Confluence Criteria for Logically Constrained Rewrite Systems

Jonas Schöpf(✉) and Aart Middeldorp

Department of Computer Science, Universität Innsbruck, Innsbruck, Austria
{jonas.schoepf,aart.middeldorp}@uibk.ac.at

Abstract. Numerous confluence criteria for plain term rewrite systems are known. For logically constrained rewrite system, an attractive extension of term rewriting in which rules are equipped with logical constraints, much less is known. In this paper we extend the strongly-closed and (almost) parallel-closed critical pair criteria of Huet and Toyama to the logically constrained setting. We discuss the challenges for automation and present crest, a new tool for logically constrained rewriting in which the confluence criteria are implemented, together with experimental data.

Keywords: Confluence · Term Rewriting · Constraints · Automation

1 Introduction

Logically constrained rewrite systems constitute a general rewrite formalism with native support for constraints that are handled by SMT solvers. They are useful for program analysis, as illustrated in numerous papers [2,3,5,13]. Several results from term rewriting have been lifted to constrained rewriting. We mention termination analysis [6,7,12], rewriting induction [3], completion [12] as well as runtime complexity analysis [13].

In this paper we are concerned with confluence analysis of logically constrained rewrite systems (LCTRSs for short). Only two sufficient conditions for confluence of LCTRSs are known. Kop and Nishida considered (weak) orthogonality in [8]. Orthogonality is the combination of left-linearity and the absence of critical pairs, in a weakly orthogonal system trivial critical pairs are allowed. Completion of LCTRSs is the topic of [12] and the underlying confluence condition of completion is the combination of termination and joinability of critical pairs. In this paper we add two further confluence criteria. Both of these extend known conditions for standard term rewriting to the constrained setting. The first is the combination of linearity and strong closedness of critical pairs, introduced by Huet [4]. The second, also due to [4], is the combination of left-linearity and parallel closedness of critical pairs. We also consider an extension of the latter, due to Toyama [11].

This research is supported by FWF (Austrian Science Fund) Project I 5943-N.

© The Author(s) 2023

Overview. The remainder of this paper is organized as follows. In the next section we summarize the relevant background. Section 3 recalls the existing confluence criteria for LCTRSs and some of the underlying results. The new confluence criteria for LCTRSs are reported in Sect. 4. In Sect. 5 the automation challenges we faced are described and we present our prototype implementation crest. Experimental results are reported in Sect. 6, before we conclude in Sect. 7.

2 Preliminaries

We assume familiarity with the basic notions of term rewrite systems (TRSs) [1], but shortly recapitulate terminology and notation that we use in the remainder. In particular, we recall the notion of logically constrained rewriting as defined in [3,8].

We assume a many-sorted signature $\mathcal{F}$ and a set $\mathcal{V}$ of (many-sorted) variables disjoint from $\mathcal{F}$. The signature $\mathcal{F}$ is split into term symbols from $\mathcal{F}_{\mathsf{te}}$ and theory symbols from $\mathcal{F}_{\mathsf{th}}$. The set $\mathcal{T}(\mathcal{F}, \mathcal{V})$ contains the well-sorted terms over this signature and $\mathcal{T}(\mathcal{F}_{\mathsf{th}})$ denotes the set of well-sorted ground terms that consist entirely of theory symbols. We assume a mapping $\mathcal{I}$ which assigns to every sort ι occurring in $\mathcal{F}_{\mathsf{th}}$ a carrier set $\mathcal{I}(\iota)$, and an interpretation $\mathcal{J}$ that assigns to every symbol $f \in \mathcal{F}_{\mathsf{th}}$ with sort declaration $\iota_1 \times \cdots \times \iota_n \rightarrow \kappa$ a function $f_{\mathcal{J}} \colon \mathcal{I}(\iota_1) \times \cdots \times \mathcal{I}(\iota_n) \rightarrow \mathcal{I}(\kappa)$. Moreover, for every sort ι occurring in $\mathcal{F}_{\mathsf{th}}$ we assume a set $\mathcal{V}\mathsf{al}_\iota \subseteq \mathcal{F}_{\mathsf{th}}$ of value symbols, such that all $c \in \mathcal{V}\mathsf{al}_\iota$ are constants of sort ι and $\mathcal{J}$ constitutes a bijective mapping between $\mathcal{V}\mathsf{al}_\iota$ and $\mathcal{I}(\iota)$. Thus there exists a constant symbol in $\mathcal{F}_{\mathsf{th}}$ for every value in the carrier set. The interpretation $\mathcal{J}$ naturally extends to a mapping $[\![\cdot]\!]$ from ground terms in $\mathcal{T}(\mathcal{F}_{\mathsf{th}})$ to values in $\mathcal{V}\mathsf{al} = \bigcup_{\iota \in \mathcal{D}\mathsf{om}(\mathcal{I})} \mathcal{V}\mathsf{al}_\iota$: $[\![f(t_1, \ldots, t_n)]\!] = f_{\mathcal{J}}([\![t_1]\!], \ldots, [\![t_n]\!])$ for all $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}_{\mathsf{th}})$. So every ground term in $\mathcal{T}(\mathcal{F}_{\mathsf{th}})$ has a unique value. We demand that theory symbols and term symbols overlap only on values, i.e., $\mathcal{F}_{\mathsf{te}} \cap \mathcal{F}_{\mathsf{th}} \subseteq \mathcal{V}\mathsf{al}$. A term in $\mathcal{T}(\mathcal{F}_{\mathsf{th}}, \mathcal{V})$ is called a *logical* term.

Positions are strings of positive natural numbers used to address subterms. The empty string is denoted by ϵ. We write $q \leqslant p$ and say that p is below q if $qq' = p$ for some position q', in which case $p \backslash q$ is defined to be q'. Furthermore, $q < p$ if $q \leqslant p$ and $q \neq p$. Finally, positions q and p are parallel, written as $q \parallel p$, if neither $q \leqslant p$ nor $p < q$. The set of positions of a term t is defined as $\mathcal{P}\mathsf{os}(t) = \{\epsilon\}$ if t is a variable or a constant, and as $\mathcal{P}\mathsf{os}(t) = \{\epsilon\} \cup \{iq \mid 1 \leqslant i \leqslant n \text{ and } q \in \mathcal{P}\mathsf{os}(t_i)\}$ if $t = f(t_1, \ldots, t_n)$ with $n \geqslant 1$. The subterm of t at position $p \in \mathcal{P}\mathsf{os}(t)$ is defined as $t|_p = t$ if $p = \epsilon$ and as $t|_p = t_i|_q$ if $p = iq$ and $t = f(t_1, \ldots, t_n)$. We write $s[t]_p$ for the result of replacing the subterm at position p of s with t. We write $\mathcal{P}\mathsf{os}_{\mathcal{V}}(t)$ for $\{p \in \mathcal{P}\mathsf{os}(t) \mid t|_p \in \mathcal{V}\}$ and $\mathcal{P}\mathsf{os}_{\mathcal{F}}(t)$ for $\mathcal{P}\mathsf{os}(t) \setminus \mathcal{P}\mathsf{os}_{\mathcal{V}}(t)$. The set of variables occurring in the term t is denoted by $\mathcal{V}\mathsf{ar}(t)$. A term t is linear if every variable occurs at most once in it. A substitution is a mapping σ from $\mathcal{V}$ to $\mathcal{T}(\mathcal{F}, \mathcal{V})$ such that its domain $\{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ is finite. We write $t\sigma$ for the result of applying σ to the term t.

We assume the existence of a sort bool such that $\mathcal{I}(\mathsf{bool}) = \mathbb{B} = \{\top, \bot\}$, $\mathcal{V}\mathsf{al}_{\mathsf{bool}} = \{\mathsf{true}, \mathsf{false}\}$, $[\![\mathsf{true}]\!] = \top$, and $[\![\mathsf{false}]\!] = \bot$ hold. Logical terms of sort

bool are called *constraints*. A constraint φ is *valid* if $[\![\varphi\gamma]\!] = \top$ for all substitutions γ such that $\gamma(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{V}\mathsf{ar}(\varphi)$.

A *constrained rewrite rule* is a triple $\ell \to r\ [\varphi]$ where $\ell, r \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ are terms of the same sort such that $\mathsf{root}(\ell) \in \mathcal{F}_{\mathsf{te}} \setminus \mathcal{F}_{\mathsf{th}}$ and φ is a logical term of sort bool. If $\varphi = \mathsf{true}$ then the constraint is often omitted, and the rule is denoted as $\ell \to r$. We denote the set $\mathcal{V}\mathsf{ar}(\varphi) \cup (\mathcal{V}\mathsf{ar}(r) \setminus \mathcal{V}\mathsf{ar}(\ell))$ of *logical* variables in $\rho\colon \ell \to r\ [\varphi]$ by $\mathcal{LV}\mathsf{ar}(\rho)$. We write $\mathcal{EV}\mathsf{ar}(\rho)$ for the set $\mathcal{V}\mathsf{ar}(r) \setminus (\mathcal{V}\mathsf{ar}(\ell) \cup \mathcal{V}\mathsf{ar}(\varphi))$ of variables that appear only in the right-hand side of ρ. Note that extra variables in right-hand sides are allowed, but they may only be instantiated by values. This is useful to model user input or random choice [3]. A set of constrained rewrite rules is called a *logically constrained rewrite system* (LCTRS for short).

The LCTRS $\mathcal{R}$ introduced in the example below computes the maximum of two integers.

Example 1. Before giving the rules, we need to define the term and theory symbols, the carrier sets and interpretation functions:

$$\begin{aligned}
\mathcal{F}_{\mathsf{te}} &= \{\mathsf{max}\colon \mathsf{int} \times \mathsf{int} \Rightarrow \mathsf{int}\} \cup \{\mathsf{0}, \mathsf{1}, \ldots : \mathsf{int}\} \qquad \mathcal{I}_{\mathsf{bool}} = \mathbb{B} \qquad \mathcal{I}_{\mathsf{int}} = \mathbb{Z}\\
\mathcal{F}_{\mathsf{th}} &= \{\mathsf{0}, \mathsf{1}, \ldots : \mathsf{int}\} \cup \{\mathsf{true}, \mathsf{false}\colon \mathsf{bool}\} \cup \{\neg\colon \mathsf{bool} \Rightarrow \mathsf{bool}\}\\
&\cup \{-\colon \mathsf{int} \Rightarrow \mathsf{int}\} \cup \{\wedge\colon \mathsf{bool} \times \mathsf{bool} \Rightarrow \mathsf{bool}\}\\
&\cup \{+, -\colon \mathsf{int} \times \mathsf{int} \Rightarrow \mathsf{int}\} \cup \{\leq, \geq, <, >, = : \mathsf{int} \times \mathsf{int} \Rightarrow \mathsf{bool}\}
\end{aligned}$$

The interpretations for theory symbols follow the usual semantics given in the SMT-LIB theory Ints[1] used by the SMT-LIB logic $\mathsf{QF\_LIA}$. The LCTRS $\mathcal{R}$ consists of the following constrained rewrite rules

$$\mathsf{max}(x,y) \to x\ [x \geq y] \qquad \mathsf{max}(x,y) \to y\ [y \geq x] \qquad \mathsf{max}(x,y) \to \mathsf{max}(y,x)$$

In later examples we refrain from spelling out the signature and interpretations of the theory Ints. We now define rewriting using constrained rewrite rules. LCTRSs admit two kinds of rewrite steps. Rewrite rules give rise to *rule* steps, provided the constraint of the rule is satisfied. In addition, theory calls of the form $f(v_1, \ldots, v_n)$ with $f \in \mathcal{F}_{\mathsf{th}} \setminus \mathcal{V}\mathsf{al}$ and values $v_1, \ldots, v_n$ can be evaluated in a *calculation* step. In the definition below, a substitution σ is said to *respect* a rule $\rho\colon \ell \to r\ [\varphi]$, denoted by $\sigma \vDash \rho$, if $\mathcal{D}\mathsf{om}(\sigma) = \mathcal{V}\mathsf{ar}(\ell) \cup \mathcal{V}\mathsf{ar}(r) \cup \mathcal{V}\mathsf{ar}(\varphi)$, $\sigma(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{LV}\mathsf{ar}(\rho)$, and $\varphi\sigma$ is valid. Moreover, a constraint φ is respected by σ, denoted by $\sigma \vDash \varphi$, if $\sigma(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{V}\mathsf{ar}(\varphi)$ and $\varphi\sigma$ is valid.

Definition 1. *Let $\mathcal{R}$ be an LCTRS. A* rule step $s \to_{\mathsf{ru}} t$ *satisfies $s|_p = \ell\sigma$ and $t = s[r\sigma]_p$ for some position p and constrained rewrite rule $\ell \to r\ [\varphi]$ that is respected by the substitution σ. A* calculation step $s \to_{\mathsf{ca}} t$ *satisfies $s|_p = f(v_1, \ldots, v_n)$ and $t = s[v]_p$ for some $f \in \mathcal{F}_{\mathsf{th}} \setminus \mathcal{V}\mathsf{al}$, $v_1, \ldots, v_n \in \mathcal{V}\mathsf{al}$ with $v = [\![f(v_1, \ldots, v_n)]\!]$. In this case $f(x_1, \ldots, x_n) \to y\ [y = f(x_1, \ldots, x_n)]$ with a fresh variable y is a* calculation rule. *The set of all calculation rules is denoted by $\mathcal{R}_{\mathsf{ca}}$. The relation $\to_{\mathcal{R}}$ associated with $\mathcal{R}$ is the union of $\to_{\mathsf{ru}} \cup \to_{\mathsf{ca}}$.*

[1] http://smtlib.cs.uiowa.edu/Theories/Ints.smt2.

We sometimes write $\to_{p\,|\,\rho\,|\,\sigma}$ to indicate that the rewrite step takes place at position p, using the constrained rewrite rule ρ with substitution σ.

Example 2. We have $\mathsf{max}(1+2,4) \to_{\mathcal{R}} \mathsf{max}(3,4) \to_{\mathcal{R}} \mathsf{max}(4,3) \to_{\mathcal{R}} 4$ in the LCTRS of Example 1. The first step is a calculation step. In the third step we apply the rule $\mathsf{max}(x,y) \to x\ [x \geq y]$ with substitution $\sigma = \{x \mapsto 4, y \mapsto 3\}$.

3 Confluence

In this paper we are concerned with the confluence of LCTRSs. An LCTRS $\mathcal{R}$ is *confluent* if $t \to^*_{\mathcal{R}} \cdot {}^*_{\mathcal{R}}\!\leftarrow u$ for all terms s, t and u such that $t \;{}^*_{\mathcal{R}}\!\leftarrow s \to^*_{\mathcal{R}} u$. Confluence criteria for TRSs are based on critical pairs. Critical pairs for LCTRS were introduced in [8]. The difference with the definition below is that we add dummy constraints for *extra* variables in right-hand sides of rewrite rules.

Definition 2. *An* overlap *of an LCTRS $\mathcal{R}$ is a triple $\langle \rho_1, p, \rho_2 \rangle$ with rules $\rho_1 \colon \ell_1 \to r_1\ [\varphi_1]$ and $\rho_2 \colon \ell_2 \to r_2\ [\varphi_2]$, satisfying the following conditions:*

1. *ρ_1 and ρ_2 are variable-disjoint variants of rewrite rules in $\mathcal{R} \cup \mathcal{R}_{\mathsf{ca}}$,*
2. *$p \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(\ell_2)$,*
3. *ℓ_1 and $\ell_2|_p$ are unifiable with a mgu σ such that $\sigma(x) \in \mathcal{V}\mathsf{al} \cup \mathcal{V}$ for all $x \in \mathcal{LV}\mathsf{ar}(\rho_1) \cup \mathcal{LV}\mathsf{ar}(\rho_2)$,*
4. *$\varphi_1\sigma \wedge \varphi_2\sigma$ is satisfiable, and*
5. *if $p = \epsilon$ then ρ_1 and ρ_2 are not variants, or $\mathcal{V}\mathsf{ar}(r_1) \nsubseteq \mathcal{V}\mathsf{ar}(\ell_1)$.*

In this case we call $\ell_2\sigma[r_1\sigma]_p \approx r_2\sigma\ [\varphi_1\sigma \wedge \varphi_2\sigma \wedge \psi\sigma]$ a constrained critical pair *obtained from the overlap $\langle \rho_1, p, \rho_2 \rangle$. Here*

$$\psi = \bigwedge \{x = x \mid x \in \mathcal{EV}\mathsf{ar}(\rho_1) \cup \mathcal{EV}\mathsf{ar}(\rho_2)\}$$

The set of all constrained critical pairs of $\mathcal{R}$ is denoted by $\mathsf{CCP}(\mathcal{R})$.

In the following we drop "constrained" and speak of critical pairs. The condition $\mathcal{V}\mathsf{ar}(r_1) \nsubseteq \mathcal{V}\mathsf{ar}(\ell_1)$ in the fifth condition is essential to correctly deal with extra variables in rewrite rules. The equations (ψ) added to the constraint of a critical pair save the information which variables in a critical pair were introduced by variables only occurring in the right-hand side of a rewrite rule and therefore should *only* be instantiated by values. Critical pairs as defined in [8,12] lack this information. The proof of Theorem 2 in the next section makes clear why those trivial equations are essential for our confluence criteria, see also Example 9.

Example 3. Consider the LCTRS consisting of the rule

$$\rho\colon\ \mathsf{f}(x) \to z\ [x = z\,\hat{}\,2]$$

The variable z does not occur in the left-hand side and the condition $\mathcal{V}\mathsf{ar}(r_1) \nsubseteq \mathcal{V}\mathsf{ar}(\ell_1)$ ensures that ρ overlaps with (a variant of) itself at the root position. Note that $\mathcal{R}$ is not confluent due to the non-joinable local peak $-4 \leftarrow \mathsf{f}(16) \to 4$.

Example 4. The LCTRS $\mathcal{R}$ of Example 1 admits the following critical pairs:

$$\begin{array}{ll} x \approx y\ [x \geq y \wedge y \geq x] & \langle 1, \epsilon, 2\rangle \\ x \approx \mathsf{max}(y, x)\ [x \geq y] & \langle 1, \epsilon, 3\rangle \\ y \approx \mathsf{max}(y, x)\ [y \geq x] & \langle 2, \epsilon, 3\rangle \end{array}$$

The originating overlap is given on the right, where we number the rewrite rules from left to right in Example 1.

Actually, there are three more overlaps since the position of overlap (ϵ) is the root position. Such overlaps are called *overlays* and always come in pairs. For instance, $\mathsf{max}(y, x) \approx x\ [x \geq y]$ is the critial pair originating from $\langle 3, \epsilon, 1\rangle$. For confluence criteria based on symmetric joinability conditions of critical pairs (like weak orthogonality and joinability of critical pairs for terminating systems) we need to consider just one critical pair, but this is not true for the criteria presented in the next section.

Logically constrained rewriting aims to rewrite (unconstrained) terms with constrained rules. However, for the sake of analysis, rewriting *constrained terms* is useful. In particular, since critical pairs in LCTRSs come with a constraint, confluence criteria need to consider constrained terms. The relevant notions defined below originate from [3,8].

Definition 3. *A* constrained term *is a pair $s\ [\varphi]$ of a term s and a constraint φ. Two constrained terms $s\ [\varphi]$ and $t\ [\psi]$ are* equivalent, *denoted by $s\ [\varphi] \sim t\ [\psi]$, if for every substitution γ respecting φ there is some substitution δ that respects ψ such that $s\gamma = t\delta$, and vice versa. Let $\mathcal{R}$ be an LCTRS and $s\ [\varphi]$ a constrained term. If $s|_p = \ell\sigma$ for some constrained rewrite rule $\rho\colon \ell \to r\ [\psi]$, position p, and substitution σ such that $\sigma(x) \in \mathcal{V}\mathsf{al} \cup \mathcal{V}\mathsf{ar}(\varphi)$ for all $x \in \mathcal{LV}\mathsf{ar}(\rho)$, φ is satisfiable and $\varphi \Rightarrow \psi\sigma$ is valid then*

$$s\ [\varphi] \to_{\mathsf{ru}} s[r\sigma]_p\ [\varphi]$$

is a rule step. *If $s|_p = f(s_1, \ldots, s_n)$ with $f \in \mathcal{F}_{\mathsf{th}} \setminus \mathcal{F}_{\mathsf{te}}$ and $s_1, \ldots, s_n \in \mathcal{V}\mathsf{al} \cup \mathcal{V}\mathsf{ar}(\varphi)$ then*

$$s\ [\varphi] \to_{\mathsf{ca}} s[x]_p\ [\varphi \wedge x = f(s_1, \ldots, s_n)]$$

is a calculation step. *Here x is a fresh variable. We write $\to_{\mathcal{R}}$ for $\to_{\mathsf{ru}} \cup \to_{\mathsf{ca}}$ and the rewrite relation $\overset{\sim}{\to}_{\mathcal{R}}$ on constrained terms is defined as $\sim \cdot \to_{\mathcal{R}} \cdot \sim$.*

Positions in connection with $\overset{\sim}{\to}_{\mathcal{R}}$ steps always refer to the underlying steps in $\to_{\mathcal{R}}$. We give an example of constrained rewriting.

Example 5. Consider again the LCTRS $\mathcal{R}$ of Example 1. We have

$$\begin{aligned} \mathsf{max}(x + y, \mathsf{6})\ [x \geq \mathsf{2} \wedge y \geq \mathsf{4}] &\to_{\mathcal{R}} \mathsf{max}(z, \mathsf{6})\ [x \geq \mathsf{2} \wedge y \geq \mathsf{4} \wedge z = x + y] \\ &\to_{\mathcal{R}} z\ [x \geq \mathsf{2} \wedge y \geq \mathsf{4} \wedge z = x + y] \end{aligned}$$

The first step is a calculation step. The second step is a rule step using the rule $\mathsf{max}(x, y) \to x\ [x \geq y]$ with the substitution $\sigma = \{x \mapsto z, y \mapsto \mathsf{6}\}$. Note that the constraint $(x \geq \mathsf{2} \wedge y \geq \mathsf{4} \wedge z = x + y) \Rightarrow z \geq \mathsf{6}$ is valid.

Definition 4. *A critical pair $s \approx t\ [\varphi]$ is* trivial *if $s\sigma = t\sigma$ for every substitution σ with $\sigma \vDash \varphi$.*[2] *A left-linear LCTRS having only trivial critical pairs is called* weakly orthogonal. *A left-linear TRS without critical pairs is called* orthogonal.

The following result is from [8].

Theorem 1. *Weakly orthogonal LCTRS are confluent.* □

Example 6. The following left-linear LCTRS computes the Ackermann function using term symbols from $\mathcal{F}_{\mathsf{te}} = \{\mathsf{ack} : \mathsf{int} \times \mathsf{int} \Rightarrow \mathsf{int}\} \cup \{\mathsf{0}, \mathsf{1}, \cdots : \mathsf{int}\}$ and the same theory symbols, carrier sets and interpretations as in Example 1:

$$\begin{aligned}
\mathsf{ack}(\mathsf{0}, n) &\to n + \mathsf{1}\ [n \geq \mathsf{0}] \\
\mathsf{ack}(m, \mathsf{0}) &\to \mathsf{ack}(m - \mathsf{1}, \mathsf{1})\ [m > \mathsf{0}] \\
\mathsf{ack}(m, n) &\to \mathsf{ack}(m - \mathsf{1}, \mathsf{ack}(m, n - \mathsf{1}))\ [m > \mathsf{0} \wedge n > \mathsf{0}] \\
\mathsf{ack}(m, n) &\to \mathsf{0}\ [m < \mathsf{0} \vee n < \mathsf{0}]
\end{aligned}$$

Since the conjunction of any two constraints is unsatisfiable, $\mathcal{R}$ lacks critical pairs. Hence $\mathcal{R}$ is confluent by Theorem 1.

The following result is proved in [12] and forms the basis of completion of LCTRSs.

Lemma 1. *Let $\mathcal{R}$ be an LCTRS. If $t\ {}_{\mathcal{R}}\!\leftarrow s \to_{\mathcal{R}} u$ then $t \downarrow_{\mathcal{R}} u$ or $t \xleftrightarrow[\mathsf{CCP}(\mathcal{R})]{} u$.* □

In combination with Newman's Lemma, the following confluence criterion is obtained.

Corollary 1. *A terminating LCTRS is confluent if all critical pairs are joinable.*

This is less obvious than it seems. Joinability of a critical pair $s \approx t\ [\varphi]$ cannot simply be defined as $s\ [\varphi] \xrightarrow{\sim}{}^{*}_{\mathcal{R}} \cdot {}^{*}_{\mathcal{R}}\!\xleftarrow{\sim} t\ [\varphi]$, as the following example shows.

Example 7. Consider the terminating LCTRS $\mathcal{R}$ consisting of the rewrite rules

$$\mathsf{f}(x, y) \to \mathsf{g}(x, \mathsf{1} + \mathsf{1}) \qquad\qquad \mathsf{h}(\mathsf{f}(x, y)) \to \mathsf{h}(\mathsf{g}(y, \mathsf{1} + \mathsf{1}))$$

The single critical pair $\mathsf{h}(\mathsf{g}(x, \mathsf{1} + \mathsf{1})) \approx \mathsf{h}(\mathsf{g}(y, \mathsf{1} + \mathsf{1}))$ should not be joinable because $\mathcal{R}$ is not confluent, but we do have

$$\begin{aligned}
&\mathsf{h}(\mathsf{g}(x, \mathsf{1} + \mathsf{1})) \to_{\mathsf{ca}} \mathsf{h}(\mathsf{g}(x, z))\ [z = \mathsf{1} + \mathsf{1}] \sim \mathsf{h}(\mathsf{g}(y, v))\ [v = \mathsf{1} + \mathsf{1}] \\
&\mathsf{h}(\mathsf{g}(y, \mathsf{1} + \mathsf{1})) \to_{\mathsf{ca}} \mathsf{h}(\mathsf{g}(y, v))\ [v = \mathsf{1} + \mathsf{1}]
\end{aligned}$$

due to the equivalence relation $\sim$ on constrained terms; since x and y do not appear in the constraints, there is no demand that they must be instantiated with values.

[2] The triviality condition in [8] is wrong. Here we use the corrected version in an update of [8] announced on Cynthia Kop's website (accessible at https://www.cs.ru.nl/~cynthiakop/frocos13.pdf).

The solution is not to treat the two sides of a critical pair in isolation but define joinability based on rewriting constrained term pairs. So we view the symbol $\approx$ in a constrained equation $s \approx t\ [\varphi]$ as a binary constructor symbol such that the constrained equation can be viewed as a constrained term. Steps in s take place at positions $\geqslant 1$ whereas steps in t use positions $\geqslant 2$. The same is done in completion of LCTRSs [12].

Definition 5. *We call a constrained equation $s \approx t\ [\varphi]$* trivial *if $s\sigma = t\sigma$ for any substitution σ with $\sigma \vDash \varphi$. A critical pair $s \approx t\ [\varphi]$ is* joinable *if $s \approx t\ [\varphi] \overset{\sim}{\rightarrow}^*_{\mathcal{R}} u \approx v\ [\psi]$ and $u \approx v\ [\psi]$ is trivial.*

We revisit Example 7.

Example 8. For the critical pair in Example 7 we obtain

$$\begin{aligned}
&\mathsf{h}(\mathsf{g}(x,\mathsf{1}+\mathsf{1})) \approx \mathsf{h}(\mathsf{g}(y,\mathsf{1}+\mathsf{1}))\\
&\qquad \rightarrow_{\mathsf{ca}} \mathsf{h}(\mathsf{g}(x,v)) \approx \mathsf{h}(\mathsf{g}(y,\mathsf{1}+\mathsf{1}))\ [v = \mathsf{1}+\mathsf{1}]\\
&\qquad \rightarrow_{\mathsf{ca}} \mathsf{h}(\mathsf{g}(x,v)) \approx \mathsf{h}(\mathsf{g}(y,z))\ [v = \mathsf{1}+\mathsf{1} \wedge z = \mathsf{1}+\mathsf{1}]
\end{aligned}$$

The substitution $\sigma = \{v \mapsto \mathsf{2}, z \mapsto \mathsf{2}\}$ respects the constraint $v = \mathsf{1}+\mathsf{1} \wedge z = \mathsf{1}+\mathsf{1}$ but does not equate $\mathsf{h}(\mathsf{g}(x,v))$ and $\mathsf{h}(\mathsf{g}(y,z))$.

The converse of Corollary 1 also holds, but note that in contrast to TRSs, joinability of critical pairs is not a decidable criterion for terminating LCTRSs, due to the undecidable triviality condition. Moreover, for the converse to hold, it is essential that critical pairs contain the trivial equations ψ in Definition 2.

Example 9. Consider the LCTRS $\mathcal{R}$ consisting of the rules

$$\mathsf{f}(x) \rightarrow \mathsf{g}(y) \qquad\qquad \mathsf{g}(y) \rightarrow \mathsf{a}\ [y = y]$$

which admits the critical pair $\mathsf{g}(y) \approx \mathsf{g}(y')\ [y = y \wedge y' = y']$ originating from the overlap $\langle \mathsf{f}(x) \rightarrow \mathsf{g}(y), \epsilon, \mathsf{f}(x') \rightarrow \mathsf{g}(y')\rangle$. This critical pair is joinable as y and y' are restricted to values and thus both sides rewrite to a using the second rule. As $\mathcal{R}$ is also terminating, it is confluent by Corollary 1. If we were to drop ψ in Definition 2, we would obtain the non-joinable critical pair $\mathsf{g}(y) \approx \mathsf{g}(y')$ instead and wrongly conclude non-confluence.

4 Main Results

We start with extending a confluence result of Huet [4] for linear TRSs. Below we write $\rightarrow_{\geqslant p}$ to indicate that the position of the contracted redex in the step is below position p.

Definition 6. *A critical pair $s \approx t\ [\varphi]$ is* strongly closed *if*

1. *$s \approx t\ [\varphi] \overset{\sim}{\rightarrow}^*_{\geqslant 1} \cdot \overset{\sim}{\rightarrow}^=_{\geqslant 2} u \approx v\ [\psi]$ for some trivial $u \approx v\ [\psi]$, and*
2. *$s \approx t\ [\varphi] \overset{\sim}{\rightarrow}^*_{\geqslant 2} \cdot \overset{\sim}{\rightarrow}^=_{\geqslant 1} u \approx v\ [\psi]$ for some trivial $u \approx v\ [\psi]$.*

A binary relation $\to$ on terms is *strongly confluent* if $t \to^* \cdot {}^{=}\!\!\leftarrow u$ for all terms s, t and u with $t \leftarrow s \to u$. (By symmetry, also $t \to^= \cdot {}^*\!\!\leftarrow u$ is required.) Strong confluence is a well-known sufficient condition for confluence. Huet [4] proved that linear TRSs are strongly confluent if all critical pairs are strongly closed. Below we extend this result to LCTRSs, using the above definition of strongly closed constrained critical pairs.

Theorem 2. *A linear LCTRS is strongly confluent if all its critical pairs are strongly closed.*

We give full proof details in order to illustrate the complications caused by constrained rewrite rules. The following result from [12] plays an important role.

Lemma 2. *Suppose $s \approx t\ [\varphi] \xrightarrow{\sim}_p u \approx v\ [\psi]$ and $\gamma \vDash \varphi$. If $p \geqslant 1$ then $s\gamma \to u\delta$ and $t\gamma = v\delta$ for some substitution δ with $\delta \vDash \psi$. If $p \geqslant 2$ then $s\gamma = u\delta$ and $t\gamma \to v\delta$ for some substitution δ with $\delta \vDash \psi$.* □

Proof (of Theorem 2). Consider an arbitrary local peak

$$t \leftarrow_{p_1\,|\,\rho_1\,|\,\sigma_1} s \to_{p_2\,|\,\rho_2\,|\,\sigma_2} u$$

with rewrite rules $\rho_1\colon \ell_1 \to r_1\ [\varphi_1]$ and $\rho_2\colon \ell_2 \to r_2\ [\varphi_2]$ from $\mathcal{R} \cup \mathcal{R}_{\mathsf{ca}}$. We may assume that ρ_1 and ρ_2 have no variables in common, and consequently $\mathcal{D}\mathsf{om}(\sigma_1) \cap \mathcal{D}\mathsf{om}(\sigma_2) = \varnothing$. We have $s|_{p_1} = \ell_1\sigma_1$, $t = s[r_1\sigma_1]_{p_1}$ and $\sigma_1 \vDash \varphi_1$. Likewise, $s|_{p_2} = \ell_2\sigma_2$, $u = s[r_2\sigma_2]_{p_2}$ and $\sigma_2 \vDash \varphi_2$. If $p_1 \parallel p_2$ then

$$t \to_{p_2\,|\,\rho_2\,|\,\sigma_2} t[r_2\sigma_2]_{p_2} = u[r_1\sigma_1]_{p_1} \leftarrow_{p_1\,|\,\rho_1\,|\,\sigma_1} u$$

Hence both $t \to^* \cdot {}^{=}\!\!\leftarrow u$ and $t \to^= \cdot {}^*\!\!\leftarrow u$. If p_1 and p_2 are not parallel then $p_1 \leqslant p_2$ or $p_2 < p_1$. Without loss of generality, we consider $p_1 \leqslant p_2$. Let $q = p_2 \backslash p_1$. We do a case analysis on whether or not $q \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(\ell_1)$.

- First suppose $q \notin \mathcal{P}\mathsf{os}_{\mathcal{F}}(\ell_1)$. Let $q = q_1q_2$ such that $q_1 \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(\ell_1)$ and let x be the variable in ℓ_1 at position q_1. We have $\ell_2\sigma_2 = x\sigma_1|_{q_2}$ and thus $\sigma_1(x) \notin \mathcal{V}\mathsf{al}$. Define the substitution σ_1' as follows:

$$\sigma_1'(y) = \begin{cases} x\sigma_1[r_2\sigma_2]_{q_2} & \text{if } y = x \\ \sigma_1(y) & \text{otherwise} \end{cases}$$

  We show $t \to^= s[r_1\sigma_1']_{p_1} \leftarrow u$, which yields $t \to^* \cdot {}^{=}\!\!\leftarrow u$ and $t \to^= \cdot {}^*\!\!\leftarrow u$. Since $\mathcal{R}$ is left-linear, $\ell_1\sigma_1' = \ell_1\sigma_1[x\sigma_1']_{q_1} = \ell_1\sigma_1[x\sigma_1[r_2\sigma_2]_{q_2}]_{q_1} = \ell_1\sigma_1[r_2\sigma_2]_q$ and thus $u = s[r_2\sigma_2]_{p_2} = s[\ell_1\sigma_1[r_2\sigma_2]_q]_{p_1} = s[\ell_1\sigma_1']_{p_1}$. If we can show $\sigma_1' \vDash \rho_1$ then $u \to s[r_1\sigma_1']_{p_1}$. Consider an arbitrary variable $y \in \mathcal{LV}\mathsf{ar}(\rho_1)$. If $y \neq x$ then $\sigma_1'(y) = \sigma_1(y) \in \mathcal{V}\mathsf{al}$ since $\sigma_1 \vDash \rho_1$. If $y = x$ then $x \in \mathcal{V}\mathsf{ar}(\varphi)$ since $x \in \mathcal{V}\mathsf{ar}(\ell_1)$. However, this contradicts $\sigma_1 \vDash \rho_1$ as $\sigma_1(x) \notin \mathcal{V}\mathsf{al}$. So $\sigma_1'(y) = \sigma_1(y)$ for all $y \in \mathcal{LV}\mathsf{ar}(\rho_1)$ and thus $\sigma_1' \vDash \rho_1$ is an immediate consequence of $\sigma_1 \vDash \rho_1$. It remains to show $t \to^= s[r_1\sigma_1']_{p_1}$. If $x \notin \mathcal{V}\mathsf{ar}(r_1)$ then $r_1\sigma_1' = r_1\sigma_1$ and thus $t = s[r_1\sigma_1']_{p_1}$. If $x \in \mathcal{V}\mathsf{ar}(r_1)$ then there exists a unique position

$q' \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(r_1)$ such that $r_1|_{q'} = x$, due to the right-linearity of $\mathcal{R}$. Hence $r_1\sigma_1' = r_1\sigma_1[x\sigma_1[r_2\sigma_2]_{q_2}]_{q'} = r_1\sigma_1[r_2\sigma_2]_{q'q_2}$. Since $r_1\sigma_1|_{q'q_2} = \ell_2\sigma_2$ we obtain $t = s[r_1\sigma_1]_{p_1} \to_{p_1q'q_2\,|\,\rho_2\,|\,\sigma_2} s[r_1\sigma_1']_{p_1}$ as desired.

– Next suppose $q \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(\ell_1)$. The substitution $\sigma' = \sigma_1 \cup \sigma_2$ satisfies $\ell_1|_q\sigma' = \ell_1|_q\sigma_1 = \ell_2\sigma_2 = \ell_2\sigma'$ and thus is a unifier of $\ell_1|_q$ and ℓ_2. Since $\sigma_1 \vDash \rho_1$ and $\sigma_2 \vDash \rho_2$, $\sigma'(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{LV}\mathsf{ar}(\rho_1) \cup \mathcal{LV}\mathsf{ar}(\rho_2)$. Let σ be an mgu of $\ell_1|_q$ and ℓ_2. Since σ is at least as general as σ', $\sigma(x) \in \mathcal{V}\mathsf{al} \cup \mathcal{V}$ for all $x \in \mathcal{LV}\mathsf{ar}(\rho_1) \cup \mathcal{LV}\mathsf{ar}(\rho_2)$. Since $\varphi_1\sigma' = \varphi_1\sigma_1$ and $\varphi_2\sigma' = \varphi_2\sigma_2$ are valid, $\varphi_1\sigma \wedge \varphi_2\sigma$ is satisfiable. Hence conditions 1, 2, 3 and 4 in Definition 2 hold for the triple $\langle \rho_2, q, \rho_1\rangle$. If condition 5 is *not* fulfilled then $q = \epsilon$ (and thus $p_1 = p_2$), ρ_2 and ρ_1 are variants, and $\mathcal{V}\mathsf{ar}(r_2) \subseteq \mathcal{V}\mathsf{ar}(\ell_2)$ (and thus also $\mathcal{V}\mathsf{ar}(r_1) \subseteq \mathcal{V}\mathsf{ar}(\ell_1)$). Hence $\ell_1\sigma_1 = \ell_2\sigma_2$ and $r_1\sigma_1 = r_2\sigma_2$, and thus $t = u$. In the remaining case condition 5 holds and hence $\langle \rho_2, q, \rho_1\rangle$ is an overlap. By definition, $\ell_1\sigma[r_2\sigma]_q \approx r_1\sigma\ [\varphi_2\sigma \wedge \varphi_1\sigma \wedge \psi\sigma]$ with

$$\psi = \bigwedge \{x = x \mid x \in \mathcal{EV}\mathsf{ar}(\rho_1) \cup \mathcal{EV}\mathsf{ar}(\rho_2)\}$$

is a critical pair. To simplify the notation, we abbreviate $\ell_1\sigma[r_2\sigma]_q$ to s', $r_1\sigma$ to t', and $\varphi_2\sigma \wedge \varphi_1\sigma \wedge \psi\sigma$ to φ'. Critical pairs are strongly closed by assumption, and thus both

1. $s' \approx t'\ [\varphi']\ \overset{\sim}{\to}{}^{*}_{\geqslant 1} \cdot \overset{\sim}{\to}{}^{=}_{\geqslant 2}\ u \approx v\ [\psi']$ for some trivial $u \approx v\ [\psi']$, and
2. $s' \approx t'\ [\varphi']\ \overset{\sim}{\to}{}^{*}_{\geqslant 2} \cdot \overset{\sim}{\to}{}^{=}_{\geqslant 1}\ u \approx v\ [\psi']$ for some trivial $u \approx v\ [\psi']$.

Let γ be the substitution such that $\sigma\gamma = \sigma'$. We claim that γ respects φ'. So let $x \in \mathcal{V}\mathsf{ar}(\varphi') = \mathcal{V}\mathsf{ar}(\varphi_2\sigma \wedge \varphi_1\sigma \wedge \psi\sigma)$. We have

$$\mathcal{LV}\mathsf{ar}(\rho_1) = \mathcal{V}\mathsf{ar}(\varphi_1) \cup \mathcal{EV}\mathsf{ar}(\rho_1) \qquad \mathcal{LV}\mathsf{ar}(\rho_2) = \mathcal{V}\mathsf{ar}(\varphi_2) \cup \mathcal{EV}\mathsf{ar}(\rho_2)$$

Together with $\mathcal{V}\mathsf{ar}(\psi) = \mathcal{EV}\mathsf{ar}(\rho_1) \cup \mathcal{EV}\mathsf{ar}(\rho_2)$ we obtain

$$\mathcal{LV}\mathsf{ar}(\rho_1) \cup \mathcal{LV}\mathsf{ar}(\rho_2) = \mathcal{V}\mathsf{ar}(\varphi_1) \cup \mathcal{V}\mathsf{ar}(\varphi_2) \cup \mathcal{V}\mathsf{ar}(\psi)$$

Since $\sigma'(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{LV}\mathsf{ar}(\rho_1) \cup \mathcal{LV}\mathsf{ar}(\rho_2)$, we obtain $\gamma(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{V}\mathsf{ar}(\varphi')$ and thus $\gamma \vDash \varphi'$. At this point repeated applications of Lemma 2 to the constrained rewrite sequence in item 1 yields a substitution δ respecting ψ' such that $s'\gamma \to^* u\delta$ and $t'\gamma = v\delta$. Since $u \approx v\ [\psi']$ is trivial, $u\delta = v\delta$ and hence $s'\gamma \to^* \cdot {}^{=}\!\!\leftarrow t'\gamma$. Likewise, $s'\gamma \to^= \cdot {}^{*}\!\!\leftarrow t'\gamma$ is obtained from item 2. We have

$$s'\gamma = (\ell_1\sigma[r_2\sigma]_q)\gamma = \ell_1\sigma'[r_2\sigma']_q = \ell_1\sigma_1[r_2\sigma_2]_q \qquad t'\gamma = r_1\sigma' = r_1\sigma_1$$

Moreover, $t = s[r_1\sigma_1]_{p_1} = s[t'\gamma]_{p_1}$ and $u = s[\ell_1\sigma_1[r_2\sigma_2]_q]_{p_1} = s[s'\gamma]_{p_1}$. Since rewriting is closed under contexts, we obtain $u \to^* \cdot {}^{=}\!\!\leftarrow t$ and $u \to^= \cdot {}^{*}\!\!\leftarrow t$. This completes the proof. □

Example 10. Consider the LCTRS $\mathcal{R}$ of Example 1 and its critical pairs in Example 4. The critical pair

$$x \approx \mathsf{max}(y, x)\ [x \geq y]$$

is not trivial, so Theorem 1 is not applicable and the rule $\mathsf{max}(x, y) \to \mathsf{max}(y, x)$ precludes the use of Corollary 1 to infer confluence. We do have

$$x \approx \mathsf{max}(y, x)\ [x \geq y] \xrightarrow{\geqslant 2} x \approx x\ [x \geq y]$$

by applying the rule $\mathsf{max}(x, y) \to y\ [y \geq x]$ and the resulting constrained equation $x \approx x\ [x \geq y]$ is obviously trivial. The same reasoning applies to the critical pair $y \approx \mathsf{max}(y, x)\ [y \geq x]$. The first critical pair $x \approx y\ [x \geq y \wedge y \geq x]$ in Example 4 is trivial since any (value) substitution satisfying its constraint $x \geq y \wedge y \geq x$ equates x and y. By symmetry, all critical pairs of $\mathcal{R}$ are strongly closed. Since $\mathcal{R}$ is linear, confluence follows from Theorem 2.

The second main result is the extension of Huet's parallel closedness condition on critical pairs in left-linear TRSs [4] to LCTRSs. To this end, we first define parallel rewriting for LCTRSs.

Definition 7. *Let $\mathcal{R}$ be an LCTRS. The relation $\twoheadrightarrow\!\!\!\!\!\!\!+\ _{\mathcal{R}}$ is defined on terms inductively as follows:*

1. *$x \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} x$ for all variables x,*
2. *$f(s_1, \dots, s_n) \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} f(t_1, \dots, t_n)$ if $s_i \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} t_i$ for all $1 \leqslant i \leqslant n$,*
3. *$\ell\sigma \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} r\sigma$ with $\ell \to r\ [\varphi] \in \mathcal{R}$ and $\sigma \vDash \ell \to r\ [\varphi]$,*
4. *$f(v_1, \dots, v_n) \mathrel{+\!\!\!\!\!\twoheadrightarrow} v$ with $f \in \mathcal{F}_{\mathsf{th}} \setminus \mathcal{V}\mathsf{al}$, $v_1, \dots, v_n \in \mathcal{V}\mathsf{al}$ and $v = [\![f(v_1, \dots, v_n)]\!]$.*

We write $\mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\geqslant p}$ to indicate that all positions of contracted redexes in the parallel step are below p. In the next definition we add constraints to parallel rewriting.

Definition 8. *Let $\mathcal{R}$ be an LCTRS. The relation $\mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}}$ is defined on constrained terms inductively as follows:*

1. *$x\ [\varphi] \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} x\ [\varphi]$ for all variables x,*
2. *$f(s_1, \dots, s_n)\ [\varphi] \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} f(t_1, \dots, t_n)\ [\varphi \wedge \psi]$ if $s_i\ [\varphi] \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} t_i\ [\varphi \wedge \psi_i]$ for all $1 \leqslant i \leqslant n$ and $\psi = \psi_1 \wedge \cdots \wedge \psi_n$,*
3. *$\ell\sigma\ [\varphi] \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} r\sigma\ [\varphi]$ with $\rho\colon \ell \to r\ [\omega] \in \mathcal{R}$, $\sigma(x) \in \mathcal{V}\mathsf{al} \cup \mathcal{V}\mathsf{ar}(\varphi)$ for all $x \in \mathcal{LV}\mathsf{ar}(\rho)$, φ is satisfiable and $\varphi \Rightarrow \omega\sigma$ is valid,*
4. *$f(v_1, \dots, v_n)\ [\varphi] \mathrel{+\!\!\!\!\!\twoheadrightarrow} v\ [\varphi \wedge v = f(v_1, \dots, v_n)]$ with $v_1, \dots, v_n \in \mathcal{V}\mathsf{al} \cup \mathcal{V}\mathsf{ar}(\varphi)$, $f \in \mathcal{F}_{\mathsf{th}} \setminus \mathcal{V}\mathsf{al}$ and v is a fresh variable.*

Here we assume that different applications to case 4 result in different fresh variables. The constraint ψ in case 2 collects the assignments introduced in earlier applications of case 4. (If there are none, $\psi = \mathsf{true}$ is omitted.) The same holds for $\psi_1, \dots, \psi_n$. We write $\overset{\sim}{\mathrel{+\!\!\!\!\!\twoheadrightarrow}}$ for the relation $\sim \cdot \mathrel{+\!\!\!\!\!\twoheadrightarrow}_{\mathcal{R}} \cdot \sim$.

In light of the earlier developments, the following definition is the obvious adaptation of parallel closedness for LCTRSs.

Definition 9. *A critical pair $s \approx t\ [\varphi]$ is* parallel closed *if*

$$s \approx t\ [\varphi] \mathrel{\overset{\sim}{+\!\!\!\!\!\!\rightarrow}}_{\geqslant 1} u \approx v\ [\psi]$$

for some trivial $u \approx v\ [\psi]$.

Note that the right-hand side t of the constrained equation $s \approx t\ [\varphi]$ may change due to the equivalence relation $\sim$, cf. the statement of Lemma 2.

Theorem 3. *A left-linear LCTRS is confluent if its critical pairs are parallel closed.*

To prove this result, we adapted the formalized proof presented in [10] to the constrained setting. The required changes are very similar to the ones in the proof of Theorem 2.

Example 11. Consider the LCTRS $\mathcal{R}$ with rules

$$\begin{aligned} \mathsf{f}(x,y) &\to \mathsf{g}(\mathsf{a}, y+y)\ [y \geq x \wedge y = 1] & \mathsf{a} &\to \mathsf{b} \\ \mathsf{h}(\mathsf{f}(x,y)) &\to \mathsf{h}(\mathsf{g}(\mathsf{b},2))\ [x \geq y] & \mathsf{g}(x,y) &\to \mathsf{g}(y,x) \end{aligned}$$

The single critical pair $\mathsf{h}(\mathsf{g}(\mathsf{a}, y+y)) \approx \mathsf{h}(\mathsf{g}(\mathsf{b},2))\ [y \geq x \wedge y = 1 \wedge x \geq y]$ is parallel closed:

$$\begin{aligned} &\mathsf{h}(\mathsf{g}(\mathsf{a}, y+y)) \approx \mathsf{h}(\mathsf{g}(\mathsf{b},2))\ [y \geq x \wedge y = 1 \wedge x \geq y] \\ &\qquad \mathrel{+\!\!\!\!\!\!\rightarrow}_{\geqslant 1} \mathsf{h}(\mathsf{g}(\mathsf{b}, z)) \approx \mathsf{h}(\mathsf{g}(\mathsf{b},2))\ [y \geq x \wedge y = 1 \wedge x \geq y \wedge z = y+y] \end{aligned}$$

and the obtained equation is trivial. Hence $\mathcal{R}$ is confluent by Theorem 3. Note that the earlier confluence criteria do not apply.

We also consider the extension of Huet's result by Toyama [11], which has a less restricted joinability condition on critical pairs stemming from overlapping rules at the root position. Such critical pairs are called *overlays* whereas critical pairs originating from overlaps $\langle \rho_1, p, \rho_2 \rangle$ with $p > \epsilon$ are called *inner* critical pairs.

Definition 10. *An LCTRS $\mathcal{R}$ is almost parallel-closed if every inner critical pair is parallel closed and every overlay $s \approx t\ [\varphi]$ satisfies*

$$s \approx t\ [\varphi] \mathrel{\overset{\sim}{+\!\!\!\!\!\!\rightarrow}}_{\geqslant 1} \cdot \overset{\sim}{\rightarrow}^{*}_{\geqslant 2} u \approx v\ [\psi]$$

for some trivial $u \approx v\ [\psi]$.

Theorem 4. *Left-linear almost parallel-closed LCTRSs are confluent.*

Again, the formalized proof of the corresponding result for plain TRSs in [10] can be adapted to the constrained setting.

Example 12. Consider the following variation of the LCTRS $\mathcal{R}$ in Example 11:

$$\begin{aligned} \mathsf{f}(x,y) &\to \mathsf{g}(\mathsf{a}, y+y)\ [y \geq x \wedge y = 1] & \mathsf{a} &\to \mathsf{b} \\ \mathsf{f}(x,y) &\to \mathsf{g}(\mathsf{b},2)\ [x \geq y] & \mathsf{g}(x,y) &\to \mathsf{g}(y,x) \end{aligned}$$

The overlay $\mathsf{g}(\mathsf{b},2) \approx \mathsf{g}(\mathsf{a}, y+y)\ [x \geq y \wedge y \geq x \wedge y = 1]$ is not parallel closed but one readily confirms that the condition in Definition 10 applies.

5 Automation

As it is very inconvenient and tedious to test by hand if an LCTRS satisfies one of the confluence criteria presented in the preceding sections, we provide an implementation. The natural choice would be to extend the existing tool Ctrl [9] because it is currently the only tool capable of analyzing confluence of LCTRSs. However, Ctrl is not actively maintained and not very well documented, so we decided to develop a new tool for the analysis of LCTRSs. Our tool is called crest (constrained rewriting software). It is written in Haskell, based on the Haskell `term-rewriting`[3] library and allows the logics `QF_LIA`, `QF_NIA`, `QF_LRA`.

The input format of crest is described on its website.[4] After parsing the input, crest checks that the resulting LCTRS is well-typed. Missing sort information is inferred. Next it is checked concurrently whether one of the implemented confluence criteria applies. crest supports (weak) orthogonality, strong closedness and (almost) parallel closedness. The tool outputs the computed critical pairs and a "proof" describing how these are closed, based on the first criterion that reports a YES result. Below we describe some of the challenges that one faces when automating the confluence criteria presented in the preceding sections.

First of all, how can we determine whether a constrained critical pair or more generally a constrained equation $s \approx t\ [\varphi]$ is trivial? The following result explains how this can be solved by an SMT solver.

Definition 11. *Given a constrained equation $s \approx t\ [\varphi]$, the formula $T(s,t,\varphi)$ is inductively defined as follows:*

$$T(s,t,\varphi) = \begin{cases} \mathsf{true} & \text{if } s = t \\ s = t & \text{if } s,t \in \mathcal{V}\mathsf{al} \cup \mathcal{V}\mathsf{ar}(\varphi) \\ \bigwedge_{i=1}^{n} T(s_i,t_i,\varphi) & \text{if } s = f(s_1,\dots,s_n) \text{ and } t = f(t_1,\dots,t_n) \\ \mathsf{false} & \text{otherwise} \end{cases}$$

Lemma 3. *A constrained equation $s \approx t\ [\varphi]$ is trivial if and only if the formula $\varphi \implies T(s,t,\varphi)$ is valid.*

Proof. First suppose $\varphi \implies T(s,t,\varphi)$ is valid. Let σ be a substitution with $\sigma \vDash \varphi$. Since $\sigma(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{V}\mathsf{ar}(\varphi)$, we can apply σ to the formula $\varphi \implies T(s,t,\varphi)$. We obtain $[\![\varphi\sigma]\!] = \top$ from $\sigma \vDash \varphi$. Hence also $[\![T(s,t,\varphi)\sigma]\!] = \top$. Since $T(s,t,\varphi)$ is a conjunction, the final case in the definition of $T(s,t,\varphi)$ is not used. Hence $\mathcal{P}\mathsf{os}(s) = \mathcal{P}\mathsf{os}(t)$, $s(p) = t(p)$ for all internal positions p in s and t, and $s|_p\sigma = t|_p\sigma$ for all leaf positions p in s and t. Consequently, $s\sigma = t\sigma$. This concludes the triviality proof of $s \approx t\ [\varphi]$.

For the only if direction, suppose $s \approx t\ [\varphi]$ is trivial. Note that the variables appearing in the formula $\varphi \implies T(s,t,\varphi)$ are those of φ. Let σ be an arbitrary

[3] https://hackage.haskell.org/package/term-rewriting-0.4.0.2.
[4] http://cl-informatik.uibk.ac.at/software/crest/.

assignment such that $[\![\varphi\sigma]\!] = \top$. We need to show $[\![T(s,t,\varphi)\sigma]\!] = \top$. We can view σ as a substitution with $\sigma(x) \in \mathcal{V}\mathsf{al}$ for all $x \in \mathcal{V}\mathsf{ar}(\varphi)$. We have $\sigma \vDash \varphi$ and thus $s\sigma = t\sigma$ by the triviality of $s \approx t\ [\varphi]$. Hence $T(s,t,\varphi)$ is a conjunction of equations between values and variables in φ, which are turned into identities by σ. Hence $[\![T(s,t,\varphi)\sigma]\!] = \top$ as desired. □

The second challenge is how to implement rewriting on constrained equations in particular, how to deal with the equivalence relation $\sim$ defined in Definition 3.

Example 13. The LCTRS $\mathcal{R}$

$$\mathsf{f}(x) \to z\ [z = \mathsf{3}] \qquad \mathsf{g}(\mathsf{f}(x)) \to \mathsf{a} \qquad \mathsf{g}(\mathsf{3}) \to \mathsf{a}$$

over the integers admits two critical pairs:

$$z \approx z'\ [z = \mathsf{3} \wedge z' = \mathsf{3}] \qquad \mathsf{g}(z) \approx \mathsf{a}\ [z = \mathsf{3}]$$

The first one is trivial, but to join the second one, an initial equivalence step is required:

$$\mathsf{g}(z) \approx \mathsf{a}\ [z = \mathsf{3}] \sim \mathsf{g}(\mathsf{3}) \approx \mathsf{a}\ [z = \mathsf{3}] \to \mathsf{a} \approx \mathsf{a}\ [z = \mathsf{3}]$$

The transformation introduced below avoids having to look for an initial equivalence step before a rule becomes applicable.

Definition 12. *Let $\mathcal{R}$ be an LCTRS. Given a term $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$, we replace values in t by fresh variables and return the modified term together with the constraint that collects the bindings:*

$$\mathsf{tf}(t) = \begin{cases} (t, \mathsf{true}) & \textit{if } t \in \mathcal{V} \\ (z, z = t) & \textit{if } t \in \mathcal{V}\mathsf{al} \textit{ and } z \textit{ is a fresh variable} \\ (f(s_1,\dots,s_n), \varphi_1 \wedge \cdots \wedge \varphi_n) & \textit{if } t = f(t_1,\dots,t_n) \textit{ and } \mathsf{tf}(t_i) = (s_i,\varphi_i) \end{cases}$$

Applying the transformation tf *to the left-hand sides of the rules in $\mathcal{R}$ produces*

$$\mathsf{tf}(\mathcal{R}) = \{\ell' \to r\ [\varphi \wedge \psi] \mid \ell \to r\ [\varphi] \in \mathcal{R} \textit{ and } \mathsf{tf}(\ell) = (\ell',\psi)\}$$

Example 14. Applying the transformation tf to the LCTRS $\mathcal{R}$ of Example 13 produces the rules

$$\mathsf{f}(x) \to z\ [z = \mathsf{3}] \qquad \mathsf{g}(\mathsf{f}(x)) \to \mathsf{a} \qquad \mathsf{g}(z) \to \mathsf{a}\ [z = \mathsf{3}]$$

The critical pair $\mathsf{g}(z) \approx \mathsf{a}\ [z = \mathsf{3}]$ can now be joined by an application of the modified third rule. Note that the modified rule does not overlap with the second rule because z may not be instantiated with $\mathsf{f}(x)$. Hence the modified LCTRS $\mathsf{tf}(\mathcal{R})$ is strongly closed and, because it is linear, also confluent.

In the following we show the correctness of the transformation. In particular we prove that the initial rewrite relation is preserved.

Table 1. Specific experimental results.

	result	method	time (in ms)
[12, Example 23]	Timeout	–	10017.70
[12, Example 23] corrected	YES	strongly closed	103.71
Example 6	YES	orthogonal	34.35
[8, Example 3]	YES	weakly orthogonal	50.87
Example 1	YES	strongly closed	115.33
[10, Example 1]	YES	strongly closed	3806.84
Example 11	YES	parallel closed	38.42
Example 12	YES	almost parallel closed	130.36

Lemma 4. *The relations $\rightarrow_{\mathcal{R}}$ and $\rightarrow_{\mathsf{tf}(\mathcal{R})}$ coincide on unconstrained terms.*

Proof. Consider $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. Since the transformation tf does not affect calculation steps, it suffices to consider rule steps. First assume $s = C[\ell\sigma] \rightarrow_{\mathsf{ru}} C[r\sigma] = t$ by applying the rule $\ell \rightarrow r\ [\varphi] \in \mathcal{R}$ and let $\ell' \rightarrow r\ [\varphi'] \in \mathsf{tf}(\mathcal{R})$ be its transformation. So $\mathsf{tf}(\ell) = (\ell', \psi)$ and $\varphi' = \varphi \wedge \psi$. Define the substitution

$$\sigma' = \{\ell'|_p \mapsto \ell|_p \mid (\ell', \psi) = \mathsf{tf}(\ell), p \in \mathcal{P}\mathsf{os}(\ell) \text{ and } \ell|_p \in \mathcal{V}\mathsf{al}\}$$

and let $\tau = \sigma \cup \sigma'$. Since $\mathcal{D}\mathsf{om}(\sigma) \cap \mathcal{D}\mathsf{om}(\sigma') = \varnothing$ by construction, τ is well-defined. From $\sigma \vDash \ell \rightarrow r\ [\varphi]$ and $\sigma' \vDash \psi$ we immediately obtain $\tau \vDash \ell' \rightarrow r\ [\varphi']$, which yields $s = C[\ell'\tau] \rightarrow_{\mathsf{ru}} C[r\tau] = t$ in $\mathsf{tf}(\mathcal{R})$.

For the other direction consider $s = C[\ell'\sigma] \rightarrow_{\mathsf{ru}} C[r'\sigma] = t$ by applying the rule $\ell' \rightarrow r'\ [\varphi'] \in \mathsf{tf}(\mathcal{R})$. The difference between ℓ' and its originating left-hand side ℓ in $\mathcal{R}$ is that value positions in ℓ are occupied by fresh variables in ℓ'. Because σ' respects $\varphi' = \varphi \wedge \psi$, σ' substitutes the required values at these positions in ℓ. As $\sigma \vDash \ell' \rightarrow r'\ [\varphi']$, there exists a rule $\ell \rightarrow r\ [\varphi]$ which is respected by σ and thus $s = C[\ell\sigma] \rightarrow_{\mathsf{ru}} C[r\sigma] = t$ in $\mathcal{R}$. □

As the transformation is used in the implementation and rewriting on constrained terms plays a key role, the following result is needed. The proof is similar to the first half of the proof of Lemma 4 and omitted.

Lemma 5. *The inclusion $\rightarrow_{\mathcal{R}} \subseteq \rightarrow_{\mathsf{tf}(\mathcal{R})}$ holds on constrained terms.*

6 Experimental Results

In order to evaluate our tool we performed some experiments. As there is no official database of interesting confluence problems for LCTRSs, we collected several LCTRSs from the literature and the repository of Ctrl. The problem files in the latter that contain an equivalence problem of two functions for rewriting induction were split into two separate files. The experiments were performed on an AMD Ryzen 7 PRO 4750U CPU with a base clock speed of 1.7 GHz, 8

Table 2. Comparison between confluence criteria implemented in crest.

	O	W	S	P	A
orthogonality (O)	74	74	11	74	74
weak orthogonality (W)		78	13	78	78
strongly closed (S)			20	16	20
parallel closed (P)				83	83
almost parallel closed (A)					89

cores and 32 GB of RAM. The full set of benchmarks consists of 127 problems of which crest can prove 90 confluent, 11 result in MAYBE and 26 in a timeout. With a timeout of 5 s crest needs 141.09 s to analyze the set of benchmarks. We have tested the implementation with 3 well-known SMT solvers: Z3, Yices and CVC5. Among those Z3 gives the best performance regarding time and the handling of non-linear arithmetic. Hence we use Z3 as the default SMT solver in our implementation. In Table 1 we list some interesting systems from this paper and the relevant literature. Full details are available from the website of crest. We choose 5 as the maximum number of steps in the $\to^*$ parts of the strongly closed and almost parallel closed criteria.

From Table 2 the relative power of each implemented confluence criterion on our benchmark can be inferred, i.e., it depicts how many of the 127 problems both methods can prove confluent. This illustrates that the relative applicability in theory (e.g., weakly orthogonal LCTRSs are parallel closed), is preserved in our implementation. We conclude this section with an interesting observation discovered by crest when testing [12, Example 23].

We also tested the applicability of Corollary 1, using the tool Ctrl as a black box for proving termination. Of the 127 problems, Ctrl claims 102 to be terminating and 67 of those can be shown locally confluent by crest, where we limit the number of steps in the joining sequence to 100. It is interesting to note that all of these problems are orthogonal, and so proving termination and finding a joining sequence is not necessary to conclude confluence, on the current set of problems. Of the remaining 35 problems, crest can show confluence of 5 of these by almost parallel closedness.

Example 15. The LCTRS $\mathcal{R}$ is obtained by completing a system consisting of four constrained equations:

1. $\mathsf{f}(x, y) \to \mathsf{f}(z, y) + \mathsf{1}\ [x \geq \mathsf{1} \land z = x - \mathsf{1}]$
2. $\mathsf{f}(x, \mathsf{0}) \to \mathsf{g}(\mathsf{1}, x)\ [x \leq \mathsf{1}]$
3. $\mathsf{g}(\mathsf{0}, y) \to y\ [x \leq \mathsf{0}]$
4. $\mathsf{g}(\mathsf{1}, \mathsf{1}) \to \mathsf{g}(\mathsf{1}, \mathsf{0}) + \mathsf{1}$
5. $\mathsf{h}(x) \to \mathsf{g}(\mathsf{1}, x) + \mathsf{1}\ [x \leq \mathsf{1}]$
6. $\mathsf{h}(x) \to \mathsf{f}(x - \mathsf{1}, \mathsf{0}) + \mathsf{2}\ [x \geq \mathsf{1}]$

Calling crest on $\mathcal{R}$ results in a timeout. As a matter of fact, the LCTRS is not confluent because the critical pair

$$\mathsf{g}(\mathsf{1}, x) + \mathsf{1} \approx \mathsf{f}(x - \mathsf{1}, \mathsf{0}) + \mathsf{2}\ [x \leq \mathsf{1} \land x \geq \mathsf{1}]$$

between rules 5 and 6 is not joinable. Inspecting the steps in [12, Example 23] reveals some incorrect applications of the inference rules of constrained completion, which causes rule 6 to be wrong. Replacing it with the correct rule

$$6'. \quad \mathsf{h}(x) \to (\mathsf{f}(z,\mathsf{0}) + \mathsf{1}) + \mathsf{1}\ [x > \mathsf{1} \wedge z = x - \mathsf{1}]$$

causes crest to report confluence by strong closedness.

7 Concluding Remarks

In this paper we presented new confluence criteria for LCTRSs as well as a new tool in which these criteria have been implemented. We clarified the subtleties that arise when analyzing joinability of critical pairs in LCTRSs and reported experimental results.

For plain rewrite systems many more confluence criteria are known and implemented in powerful tools that compete in the yearly Confluence Competition (CoCo).[5] In the near future we will investigate which of these can be lifted to LCTRSs. We will also advance the creation of a competition category on confluence of LCTRSs in CoCo.

Our tool crest has currently no support for termination. Implementing termination techniques in crest is of clear interest. The starting point here are the methods reported in [6,7,12]. Many LCTRSs coming from applications are actually non-confluent.[6] So developing more powerful techniques for LCTRSs is on our agenda as well.

Acknowledgments. We thank Fabian Mitterwallner for valuable discussions on the presented topics and our Haskell implementation. The detailed comments by the reviewers improved the presentation. Cynthia Kop and Deivid Vale kindly provided us with instructions and a working implementation of Ctrl.

References

1. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press, Cambridge (1998). https://doi.org/10.1017/CBO9781139172752
2. Ciobâcă, Ş, Lucanu, D.: A coinductive approach to proving reachability properties in logically constrained term rewriting systems. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) IJCAR 2018. LNCS (LNAI), vol. 10900, pp. 295–311. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94205-6_20
3. Fuhs, C., Kop, C., Nishida, N.: Verifying procedural programs via constrained rewriting induction. ACM Trans. Comput. Log. **18**(2), 14:1–14:50 (2017). https://doi.org/10.1145/3060143
4. Huet, G.: Confluent reductions: abstract properties and applications to term rewriting systems. J. ACM **27**(4), 797–821 (1980). https://doi.org/10.1145/322217.322230

[5] http://project-coco.uibk.ac.at/.
[6] Naoki Nishida, personal communication (February 2023).

5. Kojima, M., Nishida, N.: From starvation freedom to all-path reachability problems in constrained rewriting. In: Hanus, M., Inclezan, D. (eds.) PADL 2023. LNCS, vol. 13880, pp. 161–179. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-24841-2_11
6. Kop, C.: Termination of LCTRSs. In: Proceedings of the 13th International Workshop on Termination, pp. 59–63 (2013)
7. Kop, C.: Termination of LCTRSs. CoRR abs/1601.03206 (2016). https://doi.org/10.48550/ARXIV.1601.03206
8. Kop, C., Nishida, N.: Term rewriting with logical constraints. In: Fontaine, P., Ringeissen, C., Schmidt, R.A. (eds.) FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 343–358. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40885-4_24
9. Kop, C., Nishida, N.: Constrained term rewriting tool. In: Davis, M., Fehnker, A., McIver, A., Voronkov, A. (eds.) LPAR 2015. LNCS, vol. 9450, pp. 549–557. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48899-7_38
10. Nagele, J., Middeldorp, A.: Certification of classical confluence results for left-linear term rewrite systems. In: Blanchette, J.C., Merz, S. (eds.) ITP 2016. LNCS, vol. 9807, pp. 290–306. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43144-4_18
11. Toyama, Y.: Commutativity of term rewriting systems. In: Fuchi, K., Kott, L. (eds.) Programming of Future Generation Computers II, pp. 393–407. North-Holland (1988)
12. Winkler, S., Middeldorp, A.: Completion for logically constrained rewriting. In: Kirchner, H. (ed.) Proceedings of the 3rd International Conference on Formal Structures for Computation and Deduction. Leibniz International Proceedings in Informatics, vol. 108, pp. 30:1–30:18 (2018). https://doi.org/10.4230/LIPIcs.FSCD.2018.30
13. Winkler, S., Moser, G.: Runtime complexity analysis of logically constrained rewriting. In: LOPSTR 2020. LNCS, vol. 12561, pp. 37–55. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68446-4_2

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.