational logic and rewriting

Keywords and phrases Confluence, Program analysis, Rewriting-based systems

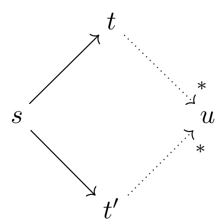
Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.44

Supplementary Material *Software (Online Tool)*: `http://zenon.dsic.upv.es/confident/`

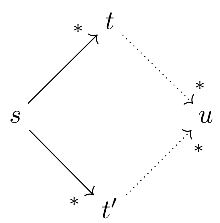
Funding Partially supported by grant RTI2018-094403-B-C32 funded by MCIN/AEI/10.13039 /501100011033 and by "ERDF A way of making Europe", and PROMETEO/2019/098.

1 Introduction

Confluence is a property of (abstract) reduction relations $\to$ guaranteeing that, for all abstract objects s (often called *expressions* without loss of generality) which can be reduced into two different reducts t and t', respectively (written $s \to^* t$ and $s \to^* t'$), there is another expression u to which both t and t' are reducible, i.e., both $t \to^* u$ and $t' \to^* u$ hold. A weaker property is *local* confluence, where only a *single* reduction step is allowed on s, i.e., $s \to t$ and $s \to t'$. As usual, they are defined by the commmutation of the diagrams:



Local confluence



confluence

These two properties of abstract reduction relations are connected by the well-known *Newman's Lemma*: if $\to$ is terminating (i.e., there is no infinite reduction sequence $t_1 \to t_2 \to \cdots$), then local confluence and confluence coincide (see, e.g., [23, Lemma 2.2.5]). Now, the following issues naturally arise: (i) How to define $\to$ from the specification of a program of a (rule-based) formalism (e.g., (conditional) term rewriting [3]) or programming language? (ii) How to prove/disprove (local) confluence of such a reduction relation? (iii) How to automate the proofs? In this paper we address these problems.

Regarding (i), we use a logical approach to define reduction relations. Given a specification $\mathcal{R}$, we obtain an inference system $\mathcal{I}(\mathcal{R})$ out from the generic description of the operational semantics of the underlying formalism or language. Then, $\to$ and $\to^*$ are defined by satisfiability of atoms $s \to t$ and $s \to^* t$ in a canonical model $\mathcal{M}_{\mathcal{R}}$ which is the Herbrand model (in an "extended" Herbrand universe where variables are treated as constants) of the atoms that can be proved using $\mathcal{I}(\mathcal{R})$. This general approach applies to many computational systems and programming languages, in particular to conditional term rewriting systems (CTRS, see, e.g., [23, Chapter 7]), and Maude [5]. In Section 3, we develop this approach with a particular focus on CTRSs (to keep things simpler). Most ideas, though, can be easily generalized. Regarding (ii), we represent confluence properties above in logic form, e.g.,

$$\begin{array}{rlll} \varphi_{\mathrm{WCR}} & (\forall x)(\forall y)(\forall z)(\exists u) & x \to y \wedge x \to z \Rightarrow y \to^* u \wedge z \to^* u & \text{Local confluence} \\ \varphi_{\mathrm{CR}} & (\forall x)(\forall y)(\forall z)(\exists u) & x \to^* y \wedge x \to^* z \Rightarrow y \to^* u \wedge z \to^* u & \text{Confluence} \end{array}$$

In Section 4, we show that (local) confluence is characterized as *satisfiability* in $\mathcal{M}_{\mathcal{R}}$, i.e., $\mathcal{R}$ is (locally) confluent iff $\mathcal{M}_{\mathcal{R}} \models \varphi_{\mathrm{CR}}$ (resp. $\mathcal{M}_{\mathcal{R}} \models \varphi_{\mathrm{WCR}}$) holds. Regarding (iii), in Section 6 we show how to translate confluence problems into combinations of *(in)feasibility* problems [11]. In this setting, automated proofs are possible by using several techniques and tools developed so far, see [21] for a summary of techniques and tools in this respect. Section 7 shows how these techniques are used to prove and disprove confluence of CTRSs. We have implemented our results as part of the new tool CONFident, which can be found here:

`http://zenon.dsic.upv.es/confident/`

Section 8 provides some details of its implementation and use. The good results of the aforementioned techniques are witnessed by our participation in the 2021 edition of the Confluence Competition (CoCo 2021) on which we report at the end of the section. Section 9 discusses some related work. Section 10 concludes. Proofs of technical results are given in an appendix.

2 Preliminaries

Given a binary relation $\mathsf{R} \subseteq A \times A$ on a set A, we often write $a \,\mathsf{R}\, b$ instead of $(a, b) \in \mathsf{R}$. The *transitive* closure of R is denoted by R^+, and its *reflexive and transitive* closure by R^*. An element $a \in A$ is *irreducible* (or an R-*normal form*), if there exists no b such that $a \,\mathsf{R}\, b$. Given $a \in A$, if there is no infinite sequence $a = a_1 \ \mathsf{R}\ a_2 \ \mathsf{R}\ \cdots\ \mathsf{R}\ a_n \ \mathsf{R} \cdots$, then a is R-*terminating* (or *well-founded*); also, R is said *terminating* if a is R-terminating for all $a \in A$. We say that R is (locally) *confluent* if, for every $a, b, c \in A$, whenever $a\ \mathsf{R}^* b$ and $a\ \mathsf{R}^* c$ (resp. $a\ \mathsf{R}\ b$ and $a\ \mathsf{R}\ c$), there exists $d \in A$ such that $b\ \mathsf{R}^* d$ and $c\ \mathsf{R}^* d$.

We use the standard notations in term rewriting (see, e.g., [23]). In this paper, $\mathcal{X}$ denotes a countable set of *variables* and $\mathcal{F}$ denotes a *signature*, i.e., a set of *function symbols* $\{f, g, \ldots\}$ (disjoint from $\mathcal{X}$), each with 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 $\mathcal{T}(\mathcal{F}, \mathcal{X})$. The set of *ground* terms (i.e., terms without variable occurrences) is denoted $\mathcal{T}(\mathcal{F})$. The set of variables occurring in t is $\mathcal{V}ar(t)$. By abuse of notation, we use $\mathcal{V}ar$ also with sequences of terms or other expressions to denote the set of variables occurring in them. Terms are viewed as labeled trees in the usual way. *Positions*

$p, q, \ldots$ are represented by chains of positive natural numbers used to address subterms $t|_p$ of t. The *set of positions* of a term t is $\mathcal{P}os(t)$. A substitution is a mapping from variables into terms which is homomorphically extended to a mapping from terms to terms.

A *conditional rule* (with label α) is written $\alpha : \ell \to r \Leftarrow C$, where $\ell \in \mathcal{T}(\mathcal{F}, \mathcal{X}) - \mathcal{X}$ and $r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ are called the *left-* and *right-hand sides* of the rule, respectively, and the *conditional part* C is a sequence $s_1 \approx t_1, \cdots, s_n \approx t_n$ with $s_1, t_1, \ldots, s_n, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ for some $n \geq 0$. The case $n = 0$ corresponds to an *empty* conditional part. A Conditional Term Rerwiting System (CTRS) $\mathcal{R}$ is a set of conditional rules; if all rules $\ell \to r \Leftarrow C$ in $\mathcal{R}$ have an empty conditional part and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)$ holds, then $\mathcal{R}$ is called a Term Rewriting System (TRS).

3 Term Rewriting as Satisfiability

In term rewriting variables occurring in terms t_i in reduction sequences $t_1 \to t_2 \to \cdots \to$ are treated *as constants* in the sense that they are not instantiated in any way. This is in contrast with variables occurring in rules of TRSs which are instantiated to implement reduction steps by means of matching substitutions. In the following we provide a formal presentation of this fact which permits the definition of a canonical model $\mathcal{M}_\mathcal{R}$ which captures the reduction of terms with variables, in contrast to the usual (ground) models developed elsewhere (e.g., [6]) which are better suited to capture *ground rewriting*, i.e., rewriting of *ground* terms.

▶ Remark 1 (Confluence and ground confluence). In general, confluence and *ground* confluence (i.e., confluence of $\to$ when restricted to ground terms) of (C)TRSs do *not* coincide. For instance, the TRS $\mathcal{R} = \{f(x) \to a, f(x) \to x\}$ over the signature $\mathcal{F} = \{a, f\}$ is ground confluent, but not confluent. If a new constant b is added to $\mathcal{F}$, then $\mathcal{R}$ is not ground confluent anymore.

In Section 4 we use $\mathcal{M}_\mathcal{R}$ to provide a characterization of confluence properties as satisfiability in $\mathcal{M}_\mathcal{R}$. In the following, as anticipated by the expression of (local) confluence using first-order formulas $\varphi_{\rm CR}$ and $\varphi_{\rm WCR}$, we view term rewriting from a logical point of view. A first-order language with function symbols $f, g, \ldots$ from a signature $\mathcal{F}$ and predicate symbols $P, Q, \ldots$ from a signature Π is considered where atoms and formulas are built in the usual way. The pair $\mathcal{F}, \Pi$ is often called a *signature with predicates* [9]. In particular, rewriting expressions $s \to t$ (one-step reduction), $s \to^* t$ (zero or many-step reduction), $s \downarrow t$ (joinability), etc., are viewed as *atoms* with (binary) predicate symbols $\to$, $\to^*$, $\downarrow$, etc.

3.1 Operational Semantics of Conditional Rewriting in Logic Form

Conditions $s \approx t$ in conditional rules admit several *semantics*, i.e., forms to evaluate them see, e.g., [23, Definition 7.1.3]. *Oriented* CTRSs are those whose conditions $s \approx t$ are handled as *reachability* tests. *Join* CTRSs use *joinability* tests instead. *Semiequational* CTRSs use *convertibility* tests. For oriented CTRSs $\mathcal{R}$, an inference system $\mathcal{I}_O(\mathcal{R})$ is obtained from the following generic inference system $\mathfrak{I}_{\text{O-CTRS}}$:

$$\text{(Rf)}\quad \frac{}{x \to^* x} \qquad \text{(C)}_{f,i}\quad \frac{x_i \to y_i}{f(x_1, \ldots, x_i, \ldots, x_k) \to f(x_1, \ldots, y_i, \ldots, x_k)} \text{ for all } f \in \mathcal{F} \text{ and } 1 \leq i \leq k$$

$$\text{(T)}\quad \frac{x \to y \qquad y \to^* z}{x \to^* z} \qquad \text{(Rl)}_\alpha\quad \frac{s_1 \to^* t_1 \quad \cdots \quad s_n \to^* t_n}{\ell \to r} \text{ for } \alpha : \ell \to r \Leftarrow s_1 \approx t_1, \ldots, s_n \approx t_n \in \mathcal{R}$$

$$(\forall x)\ x \to^* x \tag{4}$$
$$(\forall x, y, z)\ x \to y \wedge y \to^* z \Rightarrow x \to^* z \tag{5}$$
$$(\forall x, y, z)\ x \to y \Rightarrow f(x, z) \to f(y, z) \tag{6}$$
$$(\forall x, y, z)\ x \to y \Rightarrow f(z, x) \to f(z, y) \tag{7}$$
$$a \to b \tag{8}$$
$$(\forall x)\ f(x, a) \to^* f(b, b) \Rightarrow f(c, x) \to x \tag{9}$$
$$(\forall y)\ (y, y) \to b \tag{10}$$

Figure 1 Theory $\overline{\mathcal{R}}_O$ for the oriented semantics of $\mathcal{R}$ in Example 3.

by *specializing* $(\mathrm{C})_{f,i}$ for each k-ary symbol f in the signature $\mathcal{F}$ and $1 \leq i \leq k$ and $(\mathrm{Rl})_\alpha$ for all conditional rules $\alpha : \ell \to r \Leftarrow C$ in $\mathcal{R}$ [14, Section 4.5]. Inference rules in $\mathcal{I}_O(\mathcal{R})$ are *schematic*: each inference rule $\frac{B_1 \ \cdots \ B_n}{A}$ in $\mathcal{I}_O(\mathcal{R})$ can be used under any *instance* $\frac{\sigma(B_1) \ \cdots \ \sigma(B_n)}{\sigma(A)}$ of the rule by a substitution σ. For *join* CTRSs, we replace rule $(\mathrm{Rl})_\alpha$ by

$$(\mathrm{Rl})^J_\alpha \quad \frac{s_1 \to^* z_1 \quad t_1 \to^* z_1 \quad \cdots \quad s_n \to^* z_n \quad t_n \to^* z_n}{\ell \to r}$$

where $z_1, \ldots, z_n$ do not occur in ℓ, r, s_i, t_i for $1 \leq i \leq n$. In this way, we obtain $\mathfrak{I}_{\text{J-CTRS}}$ and $\mathcal{I}_J(\mathcal{R})$ from $\mathfrak{I}_{\text{J-CTRS}}$ as before. Note that the joinability predicate $\downarrow$ is not necessary.

▶ Remark 2 (Semi-equational CTRSs). For semi-equational CTRSs we would proceed similarly, defining a new rule $(\mathrm{Rl})^{SE}_\alpha$ borrowing $(\mathrm{Rl})_\alpha$ where $\leftrightarrow^*$ is used instead of $\to^*$, and adding more inference rules to deal with $\leftrightarrow^*$: first $\frac{x \to y}{x \leftrightarrow y}$, also $\frac{y \to x}{x \leftrightarrow y}$, and then $\frac{x \leftrightarrow y \quad y \leftrightarrow^* z}{x \leftrightarrow^* z}$.

We obtain a theory $\overline{\mathcal{R}}_O$ (resp. $\overline{\mathcal{R}}_J$, etc.) from $\mathcal{I}_O(\mathcal{R})$ (resp. $\mathcal{I}_J(\mathcal{R})$, etc.) as follows [14, Section 4.5]: the inference rules $(\rho)\frac{B_1 \ \cdots \ B_n}{A}$ in $\mathcal{I}(\mathcal{R})$ are considered as *sentences* $\overline{\rho}$ of the form $(\forall \vec{x})\, B_1 \wedge \cdots \wedge B_n \Rightarrow A$, where $\vec{x}$ is the sequence of variables occurring in atoms $B_1, \ldots, B_n$ and A; if empty, we just write $B_1 \wedge \cdots \wedge B_n \Rightarrow A$.

▶ **Example 3.** Consider the CTRS $\mathcal{R}$

$$a \to b \tag{1}$$
$$f(c, x) \to x \Leftarrow f(x, a) \approx f(b, b) \tag{2}$$
$$f(y, y) \to b \tag{3}$$

The theory $\overline{\mathcal{R}}_O$ can be found in Figure 1. Note that this gives $\mathcal{R}$ the computational semantics of an *oriented* CTRS. Also, $\overline{\mathcal{R}}_J = \{(4), (5), (6), (7), (8), (10), (11)\}$ for

$$(\forall x)(\forall z)\ f(x, a) \to^* z \wedge f(b, b) \to^* z \Rightarrow f(c, x) \to x \tag{11}$$

i.e., we use (11) instead of (9). We usually just write $\overline{\mathcal{R}}$ to denote the (appropriate) theory associated to a (join, oriented,...) CTRS. In the following, given a first-order theory Th and a formula φ, $\mathsf{Th} \vdash \varphi$ means that φ is *deducible* from (or a *logical consequence* of) Th.

For all terms s, t, we write (i) $s \to_\mathcal{R} t$ (resp. $s \to^*_\mathcal{R} t$) iff there is a (well-formed)[1] proof tree for $s \to t$ (resp. $s \to^* t$) using $\mathcal{I}(\mathcal{R})$. Equivalently, we have (ii) $s \to_\mathcal{R} t$ (resp. $s \to^*_\mathcal{R} t$) iff $\overline{\mathcal{R}} \vdash s \to t$ (resp. $\overline{\mathcal{R}} \vdash s \to^* t$) holds. The first presentation (i) is well-suited for the analysis of the termination behavior of CTRSs: we say that $\mathcal{R}$ is *operationally terminating* if there is no (well-formed) infinite proof trees for goals $s \to t$ and $s \to^* t$ in $\mathcal{I}(\mathcal{R})$ [16]. However, the proof theoretic presentation (ii) is more important in the analysis of (in)feasibility of rewriting goals in Section 4. It also suffices to

[1] By a *well-formed* proof tree we mean a proof tree where proof conditions introduced by inference rules are developed from left to right, see [16].

define *termination* of CTRSs: a CTRS $\mathcal{R}$ is terminating if $\to_{\mathcal{R}}$ is terminating. Termination and operational termination of CTRSs differ, see [17, Section 3] for a deeper discussion about differences and connections between both notions.

We use termination and operational termination in some confluence results for CTRSs (Section 7). The tool MU-TERM [12] can be used for automatically proving and disproving termination and operational termination of CTRSs.[2]

▶ **Definition 4** (Joinable terms). *Given a CTRS $\mathcal{R}$ and terms s, t, we write $s \downarrow_{\mathcal{R}} t$ if and only if there is a term u such that $s \to^*_{\mathcal{R}} u$ and $t \to^*_{\mathcal{R}} u$. We often say that s and t are* joinable.

3.2 Dealing With Variables in Terms as (Fresh) Constants

Let $\mathcal{F}$ be a signature and $\mathcal{X}$ be a set of variables such that $\mathcal{F} \cap \mathcal{X} = \emptyset$. We let $\mathcal{F}_{\mathcal{X}} = \mathcal{F} \cup C_{\mathcal{X}}$ where variables $x \in \mathcal{X}$ are considered (different) *constant* symbols c_x of $C_{\mathcal{X}} = \{c_x \mid x \in \mathcal{X}\}$ and $\mathcal{F}$ and $C_{\mathcal{X}}$ are disjoint. Note that the set $\mathcal{T}(\mathcal{F}, \mathcal{X})$ of terms with variables for the signature $\mathcal{F}$ is isomorphic to the set $\mathcal{T}(\mathcal{F}_{\mathcal{X}})$ of ground terms for $\mathcal{F}_{\mathcal{X}}$: given a term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $t^{\downarrow} \in \mathcal{T}(\mathcal{F}_{\mathcal{X}})$ is obtained by replacing each occurrence of $x \in \mathcal{X}$ in t by c_x.[3] Vice versa: given $t \in \mathcal{T}(\mathcal{F}_{\mathcal{X}})$, $t^{\uparrow} \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is obtained by replacing, for all $x \in \mathcal{X}$, each constant c_x in t by x.

▶ **Proposition 5.** *For all terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $(t^{\downarrow})^{\uparrow} = t$. For all terms $t \in \mathcal{T}(\mathcal{F}_{\mathcal{X}})$, $(t^{\uparrow})^{\downarrow} = t$.*

Also, given a substitution $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$, define $\sigma^{\downarrow} : \mathcal{X} \to \mathcal{T}(\mathcal{F}_{\mathcal{X}})$ to be $\sigma^{\downarrow}(x) = \sigma(x)^{\downarrow}$ for all $x \in \mathcal{X}$ (given $\theta : \mathcal{X} \to \mathcal{T}(\mathcal{F}_{\mathcal{X}})$, define $\theta^{\uparrow} : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ similarly). The following result shows that rewriting with terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ can be simulated as ground rewriting in $\mathcal{T}(\mathcal{F}_{\mathcal{X}})$.

▶ **Proposition 6.** *Let $\mathcal{R} = (\mathcal{F}, R)$ be a CTRS and $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then, $s \to_{\mathcal{R}} t$ if and only if $s^{\downarrow} \to_{\mathcal{R}} t^{\downarrow}$ and $s \to^*_{\mathcal{R}} t$ if and only if $s^{\downarrow} \to^*_{\mathcal{R}} t^{\downarrow}$.*

In the following, given a condition C, i.e., $s_1 \approx t_1, \ldots, s_n \approx t_n$, we write $C^{\downarrow}$ to denote $s_1^{\downarrow} \approx t_1^{\downarrow}, \ldots, s_n^{\downarrow} \approx t_n^{\downarrow}$.

3.3 A Ground Model for Rewriting Terms with Variables

Given a signature with predicates $\mathcal{F}, \Pi$, an $\mathcal{F}, \Pi$-*structure* $\mathcal{A}$ (or just *structure* if $\mathcal{F}, \Pi$ is clear from the context) consists of a *domain* (also denoted) $\mathcal{A}$ together with an interpretation of the function symbols $f \in \mathcal{F}$ and predicate symbols $P \in \Pi$ as mappings $f^{\mathcal{A}}$ and relations $P^{\mathcal{A}}$ on $\mathcal{A}$, respectively. Then, the usual interpretation of first-order formulas with respect to $\mathcal{A}$ is considered [20, page 60]. An $\mathcal{F}, \Pi$-model for a theory Th, i.e., a set of first-order sentences (formulas whose variables are all *quantified*), is just a structure $\mathcal{A}$ that makes them all true, written $\mathcal{A} \models \mathsf{Th}$. A formula φ is a *logical consequence* of a theory Th (written $\mathsf{Th} \models \varphi$) iff every model $\mathcal{A}$ of Th is also a model of φ. The *canonical model* $\mathcal{M}_{\mathcal{R}}$ of a CTRS $\mathcal{R}$ is defined as follows.

▶ **Definition 7** (Canonical model for conditional rewriting). *Let $\mathcal{R}$ be a CTRS. The* canonical model *$\mathcal{M}_{\mathcal{R}}$ of $\mathcal{R}$ has domain $\mathcal{T}(\mathcal{F}_{\mathcal{X}})$; each k-ary symbol $f \in \mathcal{F}$ is interpreted as $f^{\mathcal{M}_{\mathcal{R}}}(t_1, \ldots, t_k) = f(t_1, \ldots, t_k)$ for all $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}_{\mathcal{X}})$. Finally, predicate symbols $\to$ and $\to^*$ are interpreted as follows:*

$$\to^{\mathcal{M}_{\mathcal{R}}} = \{(s^{\downarrow}, t^{\downarrow}) \mid s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \wedge s \to_{\mathcal{R}} t\} \qquad (\to^*)^{\mathcal{M}_{\mathcal{R}}} = \{(s^{\downarrow}, t^{\downarrow}) \mid s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \wedge s \to^*_{\mathcal{R}} t\}$$

[2] Although the version of MU-TERM described in [12] did not allow proofs of termination of CTRSs, for the purpose of serving as a backbone for CONFident, we recently modified MU-TERM as to provide explicit use of the techniques described in [18], which can be used to prove and disprove termination of CTRSs. Thus, MU-TERM users can prove and disprove termination of CTRSs by following the instructions in `http://zenon.dsic.upv.es/muterm/?name=documentation#CTRSs`.

[3] We use $\downarrow$ as superindex denoting this *grounding* operation as in $t^{\downarrow}$, hopefully not leading to confusion with the infix use of $\downarrow$ as joinability operator, as in $s \downarrow t$.

Definition 7 generalizes to accomodate interpretations for $\leftrightarrow$ and $\leftrightarrow^*$ in semi-equational CTRSs in the obvious way.[4]

▶ **Theorem 8.** *For all CTRSs $\mathcal{R}$, $\mathcal{M}_\mathcal{R} \models \overline{\mathcal{R}}$.*

We have the following:

▶ **Proposition 9.** *Let $\mathcal{R} = (\mathcal{F}, R)$ be a CTRS, $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, and $\vec{x} = x_1, \ldots, x_n$ denote the variables occurring in s and t, i.e., $\mathcal{V}ar(s) \cup \mathcal{V}ar(t) = \{x_1, \ldots, x_n\}$. Then,*

1. *We have that $\sigma(s) \to^*_\mathcal{R} \sigma(t)$ for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$, if and only if $(s^\downarrow, t^\downarrow) \in (\to^*)^{\mathcal{M}_\mathcal{R}}$.*
2. *$\mathcal{M}_\mathcal{R} \models (\forall \vec{x})\, s \to^* t$ if and only if $(s^\downarrow, t^\downarrow) \in (\to^*)^{\mathcal{M}_\mathcal{R}}$.*

Proposition 9 shows that we can *remove* universal quantifiers from reachability formulas if variables x in the involved terms are replaced by the corresponding constants c_x.

4 Confluence of Rewriting as a Satisfiability Problem

In view of Section 3.1, it is perhaps natural to adopt a proof theoretical definition of (local) confluence of CTRSs as follows: *a CTRS is (locally) confluent if and only if $\overline{\mathcal{R}} \vdash \varphi_{CR}$ (resp. $\overline{\mathcal{R}} \vdash \varphi_{WCR}$) holds.* The following example (using a TRS) shows that this is *not* equivalent to the usual definition.

▶ **Example 10.** A well-known example of a *locally confluent* but *nonconfluent* TRS is $\mathcal{R} = \{\mathsf{b} \to \mathsf{a}, \mathsf{b} \to \mathsf{c}, \mathsf{c} \to \mathsf{b}, \mathsf{c} \to \mathsf{d}\}$. The theory $\overline{\mathcal{R}}$ for $\mathcal{R}$ is

$$\begin{array}{rclcrclcrcl} (\forall x)\, x & \to^* & x & \quad & \mathsf{b} & \to & \mathsf{a} & \quad & \mathsf{c} & \to & \mathsf{b} \\ (\forall x, y, z)\, x \to y \wedge y \to^* z \Rightarrow x & \to^* & z & & \mathsf{b} & \to & \mathsf{c} & & \mathsf{c} & \to & \mathsf{d} \end{array}$$

Unfortunately, φ_{WCR} is *not* a logical consequence of $\overline{\mathcal{R}}$ (i.e., $\overline{\mathcal{R}} \models \varphi_{\mathrm{WCR}}$ does *not* hold) and hence[5] it *cannot* be proved from $\overline{\mathcal{R}}$ (i.e., $\overline{\mathcal{R}} \vdash \varphi_{\mathrm{WCR}}$ does not hold): there is a model $\mathcal{A}$ of $\overline{\mathcal{R}}$ which is *not* a model of φ_{WCR}. The interpretation domain is $\mathcal{A} = \{0, 1, 2, 3, 4\}$, function symbols are interpreted by: $\mathsf{a}^\mathcal{A} = 0$, $\mathsf{b}^\mathcal{A} = 1$, $\mathsf{c}^\mathcal{A} = 2$, $\mathsf{d}^\mathcal{A} = 3$, and predicate symbols by

$$\begin{array}{rcl} \to^\mathcal{A} & = & \{(1,0), (1,2), (2,1), (2,3), (4,0), (4,3)\} \\ (\to^*)^\mathcal{A} & = & \{(1,0), (1,2), (2,1), (2,3), (4,0), (4,3)\} \cup \{(0,0), (1,1), (2,2), (3,3), (4,4)\} \cup \{(2,0), (1,3)\} \end{array}$$

Although $\mathcal{A} \models \overline{\mathcal{R}}$ holds, with the valuation α given by $\alpha(x) = 4$, $\alpha(y) = 0$ and $\alpha(z) = 3$, $[x \to y \wedge x \to z]^\mathcal{A}_\alpha$ holds true, but $[y \to^* u \wedge z \to^* u]^\mathcal{A}_\alpha$ is false for all valuations of u. Thus, $\mathcal{A} \models \varphi_{WCR}$ does *not* hold. Hence $\overline{\mathcal{R}} \models \varphi_{WCR}$ does *not* hold either.

Instead, we use $\mathcal{M}_\mathcal{R}$ to define (local) confluence as *satisfiability* in $\mathcal{M}_\mathcal{R}$.

▶ **Theorem 11** (Confluence of CTRSs as satisfiability in $\mathcal{M}_\mathcal{R}$). *A CTRS $\mathcal{R}$ is (locally) confluent if and only if $\mathcal{M}_\mathcal{R} \models \varphi_{CR}$ (resp. $\mathcal{M}_\mathcal{R} \models \varphi_{WCR}$) holds.*

Now, as a consequence of [14, Corollary 14] and Theorem 11, we have the following:

▶ **Corollary 12.** *Let $\mathcal{R}$ be a CTRS. If $\mathcal{M}_\mathcal{R} \vdash \varphi_{CR}$ (resp. $\mathcal{M}_\mathcal{R} \vdash \varphi_{WCR}$) holds, then $\mathcal{R}$ is (locally) confluent.*

Example 10 shows that the statement in Corollary 12 cannot be reversed.

---

[4] The theory $\overline{\mathcal{R}}_J$ associated to a join CTRS $\mathcal{R}$ uses predicates $\to$ and $\to^*$ only. Hence, no change in the definition of $\mathcal{M}_\mathcal{R}$ is necessary. According to Remark 2, though, for semi-equational CTRSs additional predicate symbols $\leftrightarrow$ and $\leftrightarrow^*$ are necessary. We just need to enrich $\mathcal{M}_\mathcal{R}$ with the corresponding interpretations for those new predicate symbols.

[5] By Gödel's completeness theorem, see, e.g., [20, Corollary 2.19], deducibility and logical consequence are equivalent, i.e., $\mathsf{Th} \vdash \varphi$ iff $\mathsf{Th} \models \varphi$.

5 Proofs of confluence using critical pairs

In proofs of confluence, joinability of critical pairs plays a main role. A *conditional critical pair* (CCP) is an expression $\langle s,t\rangle \Leftarrow C$ where $\langle s,t\rangle$ is the *peak* of the CCP, for terms s and t, and C is the *conditional part*, i.e., a sequence $s_1 \approx t_1,\ldots,s_n \approx t_n$ of conditions. They are obtained from CTRSs as follows, see, e.g., [7, Definition 3] and also [23, Definition 7.1.8(1)].

▶ **Definition 13** (Conditional critical pair). *Let $\mathcal{R}$ be a CTRS. Let $\alpha : \ell \to r \Leftarrow C$ and $\alpha' : \ell' \to r' \Leftarrow C'$ be rules of $\mathcal{R}$ sharing no variable (rename if necessary). Let $p \in \mathcal{P}os_{\mathcal{F}}(\ell)$ be a nonvariable position of ℓ such that $\ell|_p$ and ℓ' unify with* mgu *σ. Then, we call the expression $\langle \sigma(\ell[r']_p),\sigma(r')\rangle \Leftarrow \sigma(C),\sigma(C')$ a* conditional critical pair *(CCP) of $\mathcal{R}$. If α and α' are (possibly renamed versions of) the same rule, the case $p = \Lambda$ is not considered to obtain a CCP.*

CCPs $\langle s,t\rangle \Leftarrow C$ whose conditional part C is empty are called *critical pairs* and simply written $\langle s,t\rangle$ as in the usual notation and definition, see, e.g., [23, Definition 4.2.1]. TRSs have (unconditional) critical pairs only; the set of critical pairs of a TRS $\mathcal{R}$ is denoted $\mathsf{CP}(\mathcal{R})$. In the following, $\mathsf{CCP}(\mathcal{R})$ denotes the set of CCPs of a CTRS $\mathcal{R}$. Note that $\mathsf{CP}(\mathcal{R}) \subseteq \mathsf{CCP}(\mathcal{R})$, as ordinary, unconditional critical pairs are particular CCPs with an empty conditional part. Although conditions $s_i \approx t_i$ admit multiple interpretations (as joinability, reachability, etc.), joinability of a critical pair is homogeneously defined as follows [23, Definition 7.1.8(2)]:

▶ **Definition 14** (Joinable conditional critical pair). *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s,t\rangle \Leftarrow C$ be a critical pair. We say that π is* joinable *if $\sigma(s) \downarrow_{\mathcal{R}} \sigma(t)$ holds for all substitutions σ such that $\sigma(C)$ holds. Otherwise, π is not joinable.*

An important aspect in the analysis of confluence is checking (conditional) critical pairs for (non)joinability. The following result provides a logical characterization of joinability of the CCPs of a CTRS $\mathcal{R}$ as satisfiability in $\mathcal{M}_{\mathcal{R}}$.

▶ **Proposition 15.** *Let $\mathcal{R}$ be a CTRS. A CCP $\pi : \langle s,t\rangle \Leftarrow C$ is joinable if and only if $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds, where $\vec{x} = x_1,\ldots,x_m$ are the variables occurring in C,s,t and $z \notin \mathcal{V}ar(C,s,t)$.*

The following sections investigate how to prove and disprove joinability of conditional critical pairs by (dis)proving appropriate feasibility problems using existing tools like infChecker[6] to automatically prove and disprove such feasibility problems [11].

6 Joinability of Terms and Feasibility Problems

Given a set $\mathbb{P}$ of (binary) predicates, let $\mathbb{T} = \{\mathsf{Th}_{\bowtie} \mid \bowtie \in \mathbb{P}\}$ be a $\mathbb{P}$-indexed set of first-order theories $\mathsf{Th}_{\bowtie}$ defining predicates $\bowtie$. An *f-condition* is an atom $s \bowtie t$ where $\bowtie \in \mathbb{P}$ and $s,t \in \mathcal{T}(\mathcal{F},\mathcal{X})$. Sequences $\mathsf{F} = (\gamma_i)_{i=1}^n = (\gamma_1,\ldots,\gamma_n)$ of f-conditions are called *f-sequences*. We often drop "f-" when no confusion arises. Empty sequences are written $()$.

▶ **Definition 16** (Feasibility). *A condition $s \bowtie t$ is* $(\mathbb{T},\sigma)$-feasible *if $\mathsf{Th}_{\bowtie} \vdash \sigma(s) \bowtie \sigma(t)$ holds; otherwise, it is* $(\mathbb{T},\sigma)$-infeasible. *We also say that $s \bowtie t$ is* $\mathbb{T}$-feasible *(or $\mathsf{Th}_{\bowtie}$-feasible, or just feasible if no confusion arises) if it is $(\mathbb{T},\sigma)$-feasible for some substitution σ; otherwise, we call it* infeasible.

A sequence F is $\mathbb{T}$-feasible (or just feasible*) iff there is a substitution σ such that, for all $\gamma \in \mathsf{F}$, γ is $(\mathbb{T},\sigma)$-feasible. Note that $()$ is trivially feasible.*

In the following, $\mathsf{Th}_{\bowtie} = \overline{\mathcal{R}}$ for all $\bowtie \in \{\to,\to^*,\downarrow,\ldots\}$.

[6] http://zenon.dsic.upv.es/infChecker/

6.1 Proving Conditional Joinability

Proposition 15 characterizes joinability of the CCPs of a CTRS $\mathcal{R}$ as the satisfiability of a logical sentence in $\mathcal{M}_{\mathcal{R}}$. In the following, we show how to advantageously use the results in [14, 11] to prove and disprove joinability of CCPs. The following consequence of Proposition 15 and [14, Corollary 14] provides a sufficient condition for joinability of CCPs.

▶ **Corollary 17.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s,t\rangle \Leftarrow C$ be a critical pair. If $\overline{\mathcal{R}} \vdash (\forall\vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds, then π is joinable.*

This result can be used together with theorem provers like Prover9 [19] for a practical use in proofs of joinability of critical pairs. The following result is a consequence of Proposition 15.

▶ **Corollary 18.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s,t\rangle \Leftarrow C$ be a critical pair. If $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible, then π is joinable.*

▶ **Example 19.** Consider the following variant $\mathcal{R}$ of the CTRS in Example 3:

$$a \to b \tag{12}$$

$$f(c,x) \to a \Leftarrow f(x,a) \approx f(b,b) \tag{13}$$

$$f(y,y) \to b \tag{14}$$

Note that rule (13) is feasible, both under join and oriented semantics: $f(\underline{a},a) \to f(b,\underline{a}) \to f(b,b)$ (which implies $f(a,a) \downarrow f(b,b)$). The only CCP is $\pi : \langle a,b\rangle \Leftarrow f(c,a) \approx f(b,b)$. Since $a \to^* z, b \to^* z$ is $\overline{\mathcal{R}}$-feasible (both for the join and oriented semantics of $\mathcal{R}$), by Corollary 18, π is joinable.

6.2 Disproving Conditional Joinability

Regarding proofs of *non-joinability*, we show how to formulate it as an feasibility problem.

▶ **Proposition 20.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s,t\rangle \Leftarrow C$ be a critical pair such that $C^{\downarrow}$ is $\overline{\mathcal{R}}$-feasible. If $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-infeasible, then π is not joinable.*

▶ **Example 21.** Consider the following CTRS $\mathcal{R}$

$$f(x,x) \to x \Leftarrow f(x,x) \approx b \tag{15}$$

$$f(y,y) \to b \tag{16}$$

There is only one critical pair $\pi : \langle x,b\rangle \Leftarrow f(x,x) \approx b$. Note that $f(c_x,c_x) \approx b$ is $\overline{\mathcal{R}}$-feasible due to the unconditional rule (this can be proved with infChecker). Non-joinability of π can be proved as the $\mathcal{R}$-infeasibility of

$$c_x \to^* z, b \to^* z \tag{17}$$

using infChecker. By Proposition 20, π is not joinable.

The following result characterizes joinability of CCPs $\langle s,t\rangle \Leftarrow C$ where the conditional part C and the peak $\langle s,t\rangle$ share no variable.

▶ **Proposition 22.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s,t\rangle \Leftarrow C$ be a critical pair such that $\mathcal{V}ar(s,t) \cap \mathcal{V}ar(C) = \emptyset$. Then, π is joinable if and only if C is $\overline{\mathcal{R}}$-infeasible or $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible.*

▶ **Example 23.** Consider the CTRS $\mathcal{R}$ in Example 3. As in Example 19 for (13), rule (2) is feasible, both under join and oriented semantics. There is a single critical pair $\pi : \langle c,b\rangle \Leftarrow f(c,a) \approx f(b,b)$.

- As a *join* CTRS, $\overline{\mathcal{R}}$-feasibility of $f(c,a) \downarrow f(b,b)$ together with $\overline{\mathcal{R}}$-infeasiblity of $c \to^* z, b \to^* z$ can both be proved with infChecker. Thus, π is not joinable.
- As an *oriented* CTRS, $\overline{\mathcal{R}}$-infeasibility of $f(c,a) \to^* f(b,b)$ can be proved with infChecker. Thus, π is joinable.

For TRSs, whose critical pairs have no conditional part, we have the following characterization of joinability as a consequence of Proposition 22.

▶ **Corollary 24.** *Let $\mathcal{R}$ be a (C)TRS. A critical pair $\langle s,t\rangle$ is joinable if and only if $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible.*

Actually, Corollary 24 characterizes *joinability of terms s and t* (being part of a critical pair or not).

▶ **Example 25.** Consider the following TRS from COPS (`http://cops.uibk.ac.at/?q=999`)

$$a(b(x)) \to b(c(x)) \quad (18)$$
$$c(b(x)) \to b(c(x)) \quad (19)$$
$$c(b(x)) \to c(c(x)) \quad (20)$$
$$b(b(x)) \to a(c(x)) \quad (21)$$
$$a(b(x)) \to a(b(x)) \quad (22)$$
$$c(c(x)) \to c(b(x)) \quad (23)$$
$$a(c(x)) \to c(a(x)) \quad (24)$$

By Corollary 24, joinability of the critical pair $\langle a(a(c(x))), b(c(b(x)))\rangle$ can be *disproved* as the infeasibility of $a(a(c(c_x))) \to^* z, b(c(b(c_x))) \to^* z$, which is proved by infChecker.

7 Confluence of CTRSs

In the analysis of confluence of CTRSs, a crucial notion is that of *conditional* critical pairs associated to a CTRS $\mathcal{R}$. We have the following (well-known) fact.

▶ **Proposition 26.** *Let $\mathcal{R}$ be a CTRS. If* $\mathsf{CCP}(\mathcal{R})$ *contains a non-joinable CCP, then $\mathcal{R}$ is not (locally) confluent.*

▶ **Example 27.** For the TRS $\mathcal{R}$ in Example 25, since $\mathsf{CP}(\mathcal{R})$ contains a nonjoinable critical pair $\langle a(a(c(x))), b(c(b(x)))\rangle$, by Proposition 26 we conclude that $\mathcal{R}$ is not confluent.

▶ **Example 28.** As a consequence of Proposition 26, $\mathcal{R}$ in Example 3, when considered as a join CTRS, is not confluent. Except for CONFident, no tool available on the confluence platform CoCoWeb [13], which provides access to several confluence tools, was able to reach this conclusion, as join CTRSs are accepted (as part of COPS syntax), but currently unsupported by other confluence tools in the platform. CONFident is able to provide a negative answer using Proposition 22 to prove nonjoinability of the only CCP, and then Proposition 26 to conclude nonconfluence.

Dershowitz, Okada, and Sivakumar proved that *a terminating (*noetherian in their terminology*) join CTRSs is confluent if all its critical pairs are joinable overlays* [7, Theorem 4], where a (conditional) critical pair is an overlay if the critical position is the top position Λ [7, Definition 8].

▶ **Example 29.** Note that the CCP π for $\mathcal{R}$ in Example 19 is an overlay. It is joinable, as proved in the example (both for the join and oriented semantics). The CTRS $\mathcal{R}$ is terminating as the underlying TRS $\mathcal{R}_u = \{a \to b, f(c,x) \to a, f(x,x) \to b\}$ is clearly terminating. Thus, by [7, Theorem 4], $\mathcal{R}$ (viewed as a join CTRS) is confluent.

Unfortunately, this does *not* hold for oriented CTRSs.

▶ **Example 30.** The following oriented CTRS $\mathcal{R}$ [26, Counterexample 3.3]

$$a \to b \quad (25)$$
$$f(x) \to c \Leftarrow x \approx a \quad (26)$$

is terminating (the underlying TRS $\mathcal{R}_u = \{a \to b, f(x) \to c\}$ is clearly terminating), and has no (conditional) critical pair. However, $f(b) \leftarrow f(a) \to c$, but c is irreducible and $f(b)$ also is as the conditional part $x \approx a$ of rule (26), when instantiated by $b \approx a$ is not satisfiable by using a reachability test $b \to^* a$. Hence $f(b)$ and c are *not* joinable and $\mathcal{R}$ is not confluent.

Normal CTRSs are CTRSs where terms t in conditions $s \approx t$ of the conditional part of rules are *ground, irreducible terms.*

▶ Remark 31 (Normal join, oriented, and semiequational CTRSs). Nowadays, the notion of a normal CTRS $\mathcal{R}$ usually assumes that $\mathcal{R}$ is an oriented CTRS, see, e.g., [23, Definition 7.1.3]. Other authors, though, have defined the notion of a normal *join* CTRS as one whose joinability conditions $s \downarrow t$ in conditional rules always satisfy the restriction of t being an irreducible ground term [7, Definition 2]; then, the authors remark that normal join CTRSs can be seen as what we call normal CTRSs today. Hence, normal join and oriented CTRSs coincide. As for *semi-equational* CTRSs, if normality is required, then $s \leftrightarrow^* t$ if and only if $s \to^* t$ because t is irreducible. Therefore, when referring to normal join, oriented, or semiequational CTRSs we are actually dealing with one and the same kind of CTRSs.

For this reason, conditions of normal CTRSs can be equivalently handled as joinability conditions $s_i \downarrow t_i$. Neither $\mathcal{R}$ in Example 30 nor $\mathcal{R}$ in Example 19 are normal. The following result, which is a simple consequence of [7, Theorem 4], is useful:

▶ **Corollary 32.** *A terminating normal CTRS is confluent if all its critical pairs are joinable overlays.*

▶ **Example 33.** Consider the following normal CTRS

$$
\begin{aligned}
c &\to b && (27)\\
d &\to b && (28)\\
f(a,x) &\to c \Leftarrow x \approx a && (29)\\
f(x,x) &\to d \Leftarrow x \approx a && (30)\\
g(x) &\to d \Leftarrow g(x) \approx b && (31)\\
g(a) &\to f(a,a) && (32)
\end{aligned}
$$

which is terminating, as the underlying TRS $\mathcal{R}_u$ is clearly terminating. The system has two (overlay) conditional critical pairs which are feasible and joinable:

$$
\begin{aligned}
\langle c,d\rangle &\Leftarrow a \approx a && \text{with (29) and (30)} && (33)\\
\langle d,f(a,a)\rangle &\Leftarrow g(a) \approx b && \text{with (31) and (32)} && (34)
\end{aligned}
$$

As for (33), using (27) and (28) we join c and d into b. Regarding (34), we have $\underline{f(a,a)} \to_{(30)} d$. By Corollary 32, $\mathcal{R}$ is confluent. No tool in CoCoWeb is able to prove it.

An oriented CTRS $\mathcal{R}$ is called *deterministic* (DCTRS) if for each rule $\ell \to r \Leftarrow s_1 \approx t_1, \ldots, s_n \approx t_n$ in $\mathcal{R}$ and each $1 \leq i \leq n$, we have $\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(\ell) \cup \bigcup_{j=1}^{i-1} \mathcal{V}ar(t_j)$. In the literature on confluence of DCTRSs, some results that use termination properties of CTRSs to guarantee confluence have been reported. For instance, [2, Theorem 4.1] establishes that *a quasi-reductive strongly deterministic CTRS is confluent if and only if its CCPs are all joinable.* Strongly deterministic CTRSs (SDCTRSs) are DCTRSs $\mathcal{R}$ where all conditions $s \approx t$ in the conditional part C of rules $\ell \to r \Leftarrow C \in \mathcal{R}$ are *strongly irreducible*, i.e., every instance $\sigma(t)$ of t by an irreducible substitution σ is irreducible [2, Definition 4.1]. Clearly, normal DCTRSs are SDCTRSs. Quasi-reductiveness (see, e.g., [23, Definition 7.2.36]) guarantees quasi-decreasingness of the DCTRS (see [23, Definition 7.2.39 and Lemma 7.2.40]). As proved in [16], quasi-decreasingness is equivalent to *operational termination.* Other results in the literature (see [23, Section 7.3]) usually require quasi-decreasingness, i.e, operational termination. Operationally terminating CTRSs are *terminating*, but not vice versa. For instance, viewed as an *oriented* CTRS, the (deterministic) CTRS $\mathcal{R}$ in Example 33 is terminating (this can be proved by using MU-TERM) but it is *not* operationally terminating (this can also be proved with MU-TERM). Therefore, the aforementioned results in [2, 23] *cannot* be used to prove confluence of $\mathcal{R}$ in Example 33. Further results for proving confluence of terminating CTRSs are reported in [10]; however, they apply to *join* CTRSs only.

Most confluence criteria for proving confluence of CTRSs involve checking (non)joinability of (feasible) CCPs, possibly in connection with other structural or syntactical requirements on the CTRS (e.g., left-linearity, etc.). The focus in this paper has been the investigation of (non)joinability criteria which can be used together with these confluence criteria. The following section discusses our implementation of those techniques and its impact in proofs of confluence of CTRSs in practice.

Table 1 Meaning of `CONDITIONTYPE` in COPS syntax.

`CONDITIONTYPE`	here `==` means
`ORIENTED`	$\to^*$
`JOIN`	$\downarrow$
`SEMI-EQUATIONAL`	$\leftrightarrow^*$

8 Implementation and Experimental Evaluation

CONFident 1.0 is written in Haskell and consists of 80 Haskell modules with around 14000 lines of code. The tool is accessible through its web interface (see Section 1). The input format is an extended version of the Confluence Competition (CoCo) format [21], which is the official format used in the *confluence* (CR) category of the competition. The input is a CTRS $\mathcal{R}$ in TPDB/COPS format[7]. As in COPS syntax, symbol `==` stands for $\approx$ above to specify the conditional part of rewrite rules. Its meaning depends on the `CONDITIONTYPE` section of the input specifying how the conditions of rules are evaluated [23, Definition 7.1.3] according to Table 1. Furthermore, in our extended version of TPDB/COPS syntax we can combine those relations by using them directly in the condition part of the rules: we use `->*` for $\to^*$, `->*<-` for $\downarrow$ and `<->*` for $\leftrightarrow^*$.

The implementation is based on a divide and conquer schema where, given an input problem, there is a set of techniques and an application strategy for those techniques. The techniques can simplify the problem, reduce it into a set of simpler problems or just give a positive or negative answer. We consider two types of problems: *Rewriting problems* and *Conditional Rewriting* problems. Each type of problem has its own strategy and processors. Although there are processors that can be applied to Rewriting and Conditional Rewriting problems indifferently, from the implementation point of view we prefer to implement them separately because we can apply simplifications when conditions are not considered. According to Section 3.1, when the system is parsed, the tool computes $\overline{\mathcal{R}}_J$, $\overline{\mathcal{R}}_O$, or $\overline{\mathcal{R}}_{\mathrm{SE}}$ (depending on the `CONDITIONTYPE` section) and then applies the appropriate strategy. Our proof strategy is based on experimentation: we try to first apply techniques that simplify the problems and reduce them into simpler problems (e.g., remove unnecesary rules and apply modularity results). When all simplification techniques have been applied, we analyze the problem in order to extract good properties that guide the strategy (linearity, weak normalization, termination, operational termination, strongly deterministic conditions, or right stability, see [16, 23] for definitions of these concepts). Then we calculate its conditional critical pairs and apply the techniques presented in the paper combined with classical techniques to check the joinability or non-joinability of the critical pairs. We also apply transformations to convert CTRSs into confluence equivalent TRSs and CS-TRSs [15]. If the final answer is YES or NO, the tool displays a report in plain text. Otherwise, MAYBE is returned. More details can be found in [28].

We participated in the CTRS (CR) category of CoCo 2021.[8] With a timeout of 60 seconds, the participating tools are expected to return a proof of confluence or nonconfluence (or a *maybe* answer) for each of the considered problems. The other participating tools this year were CO3 [22] and ACP [1]. The test set used in CoCo 2021 included 100 examples (see `http://cops.uibk.ac.at/results/?y=2021&c=CTRS`). The following table sumarizes the obtained results:[9]

CTRS CR Tool	Yes	No	Total
CONFident	37	24	61
CO3	28	19	47
ACP	29	15	44

Accordingly, CONFident was declared the winner of the competition.[10]

[7] See `http://zenon.dsic.upv.es/muterm/?name=documentation#formats`

[8] `http://project-coco.uibk.ac.at/2021/`

[9] The 2020 version of the tool ConCon `http://cl-informatik.uibk.ac.at/software/concon/` participated in CoCo 2021 as the winner of the CTRS category in 2020. Its results are displayed in the aforementioned web page.

[10] See `http://project-coco.uibk.ac.at/2021/results.php`

CONFident is able to obtain confluence proofs not only for *oriented* CTRSs (which is the focus of CoCo CTRS category) but also for *join* CTRSs as explained above (and currently unsupported by the tools participating in the confluence competition). Full proofs for the discussed examples of join CTRSs can also be found in the benchmarks section of the tool web site.

9 Related Work

Plaisted's presentation of conditional rewriting [24] is related to ours. Conditional rules are viewed as (universally quantified) formulas $C \Rightarrow \ell \to r$, which can be seen as first-order formulas. Semantic interpretations, though, consist of a *base* domain D_B (an "ordinary" domain as introduced in Section 3.3) and an *extended domain* $D_E = \mathcal{T}(\mathcal{F} \cup D_B)$ where values of the domain D_B are treated as *constants*. Symbols f have an interpretation[11] f^I, i.e., a mapping $f^I : D_E \times \cdots \times D_E \to D_E$ defined so that $f^I(t_1, \ldots, t_k) = f(t_1, \ldots, t_k)$. Conveniently, if $a \in D_B$, then $a^I = a$. Terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ are interpreted as usual, except that Plaisted also interprets variables $x \in \mathcal{X}$ as $x^I \in D_E$. The usual *valuation of variables* of first-order logic is therefore integrated as part of the interpretation I. In this respect, his semantic approach differs from the usual one in first order logic (indeed, he rather speaks of *term logic* when referring it). Predicates $\to$ and $\to^*$ are interpreted as subsets of $D_E \times D_E$. Atoms $s \to t$ and $s \to^* t$ are then interpreted as expected: $(s \to t)^I = s^I \to^I t^I$ and $(s \to^* t)^I = s^I (\to^*)^I t^I$. An interpretation I is a *rewriting model* of a CTRS $\mathcal{R}$ if I satisfies the formulas in $\mathcal{R}$ together with a number of *axioms* A which, essentially, are the ones we obtain from the inference rules (Rf), (T), and $(C)_{f,i}$ for $f \in \mathcal{F}$ and $1 \leq i \leq ar(f)$. Plaisted writes $\mathcal{R} \models_m \varphi$ if φ is true in *all* minimal rewriting models of $\mathcal{R}$.[12] Then, a CTRS $\mathcal{R}$ is said to be *confluent* if $\mathcal{R} \models_m \varphi_{\rm CR}$ holds. Finally, on page 219, the confluence property of a CTRS is proved equivalent to $\mathcal{R}_I \models_m \varphi_{\rm CR}$ for all minimal rewriting models I of $\mathcal{R}$, where $\mathcal{R}_I$ is the (possibly infinite) TRS (i.e., without conditional rules) obtained from $\mathcal{R}$ and I by considering rules $s \to t$ (where $s, t \in \mathcal{T}(\mathcal{F} \cup D_B)$) such that $s \to t$ is a ground instance of $\ell \to r$ for some conditional rule $\ell \to r \Leftarrow C$, where variables are replaced by terms in $\mathcal{T}(\mathcal{F} \cup D_B)$ and the corresponding instance C' of C is true in I. Similar definitions are provided for local confluence and joinability of critical pairs (which Plaisted calls to *pass the critical pair test*). Note that, since there can be infinitely many interpretations I for a given CTRS $\mathcal{R}$, proofs of confluence in term logic involve the consideration of infinitely many TRSs $\mathcal{R}_I$. In contrast, our definitions of confluence, local confluence, and joinability of CCPs use a single model $\mathcal{M}_{\mathcal{R}}$.

In the so-called *first-order theory of rewriting* (*FOThR* in the following), a restricted first-order language (without constant or function symbols), is used. The predicate symbols $\to$ and $\to^*$ are interpreted on the least (ground) Herbrand model $\mathcal{H}_{\mathcal{R}}$ for the signature $\mathcal{F}$ and predicates $\to$ and $\to^*$ [6]. In *FOThR* properties of TRSs $\mathcal{R} = (\mathcal{F}, R)$ are expressed by satisfiability in $\mathcal{H}_{\mathcal{R}}$ of *FOThR*. For instance $\mathcal{H}_{\mathcal{R}} \models \varphi_{\rm CR}$ means "*the TRS $\mathcal{R}$ is ground confluent*" (the restriction to ground confuence is due to the use of $\mathcal{H}_{\mathcal{R}}$, which consists of atoms $s \to t$ and $s \to^* t$ where $s, t \in \mathcal{T}(\mathcal{F})$). Decision algorithms for *FOThR* exist for restricted classes of TRSs $\mathcal{R}$ like left-linear right-ground TRSs, where variables are allowed in the left-hand side of the rules (without repeated occurrences of the same variable) but disallowed in the right-hand side [25]. However, a simple fragment of *FOThR* like the *First-Order Theory of One-Step Rewriting*, where only a single predicate symbol $\to$ representing one-step rewritings with $\mathcal{R}$ is allowed, has been proved undecidable even for *linear* TRSs [27]. In contrast, we use the full expressive power of first-order logic to represent not only TRSs but also CTRSs . Also in contrast to *FOThR*, where function symbols are not allowed in formulas, we can use *arbitrary* sentences involving arbitrary terms. This is crucial, for instance, to investigate joinability of CCPs $\langle s, t \rangle \Leftarrow C$, as s and t are arbitrary terms, and C usually involves nonvariable terms.

On the other hand, the idea of turning variables into constants to see terms with variables as ground terms of an extended signature is standard in algebraic specifications, see, e.g., [9, page 9]. However, as far as we know, such a model has not been used in the definition or verification of

[11] Plaisted interprets symbols in two different ways. This is due to the more general kind of conditional systems he considers, where the conditional part of rules can include first-order literals defined by an additional first-order theory. Our simplified presentation suffices to handle the CTRSs considered here.

[12] Plaisted obtains each of such minimal models as follows: given an interpretation I, he takes the least model of the Horn clauses obtained as the ground instances of rules $\alpha : \ell \to r \Leftarrow C$ when variables in α are replaced by terms in $\mathcal{T}(\mathcal{F} \cup D_B)$ (see the proof of his Theorem 1).

computational properties like confluence, which is the main focus of this paper. Also, the use of $\mathcal{M}_{\mathcal{R}}$ in Section 4 to define joinability of CCPs as satisfaction in $\mathcal{M}_{\mathcal{R}}$, and the translation in Section 6 of joinability problems into feasibility problems where terms with variables are "grounded" using $\_^{\downarrow}$ is, to the best of our knowledge, also new.

Research on confluence of CTRSs goes back to [4, 7], and many advances have been introduced in the last years, leading to the construction of several tools which are able to automatically prove it, see [21] and the references therein. To the best of our knowledge, though, our characterization of (local) confluence of CTRSs as the satisfiability of appropriate logical formulas in $\mathcal{M}_{\mathcal{R}}$ (Theorem 11) and its practical use in Section 7 is new. Also, the idea of decomposing proofs of confluence into (in)feasibility problems by taking into account the structure of the logic formula, and the use of constants instead of variables to improve these proves seems to be new.

10 Conclusions and Future Work

In this paper, we deal with computational (reduction) relations $\to$ and $\to^*$ associated to reduction-based systems $\mathcal{R}$ in logic form: reduction steps are defined by provability in a given inference system $\mathcal{I}(\mathcal{R})$ obtained from $\mathcal{R}$ and the generic system describing the operational semantics of the language of $\mathcal{R}$, or, equivalently, as logical consequences of a theory $\overline{\mathcal{R}}$ obtained similarly. We have characterized (local) confluence of CTRSs $\mathcal{R}$ as the satisfiability of appropriate first-order formulas φ_{WCR} and φ_{CR} in a canonical model $\mathcal{M}_{\mathcal{R}}$ where variables are treated as constants and terms with variables in $\mathcal{T}(\mathcal{F},\mathcal{X})$ are treated as ground terms in $\mathcal{T}(\mathcal{F}\cup C_{\mathcal{X}})$ (Theorem 11). We have also similarly characterized joinability of CCPs $\langle s,t\rangle \Leftarrow C$ (Proposition 15). Then, we show how to translate joinability problems into (combinations of) feasibility problems which can be solved using appropriate techniques and tools. For this purpose, the introduction of constants $c_x \in C_{\mathcal{X}}$ instead of variables $x \in \mathcal{X}$ in feasibility goals has been useful to obtain faster proofs.

We have developed a new tool implementing our results: CONFident. We participated in the 2021 edition of the Confluence Competition (CoCo) in the CTRS CR (confluence of CTRSs) category obtaining good results.

As for future work, we plan to apply our techniques to prove confluence of rewriting-based programming languages like Maude, whose conditional rules include componenents not explicitly considered here (matching conditions, etc.) but whose semantics can be defined by using the general approach sketched in Section 3. Since the analysis of confluence of rewrite theories (which provide the formal basis for the operational description of Maude programs) is also based in the analysis of joinability of the appropriate critical pairs [8], we think that our approach will be useful as well.

References

1. Takahito Aoto, Junichi Yoshida, and Yoshihito Toyama. Proving confluence of term rewriting systems automatically. In Ralf Treinen, editor, *Rewriting Techniques and Applications, 20th International Conference, RTA 2009, Brasília, Brazil, June 29 - July 1, 2009, Proceedings*, volume 5595 of *Lecture Notes in Computer Science*, pages 93–102. Springer, 2009. doi:10.1007/978-3-642-02348-4_7.
2. Jürgen Avenhaus and Carlos Loría-Sáenz. On conditional rewrite systems with extra variables and deterministic logic programs. In Frank Pfenning, editor, *Logic Programming and Automated Reasoning, 5th International Conference, LPAR'94, Kiev, Ukraine, July 16-22, 1994, Proceedings*, volume 822 of *Lecture Notes in Computer Science*, pages 215–229. Springer, 1994. doi:10.1007/3-540-58216-9_40.
3. Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.
4. Jan A. Bergstra and Jan Willem Klop. Conditional rewrite rules: Confluence and termination. *J. Comput. Syst. Sci.*, 32(3):323–362, 1986. doi:10.1016/0022-0000(86)90033-4.
5. Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn L. Talcott. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007. doi:10.1007/978-3-540-71999-1.

6 Max Dauchet and Sophie Tison. The theory of ground rewrite systems is decidable. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 242–248. IEEE Computer Society, 1990. doi:10.1109/LICS.1990.113750.

7 Nachum Dershowitz, Mitsuhiro Okada, and G. Sivakumar. Confluence of conditional rewrite systems. In Stéphane Kaplan and Jean-Pierre Jouannaud, editors, *Conditional Term Rewriting Systems, 1st International Workshop, Orsay, France, July 8-10, 1987, Proceedings*, volume 308 of *Lecture Notes in Computer Science*, pages 31–44. Springer, 1987. doi:10.1007/3-540-19242-5_3.

8 Francisco Durán and José Meseguer. On the church-rosser and coherence properties of conditional order-sorted rewrite theories. *J. Log. Algebraic Methods Program.*, 81(7-8):816–850, 2012. doi:10.1016/j.jlap.2011.12.004.

9 Joseph A. Goguen and José Meseguer. Models and equality for logical programming. In Hartmut Ehrig, Robert A. Kowalski, Giorgio Levi, and Ugo Montanari, editors, *TAPSOFT'87: Proceedings of the International Joint Conference on Theory and Practice of Software Development, Pisa, Italy, March 23-27, 1987, Volume 2: Advanced Seminar on Foundations of Innovative Software Development II and Colloquium on Functional and Logic Programming and Specifications (CFLP)*, volume 250 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 1987. doi:10.1007/BFb0014969.

10 Bernhard Gramlich and Claus-Peter Wirth. Confluence of terminating conditional rewrite systems revisited. In Harald Ganzinger, editor, *Rewriting Techniques and Applications, 7th International Conference, RTA-96, New Brunswick, NJ, USA, July 27-30, 1996, Proceedings*, volume 1103 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 1996. doi:10.1007/3-540-61464-8_56.

11 Raúl Gutiérrez and Salvador Lucas. Automatically proving and disproving feasibility conditions. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II*, volume 12167 of *Lecture Notes in Computer Science*, pages 416–435. Springer, 2020. doi:10.1007/978-3-030-51054-1_27.

12 Raúl Gutiérrez and Salvador Lucas. MU-TERM: Verify termination properties automatically (system description). In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II*, volume 12167 of *Lecture Notes in Computer Science*, pages 436–447. Springer, 2020. doi:10.1007/978-3-030-51054-1_28.

13 Nao Hirokawa, Julian Nagele, and Aart Middeldorp. Cops and CoCoWeb: Infrastructure for Confluence Tools. In Didier Galmiche, Stephan Schulz, and Roberto Sebastiani, editors, *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, volume 10900 of *Lecture Notes in Computer Science*, pages 346–353. Springer, 2018. doi:10.1007/978-3-319-94205-6_23.

14 Salvador Lucas. Proving semantic properties as first-order satisfiability. *Artif. Intell.*, 277, 2019. doi:10.1016/j.artint.2019.103174.

15 Salvador Lucas. Applications and extensions of context-sensitive rewriting. *Journal of Logical and Algebraic Methods in Programming*, 121:100680, 2021. doi:10.1016/j.jlamp.2021.100680.

16 Salvador Lucas, Claude Marché, and José Meseguer. Operational termination of conditional term rewriting systems. *Inf. Process. Lett.*, 95(4):446–453, 2005. doi:10.1016/j.ipl.2005.05.002.

17 Salvador Lucas and José Meseguer. Dependency pairs for proving termination properties of conditional term rewriting systems. *J. Log. Algebr. Meth. Program.*, 86(1):236–268, 2017. doi:10.1016/j.jlamp.2016.03.003.

18 Salvador Lucas, José Meseguer, and Raúl Gutiérrez. The 2D Dependency Pair Framework for conditional rewrite systems. Part I: Definition and basic processors. *J. Comput. Syst. Sci.*, 96:74–106, 2018. doi:10.1016/j.jcss.2018.04.002.

19 William McCune. Prover9 & Mace4. Technical report, University of New Mexico, 2005–2010. URL: http://www.cs.unm.edu/~mccune/prover9/.

20 Elliott Mendelson. *Introduction to mathematical logic (4. ed.).* Chapman and Hall, 1997.

21 A. Middeldorp, J. Nagele, and K. Shintani. CoCo 2019: report on the eight confluence competition. *J.International Journal on Software Tools for Technology Transfer*, to appear, 2021. doi:10.1007/s10009-021-00620-4.

22 Naoki Nishida, Makishi Yanagisawa, and Karl Gmeiner. On Proving Confluence of Conditional Term Rewriting Systems via the Computationally Equivalent Transformation. In *3rd International Workshop on Confluence, IWC 2014, July 13, 2014, Vienna, Austria*, page 42, 2014.

23 Enno Ohlebusch. *Advanced topics in term rewriting.* Springer, 2002.

24 David A. Plaisted. A logic for conditional term rewriting systems. In Stéphane Kaplan and Jean-Pierre Jouannaud, editors, *Conditional Term Rewriting Systems, 1st International Workshop, Orsay, France, July 8-10, 1987, Proceedings*, volume 308 of *Lecture Notes in Computer Science*, pages 212–227. Springer, 1987. doi:10.1007/3-540-19242-5_16.

25 Franziska Rapp and Aart Middeldorp. Automating the first-order theory of rewriting for left-linear right-ground rewrite systems. In Delia Kesner and Brigitte Pientka, editors, *1st International Conference on Formal Structures for Computation and Deduction, FSCD 2016, June 22-26, 2016, Porto, Portugal*, volume 52 of *LIPIcs*, pages 36:1–36:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.FSCD.2016.36.

26 Taro Suzuki, Aart Middeldorp, and Tetsuo Ida. Level-confluence of conditional rewrite systems with extra variables in right-hand sides. In Jieh Hsiang, editor, *Rewriting Techniques and Applications, 6th International Conference, RTA-95, Kaiserslautern, Germany, April 5-7, 1995, Proceedings*, volume 914 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 1995. doi:10.1007/3-540-59200-8_56.

27 Ralf Treinen. The first-order theory of linear one-step rewriting is undecidable. *Theor. Comput. Sci.*, 208(1-2):179–190, 1998. doi:10.1016/S0304-3975(98)00083-8.

28 Miguel Vítores. *CONFident: a tool for confluence analysis of rewriting systems.* PhD thesis, Departamento de Sistemas Informáticos y Computación. Universitat Politècnica de València, December 2021.

A Proofs of theorems

▶ **Proposition 6.** *Let $\mathcal{R} = (\mathcal{F}, R)$ be a CTRS and $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then, $s \to_{\mathcal{R}} t$ if and only if $s^{\downarrow} \to_{\mathcal{R}} t^{\downarrow}$ and $s \to^*_{\mathcal{R}} t$ if and only if $s^{\downarrow} \to^*_{\mathcal{R}} t^{\downarrow}$.*

Proof. We develop the proof for oriented CTRSs. For join or equational CTRSs, it is similar. We proceed by multiple induction on the depth d of the proof trees used to prove each goal $s \to t$ (for $s \to_{\mathcal{R}} t$) and $s \to^* t$ (for $s \to^*_{\mathcal{R}} t$). If $d = 0$, then we consider two cases (we develop the *only if* part; the *if* part is analogous):

- $s \to t$ is proved using $(\mathrm{Rl})_\alpha$ for an unconditional rule $\alpha : \ell \to r$, i.e., there is a substitution σ such that $s = \sigma(\ell)$ and $t = \sigma(r)$. Since $s^{\downarrow} = \sigma(\ell)^{\downarrow} = \sigma^{\downarrow}(\ell)$ and $t^{\downarrow} = \sigma(\ell)^{\downarrow} = \sigma^{\downarrow}(\ell)$, we have that $s^{\downarrow} \to t^{\downarrow}$ is proved using the same rule.
- $s \to t$ is proved using (Rf). In this case, $s = t$ and hence $s^{\downarrow} \to^* t^{\downarrow}$ is proved using (Rf).

If $d > 0$, then

- $s \to t$ is proved in one of the following two possible ways:
  - using rule $(\mathrm{C})_{f,i}$ where $s = f(s_1,\ldots,s_i,\ldots s_k)$, $t = f(s_1,\ldots,t_i,\ldots,s_k)$ for some terms $s_1,\ldots,s_k,t_i$, using a proof tree $\frac{T}{s\to t}$, where T is of depth $d-1$ and $s_i \to t_i$ is the root of T. By the induction hypothesis, $s_i^{\downarrow} \to t_i^{\downarrow}$ can be proved and hence $f(s_i^{\downarrow},\ldots,s_i^{\downarrow},\ldots,s_k^{\downarrow}) = s^{\downarrow} \to t^{\downarrow} = f(s_i^{\downarrow},\ldots,t_i^{\downarrow},\ldots,s_k^{\downarrow})$ can be proved as well using $(\mathrm{C})_{f,i}$.
  - using rule $(\mathrm{Rl})_\alpha$ for some rule $\alpha : \ell \to r \Leftarrow s_1 \approx t_1,\ldots,s_n \approx t_n$ and proof tree $\frac{T_1 \quad \cdots \quad T_n}{s\to t}$, where $s = \sigma(\ell)$ and $t = \sigma(r)$ for some substitution σ, and, for all $1 \le i \le n$ T_i, is a proof tree with root $\sigma(s_i) \to^* \sigma(t_i)$ and depth at most $d-1$. By the induction hypothesis, $\sigma(s_i)^{\downarrow} \to^* \sigma(t_i)^{\downarrow}$ can be proved for all $1 \le i \le n$ using proof trees $T_i^{\downarrow}$ with root $\sigma(s_i)^{\downarrow} \to^* \sigma(t_i)^{\downarrow}$. Since for all terms $u \in \mathcal{T}(\mathcal{F},\mathcal{X})$, $\sigma(u)^{\downarrow} = \sigma^{\downarrow}(u)$, there is a proof of $s^{\downarrow} = \sigma^{\downarrow}(\ell) \to \sigma^{\downarrow}(\ell) = t^{\downarrow}$ using $(\mathrm{Rl})_\alpha$ with proof tree $\frac{T_1^{\downarrow} \quad \cdots \quad T_n^{\downarrow}}{s^{\downarrow}\to t^{\downarrow}}$.
- $s \to^* t$ is proved using (T) using a proof tree $\frac{T_1 \quad T_2}{s\to^* t}$ where T_1 is a proof tree with root $s \to u$ of depth at most $d-1$ for some term u and T_2 is a proof tree with root $u \to^* t$ of depth at most $d-1$. By the induction hypothesis, there are proof trees $T_1^{\downarrow}$ and $T_2^{\downarrow}$ with roots $s^{\downarrow} \to u^{\downarrow}$ and $u^{\downarrow} \to^* t^{\downarrow}$. Thus, $s^{\downarrow} \to^* t^{\downarrow}$ is proved by the proof tree $\frac{T_1^{\downarrow} \quad T_2^{\downarrow}}{s^{\downarrow}\to^* t^{\downarrow}}$. ◀

▶ **Theorem 8** *For all CTRSs $\mathcal{R}$, $\mathcal{M}_{\mathcal{R}} \models \overline{\mathcal{R}}$.*

Proof. We develop the proof for oriented CTRSs, for join and semi-equational CTRSs being similar. We consider the sentences derived from each of the four inference rules in $\mathfrak{I}_{\text{O-CTRS}}$:

- From rule (Rf) a single sentence $(\forall x)\, x \to^* x \in \overline{\mathcal{R}}$ is obtained. We need to prove that for all $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, $(t^{\downarrow},t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ holds (remind that $\mathcal{T}(\mathcal{F},\mathcal{X})$ and $\mathcal{T}(\mathcal{F}_{\mathcal{X}})$ are isomorphic). Since for all terms $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, $t \to_{\mathcal{R}}^* t$ can be proved in $\mathcal{I}(\mathcal{R})$ by using axiom (Rf), by definition of $\mathcal{M}_{\mathcal{R}}$, we have $(t^{\downarrow},t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ as required.
- From rule (T), a single sentence $(\forall x,y,z)\; x \to y \wedge y \to^* z \Rightarrow x \to^* z \in \overline{\mathcal{R}}$ is obtained. Then, $\mathcal{M}_{\mathcal{R}} \models (\forall x,y,z)\; x \to y \wedge y \to^* z \Rightarrow x \to^* z$ holds if and only if for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$, whenever both $(\sigma^{\downarrow}(x),\sigma^{\downarrow}(y)) \in \to^{\mathcal{M}_{\mathcal{R}}}$ and $(\sigma^{\downarrow}(y),\sigma^{\downarrow}(z)) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ hold, then $(\sigma^{\downarrow}(x),\sigma^{\downarrow}(z)) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ holds as well. If both $(\sigma^{\downarrow}(x),\sigma^{\downarrow}(y)) \in \to^{\mathcal{M}_{\mathcal{R}}}$ and $(\sigma^{\downarrow}(y),\sigma^{\downarrow}(z)) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ hold, then, by definition of $\mathcal{M}_{\mathcal{R}}$, we have $\sigma(x) \to_{\mathcal{R}} \sigma(y)$ and $\sigma(y) \to_{\mathcal{R}}^* \sigma(z)$. Hence, $\sigma(x) \to_{\mathcal{R}}^* \sigma(z)$ can be proved in $\mathcal{I}(\mathcal{R})$ and therefore $(\sigma^{\downarrow}(x),\sigma^{\downarrow}(z)) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ as desired.
- For all k-ary symbols $f \in \mathcal{F}$ and $1 \le i \le k$, from $(\mathrm{C})_{f,i}$ a sentence $(\forall x_1)\cdots(\forall x_k)(\forall y_i)x_i \to y_i \Rightarrow f(x_1,\ldots,x_i,\ldots,x_k) \to f(x_1,\ldots,y_i,\ldots,x_k)$ is obtained. It holds in $\mathcal{M}_{\mathcal{R}}$ because, for all terms $s_1,\ldots,s_k,t_i \in \mathcal{T}(\mathcal{F},\mathcal{X})$, if $(s_i^{\downarrow},t_i^{\downarrow}) \in \to^{\mathcal{M}_{\mathcal{R}}}$, then, by definition of $\mathcal{M}_{\mathcal{R}}$, $s_i \to_{\mathcal{R}} t_i$ can be proved in $\mathcal{I}(\mathcal{R})$, and using $(\mathrm{C})_{f,i}$ we know that $f(s_1,\ldots,s_i,\ldots,s_k) \to_{\mathcal{R}} f(s_1,\ldots,t_i,\ldots,s_k)$ can also be proved, i.e., $(f(s_1^{\downarrow},\ldots,s_i^{\downarrow},\ldots,s_k^{\downarrow}), f(s_1^{\downarrow},\ldots,t_i^{\downarrow},\ldots,s_k^{\downarrow})) \in \to^{\mathcal{M}_{\mathcal{R}}}$ holds.
- As for $(\mathrm{Rl})_\alpha$, with $\alpha : \ell \to r \Leftarrow s_1 \approx t_1,\ldots,s_n \to t_n$, there is a sentence $(\forall \vec{x}) \bigwedge_{i=1}^n s_i \to^* t_i \Rightarrow \ell \to r$ in $\overline{\mathcal{R}}$. Then, $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x}) \bigwedge_{i=1}^n s_i \to^* t_i \Rightarrow \ell \to r$ holds if and only if for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$, whenever $(\sigma^{\downarrow}(s_i),\sigma^{\downarrow}(t_i)) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ holds for all $1 \le i \le n$, then $(\sigma^{\downarrow}(\ell),\sigma^{\downarrow}(r)) \in \to^{\mathcal{M}_{\mathcal{R}}}$ holds as well. By definition of $\mathcal{M}_{\mathcal{R}}$, if $(\sigma^{\downarrow}(s_i),\sigma^{\downarrow}(t_i)) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ holds for all $1 \le i \le n$, then $\sigma(s_i) \to_{\mathcal{R}}^* \sigma(t_i)$ holds for all $1 \le i \le n$. Therefore, $\sigma(\ell) \to_{\mathcal{R}} \sigma(r)$ can be proved in $\mathcal{I}(\mathcal{R})$, and hence $(\sigma^{\downarrow}(\ell),\sigma^{\downarrow}(r)) \in \to^{\mathcal{M}_{\mathcal{R}}}$, as desired. ◀

▶ **Proposition 9.** *Let $\mathcal{R} = (\mathcal{F},R)$ be a CTRS, $s,t \in \mathcal{T}(\mathcal{F},\mathcal{X})$, and $\vec{x} = x_1,\ldots,x_n$ denote the variables occurring in s and t, i.e., $\mathcal{V}ar(s) \cup \mathcal{V}ar(t) = \{x_1,\ldots,x_n\}$. Then,*

1. *We have that $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$, if and only if $(s^{\downarrow},t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$.*
2. *$\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})\, s \to^* t$ if and only if $(s^{\downarrow},t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$.*

Proof.

1. As for the *if* part, if $(s^{\downarrow},t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ holds, then, by definition of $\mathcal{M}_{\mathcal{R}}$, $s \to_{\mathcal{R}}^* t$ holds. By closedness of $\to^*$ under substitution application, for all substitutions σ, we have $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$. Regarding the *only if* part, assume that for all substitutions σ, $\sigma(s) \to_{\mathcal{R}}^* \sigma(t)$ holds. In particular, for the empty substitution ϵ, we have $s = \epsilon(s) \to_{\mathcal{R}}^* \epsilon(t) = t$, i.e., $(s^{\downarrow},t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$.

2. The *if* part is as in the previous item, considering the definition of satisfiability in $\mathcal{M}_{\mathcal{R}}$ of a universally quantified formula. Regarding the *only if* part, if $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})\, s \to^* t$ holds, then for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$, $(\sigma^{\downarrow}(s), \sigma^{\downarrow}(t)) \in (\to^*_{\mathcal{R}})^{\mathcal{M}_{\mathcal{R}}}$ holds. In particular, for the empty substitution ϵ, we have $\epsilon^{\downarrow}(s) = s^{\downarrow}$ and $\epsilon^{\downarrow}(t) = t^{\downarrow}$, i.e., $(s^{\downarrow}, t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ holds. ◀

▶ **Theorem 11.** *A CTRS is (locally) confluent if and only if $\mathcal{M}_{\mathcal{R}} \models \varphi_{CR}$ (resp. $\mathcal{M}_{\mathcal{R}} \models \varphi_{WCR}$) holds.*

Proof. We develop the proof for confluence (i.e., φ_{CR}). For local confluence (i.e., φ_{WCR})) it is analogous. For the *if* part, if $\mathcal{M}_{\mathcal{R}} \models \varphi_{CR}$ holds, then, for all terms $s, t, u \in \mathcal{T}(\mathcal{F},\mathcal{X})$, whenever both $(s^{\downarrow}, t^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ and $(s^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ hold, there is $v \in \mathcal{T}(\mathcal{F},\mathcal{X})$ such that both $(t^{\downarrow}, v^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ and $(u^{\downarrow}, v^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ hold. By using Proposition 9, we conclude that, if $s \to^*_{\mathcal{R}} t$ and $s \to^*_{\mathcal{R}} u$ hold, then $t \to^*_{\mathcal{R}} v$ and $u \to^*_{\mathcal{R}} v$. Hence, $\mathcal{R}$ is confluent.

As for the *only if* part, if $\mathcal{R}$ is confluent, then for all terms $s, t, u \in \mathcal{T}(\mathcal{F},\mathcal{X})$, whenever $s \to^*_{\mathcal{R}} t$ and $s \to^*_{\mathcal{R}} u$, there is $v \in \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $t \to^*_{\mathcal{R}} v$ and $u \to^*_{\mathcal{R}} v$. By definition of $\mathcal{M}_{\mathcal{R}}$, this means that whenever $(s^{\downarrow}, t^{\downarrow}), (s^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$, we also have $(t^{\downarrow}, v^{\downarrow}), (u^{\downarrow}, v^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$. Thus, by Proposition 9, $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x}) s \to^* t \wedge s \to^* u$ implies $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x}) t \to^* v \wedge u \to^* v$, i.e., $\mathcal{M}_{\mathcal{R}} \models \varphi_{CR}$ holds. ◀

▶ **Proposition 15.** *Let $\mathcal{R}$ be a CTRS. A CCP $\pi : \langle s, t\rangle \Leftarrow C$ is joinable if and only if $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds, where $\vec{x} = x_1, \ldots, x_m$ are the variables occurring in C, s, t and $z \notin \mathcal{V}ar(C, s, t)$.*

Proof. We treat the particular case of oriented CTRSs. For join or semi-equational CTRSs it is similar. Let $C = s_1 \approx t_1, \ldots, s_n \approx t_n$. As for the *only if* part, if π is joinable, then for all substitutions $\sigma \in \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $\sigma(C)$ holds, i.e., for all $1 \leq i \leq n$, $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$ holds, there is a term $u \in \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $\sigma(s) \to^*_{\mathcal{R}} u$ and $\sigma(t) \to^*_{\mathcal{R}} u$ holds as well. By Proposition 6, if $\sigma(C)$ holds, then $\sigma^{\downarrow}(C)$ holds as well. Furthermore, if $\sigma(s) \to^*_{\mathcal{R}} u$ and $\sigma(t) \to^*_{\mathcal{R}} u$, then $\sigma^{\downarrow}(s) \to^*_{\mathcal{R}} u^{\downarrow}$ and $\sigma^{\downarrow}(t) \to^*_{\mathcal{R}} u^{\downarrow}$. Therefore, for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$, whenever $\sigma^{\downarrow}(C)$ holds, then there is $z \in \mathcal{T}(\mathcal{F}_{\mathcal{X}})$ such that $\sigma^{\downarrow}(s) \to^* z \wedge \sigma^{\downarrow}(t) \to^* z$ holds as well, i.e., $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds.

As for the *if* part, if $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})(\exists z) C \Rightarrow s \to^* z \wedge t \to^* z$ holds, then by definition of satisfiability in $\mathcal{M}_{\mathcal{R}}$, for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$, if $(\sigma(s_i)^{\downarrow}, \sigma(t_i)^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ holds for all $1 \leq i \leq n$, then there is $u \in \mathcal{T}(\mathcal{F},\mathcal{X})$ such that both $(\sigma(s)^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ and $(\sigma(t)^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ hold as well. By definition of $\mathcal{M}_{\mathcal{R}}$, for all substitutions σ, whenever $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$ holds for all $1 \leq i \leq n$, we have $\sigma(s) \to^* u$ and $\sigma(s) \to^* u$, i.e., π is joinable. ◀

▶ **Corollary 17.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s, t\rangle \Leftarrow C$ be a critical pair. If $\overline{\mathcal{R}} \vdash (\forall \vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds, then π is joinable.*

Proof. If $\overline{\mathcal{R}} \vdash (\forall \vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds, then $\overline{\mathcal{R}} \models (\forall \vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds as well. By Theorem 8, $\mathcal{M}_{\mathcal{R}} \models \overline{\mathcal{R}}$ holds. Hence, $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})(\exists z)\, C \Rightarrow s \to^* z \wedge t \to^* z$ holds. By Proposition 15, π is joinable. ◀

▶ **Corollary 18.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s, t\rangle \Leftarrow C$ be a critical pair. If $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible, then π is joinable.*

Proof. If $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible, there is a term $u \in \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $s^{\downarrow} \to^*_{\mathcal{R}} u^{\downarrow}$ and $t^{\downarrow} \to^*_{\mathcal{R}} u^{\downarrow}$, i.e., by Proposition 6, $s \to^*_{\mathcal{R}} u$ and $t \to^*_{\mathcal{R}} u$, hence $(s^{\downarrow}, u^{\downarrow}), (t^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$. By Proposition 9, $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})\, s \to^* u \wedge t^{\downarrow} \to^* u$, i.e., $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})(\exists z)\, s \to^* z \wedge t^{\downarrow} \to^* z$ holds. By Proposition 15, π is joinable. ◀

▶ **Proposition 20.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s, t\rangle \Leftarrow C$ be a critical pair such that $C^{\downarrow}$ is $\overline{\mathcal{R}}$-feasible. If $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-infeasible, then π is not joinable.*

Proof. By contradiction. If π is joinable, then for all substitutions $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$, if $\sigma(C)$ holds, then there is a term u such that $\sigma(s) \to^* u$ and $\sigma(t) \to^* u$. Since $C^{\downarrow}$ is $\overline{\mathcal{R}}$-feasible, no instantiation of variables in C is necessary for the condition C of π to hold, i.e., $\epsilon(C)$ holds and therefore $\epsilon(s) = s \downarrow_{\mathcal{R}} t = \epsilon(t)$ holds as well. By Proposition 6, $s^{\downarrow} \downarrow_{\mathcal{R}} t^{\downarrow}$ holds, i.e., $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible, leading to a contradiction. ◀

▶ **Proposition 22.** *Let $\mathcal{R}$ be a CTRS and $\pi : \langle s,t\rangle \Leftarrow C$ be a critical pair such that $\mathcal{V}ar(s,t) \cap \mathcal{V}ar(C) = \emptyset$. Then, π is joinable if and only if C is $\overline{\mathcal{R}}$-infeasible or $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible.*

Proof. By Proposition 15, π is joinable if and only if $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{x})(\exists z)C \Rightarrow s \to^* z \wedge t \to^* z$ holds. Since $\mathcal{V}ar(s,t) \cap \mathcal{V}ar(C) = \emptyset$, this is equivalent to $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{y}_1)\neg C \vee (\forall \vec{y}_2)(\exists z)s \to^* z \wedge t \to^* z$, where $\vec{y}_1$ are the variables $\mathcal{V}ar(C)$ and $\vec{y}_2$ are the variables $\mathcal{V}ar(s,t)$, with $\vec{y}_1 \cap \vec{y}_2 = \emptyset$ and $\vec{x} = \vec{y}_1 \cup \vec{y}_2$. This is equivalent to (i) $\mathcal{M}_{\mathcal{R}} \models \neg(\exists \vec{y}_1)C$ or (ii) $\mathcal{M}_{\mathcal{R}} \models (\forall \vec{y}_2)(\exists z)s \to^* z \wedge t \to^* z$. By definition of satisfiability in $\mathcal{M}_{\mathcal{R}}$ and using Proposition 6, (ii) is equivalent to the existence of a term u such that both $(s^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ and $(t^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ hold.

- Now, for the *if* part, we show that $\overline{\mathcal{R}}$-infeasibility of C implies (i) and $\overline{\mathcal{R}}$-feasibility of $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ implies (ii). First, if C is $\overline{\mathcal{R}}$-infeasible, then there is no substitution $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $\overline{\mathcal{R}} \vdash \sigma(C)$ holds. This clearly implies $\mathcal{M}_{\mathcal{R}} \models \neg(\exists \vec{y}_1)C$; otherwise, there would be a substitution $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $\mathcal{M}_{\mathcal{R}} \models \sigma(C)$ holds. By Proposition 6, though, this implies that $\overline{\mathcal{R}} \vdash \sigma(C)$ holds as well, leading to a contradiction. Second, if $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible, then there is $u \in \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $\overline{\mathcal{R}} \vdash s^{\downarrow} \to^* u^{\downarrow}$ and $\overline{\mathcal{R}} \vdash t^{\downarrow} \to^* u^{\downarrow}$, i.e., $s \to^*_{\mathcal{R}} u$ and $t \to^*_{\mathcal{R}} u$ holds. Therefore, $\mathcal{M}_{\mathcal{R}} \models (\exists z)s \to^* z \wedge t \to^* z$ holds and π is joinable.
- For the *only if* part, if (i) holds, then there is no substitution $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$ such that $\mathcal{M}_{\mathcal{R}} \models \sigma(C)$ holds. If C would be $\overline{\mathcal{R}}$-feasible, though, then, by [11, Theorem 1], $\overline{\mathcal{R}} \vdash (\exists \vec{y}_1)C$ holds. By using Theorem 8, we then conclude that $\mathcal{M}_{\mathcal{R}} \models (\exists \vec{y}_1)C$ holds, leading to a contradiction. Finally, if (ii) holds, then both $(s^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ and $(t^{\downarrow}, u^{\downarrow}) \in (\to^*)^{\mathcal{M}_{\mathcal{R}}}$ hold. By definition of $\mathcal{M}_{\mathcal{R}}$ and Proposition 6, we have $s^{\downarrow} \to^*_{\mathcal{R}} u$ and $t^{\downarrow} \to^*_{\mathcal{R}} u$, i.e., $s^{\downarrow} \to^* z, t^{\downarrow} \to^* z$ is $\overline{\mathcal{R}}$-feasible. ◀

▶ **Proposition 26.** *Let $\mathcal{R}$ be a CTRS. If* $\mathsf{CCP}(\mathcal{R})$ *contains a non-joinable CCP, then $\mathcal{R}$ is not (locally) confluent.*

Proof. If $\langle s,t\rangle \Leftarrow D \in \mathsf{CCP}(\mathcal{R})$ is not joinable, then, according to Definition 14, there is a substitution σ such that $\sigma(D)$ holds and $\sigma(s) \downarrow_{\mathcal{R}} \sigma(t)$ does not hold. Note that $s = \theta(\ell[r']_p)$ and $t = \theta(r')$ for some rules $\ell \to r \Leftarrow C$ and $\ell' \to r' \Leftarrow C'$, $p \in \mathcal{P}os_{\mathcal{F}}(\ell)$, *mgu* θ of $\ell|_p$ and ℓ', and $D = \theta(C), \theta(C')$. Since $\sigma(D) = \sigma(\theta(C)), \sigma(\theta(C'))$ holds, both $\sigma(\theta(C))$ and $\sigma(\theta(C'))$ hold as well (disregarding the join, oriented, or semiequational semantics for $\mathcal{R}$). Thus, $\sigma(\theta(\ell)) \to \sigma(s)$ using α' and $\sigma(\theta(\ell)) \to \sigma(t)$ using α. Since $\sigma(s)$ and $\sigma(t)$ are not joinable, $\mathcal{R}$ is not locally confluent. Hence, it is not confluent. ◀

▶ **Corollary 32.** *A terminating normal CTRS is confluent if all its critical pairs are joinable overlays.*

Proof. By [7, Theorem 4], a terminating conditional join CTRS whose critical pairs are all joinable overlays is confluent. Now, considering Remark 31, the statement of the corollary follows. ◀