Journal of Logical and Algebraic Methods in Programming 126 (2022) 100749

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Proving and disproving confluence of context-sensitive rewriting [☆]

Salvador Lucas [a,*], Miguel Vítores [a], Raúl Gutiérrez [b]

[a] DSIC & VRAIN, Universitat Politècnica de València, Spain
[b] Universidad Politécnica de Madrid, Spain

A R T I C L E   I N F O

Article history:
Received 30 July 2021
Received in revised form 4 January 2022
Accepted 4 January 2022
Available online 7 January 2022

Keywords:
Confluence
Context-sensitive rewriting
Program analysis

A B S T R A C T

Context-sensitive rewriting is a restriction of term rewriting where reductions are allowed on specific arguments of function symbols only, and then in particular positions of terms. Confluence is an abstract property of reduction relations guaranteeing that two diverging reduction sequences can always be joined into a common reduct. In this paper we investigate confluence of context-sensitive rewriting and present some novel results. In particular, a characterization of local confluence of context-sensitive rewriting as the joinability of an extended class of *critical pairs* which we introduce here. We also show that the treatment of joinability of critical pairs using theorem proving and solving feasibility problems is useful to automatically prove and disprove confluence of context-sensitive rewriting. Our techniques have been implemented in a new tool, CONFident. We show by means of benchmarks the impact of the new techniques discussed in the paper.

© 2022 Published by Elsevier Inc.

1. Introduction

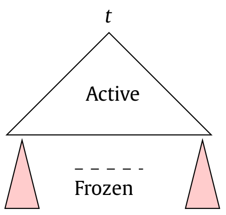
Term rewriting [2] is a computational paradigm which permits the specification of computations by means of rules $\ell \to r$ which can be used to transform expressions s by replacing instances of the left-hand side ℓ in s by the corresponding instance of the right-hand side r. This is usually written $s \to t$ and called a *rewriting step*. Term Rewriting Systems $\mathcal{R}$ are collections of rules $\ell \to r$ which are used to perform such rewriting steps and $\to_{\mathcal{R}}$ is called the one-step rewriting (or reduction) relation induced by $\mathcal{R}$. *Termination* and *confluence* are possibly the most important properties of $\to_{\mathcal{R}}$, see, e.g., [2, Chapters 5 & 6]. We say that $\to_{\mathcal{R}}$ (or just $\mathcal{R}$) is terminating if no infinite sequence $s_1 \to_{\mathcal{R}} s_2 \to_{\mathcal{R}} \cdots$ of rewriting steps can be built. Similarly, $\mathcal{R}$ is confluent if for all terms s, t, and t' such that $s \to^*_{\mathcal{R}} t$ and $s \to^*_{\mathcal{R}} t'$ (where, e.g., $s \to^*_{\mathcal{R}} t$ means that t is obtained from s by means of zero or more rewriting steps), there is a term u such that $t \to^*_{\mathcal{R}} u$ and $t' \to^*_{\mathcal{R}} u$ holds. Intuitively, this means that, even if two diverging sequences of rewriting steps $s \to^*_{\mathcal{R}} t$ and $s \to^*_{\mathcal{R}} t'$ can be issued, there will be a term u to which both t and t' can be rewritten.

Restricting reductions in term rewriting (i.e., forbidding some steps $s \to_{\mathcal{R}} t$) is an obvious way to improve the termination behavior of a rewriting-based system. A simple way to introduce such restrictions is selecting the arguments $\mu(f) \subseteq \{1, \ldots, k\}$ of k-ary function symbols f where rewriting steps can be performed, whereas they are forbidden in arguments of f not in $\mu(f)$. In this way, subterms of a term t can be partitioned into *active* (from the root of the term

☆ 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.
* Corresponding author.

https://doi.org/10.1016/j.jlamp.2022.100749
2352-2208/© 2022 Published by Elsevier Inc.

to some depth in the different branches which depend on the symbols occurring in them) or *frozen* (from that point on). Accordingly, they are able or unable to be rewritten. This is what *Context-Sensitive Rewriting* (*CSR* [20]) does. We write $s \hookrightarrow t$ if s rewrites to t taking into account these restrictions.



For instance, Maude [3] gives support to such restrictions, see [20, Section 4.3].

Confluence and termination of *CSR* were first investigated in [17,18]. The ability of *CSR* to reinforce termination in nonterminating rewrite systems has attracted the attention of a number of researchers, see [20, Section 7], [21, Sections 4, 5, and 6], and the references therein.

Example 1. The following (orthogonal, hence confluent) TRS $\mathcal{R}$ encodes a natural definition of the *factorial* function [20, Example 1.1]:

$$\mathsf{p}(\mathsf{s}(x)) \to x \quad (1) \qquad \mathsf{if}(\mathsf{true}, x, y) \to x \quad (6)$$
$$0 + x \to x \quad (2) \qquad \mathsf{if}(\mathsf{false}, x, y) \to y \quad (7)$$
$$\mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad (3) \qquad \mathsf{zero}(0) \to \mathsf{true} \quad (8)$$
$$0 \times y \to 0 \quad (4) \qquad \mathsf{zero}(\mathsf{s}(x)) \to \mathsf{false} \quad (9)$$
$$\mathsf{s}(x) \times y \to y + (x \times y) \quad (5) \qquad \mathsf{fact}(x) \to \mathsf{if}(\mathsf{zero}(x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(x)) \times x) \quad (10)$$

Rule (10) makes $\mathcal{R}$ nonterminating. Evaluation strategies like the *leftmost-outermost* strategy [26], where only the leftmost outermost redex in each term s is reduced in each step to a term t, written $s \to_{lo} t$, and which is *normalizing*[1] for *left-normal*[2] TRSs (like $\mathcal{R}$), *fail* to achieve a terminating behavior. For instance, the evaluation of $\mathsf{fact}(\mathsf{p}(0))$ never ends[3]:

$$\underline{\mathsf{fact}(\mathsf{p}(0))} \to_{lo} \mathsf{if}(\mathsf{zero}(\mathsf{p}(0)), \mathsf{s}(0), \underline{\mathsf{fact}(\mathsf{p}(\mathsf{p}(0)))} \times \mathsf{p}(0)) \to_{lo} \cdots \quad (11)$$

In contrast, with $\mu(\mathsf{if}) = \{1\}$ the second reduction step is forbidden and we obtain a *terminating behavior* which can be automatically proved.

Of course, imposing such replacement restrictions may bring other problems and we refer the reader to [20] for an account of possible solutions. Termination of *CSR* has received some attention in the past and several tools are able to prove it automatically (e.g., AProVE [7] and MU-TERM [13], see also [20, Section 7.2]).

Regarding (local) confluence of *CSR*, besides guaranteeing the unicity of normal forms computed by *CSR*, other uses have been explored: (i) it can be used as a sufficient condition for proving confluence of unrestricted rewriting [11] and also to guarantee uniqueness of normal forms obtained by unrestricted rewriting [20, Section 8.5]; (ii) it can be used (with termination of *CSR*) to guarantee uniqueness of infinitary normal forms in infinitary rewriting [20, Section 9.4]; (iii) it can be used to (dis)prove confluence of conditional TRSs and *conditional* CSR [21, Sections 5.3 & 8.1.2]; also, (iv) it has been used in the analysis of derivational complexity of normalization using *CSR* [22, Section 7.2.3].

However, except for some new results in [20, Section 8.4], the state-of-the-art of confluence of *CSR* is that of the 1996 paper [17]. For instance, in [20, Example 8.2] it is noticed that *CSR* for $\mathcal{R}$ in Example 1 is *not* confluent (thus witnessing that confluence of rewriting does *not* imply confluence of *CSR*). However, there is no technical result supporting this conclusion, which is obtained by providing a handcrafted counterexample. In order to solve this problem, the following example motivates the theoretical and practical developments in this paper.

Example 2. Consider the orthogonal (i.e., confluent) TRS $\mathcal{R}$:

$$\mathsf{f}(x) \to \mathsf{g}(x, x) \quad (12) \qquad \mathsf{g}(0, x) \to 0 \quad (13) \qquad \mathsf{h}(0) \to 0 \quad (15)$$
$$\mathsf{g}(\mathsf{s}(x), y) \to \mathsf{s}(x) \quad (14) \qquad \mathsf{h}(\mathsf{s}(x)) \to 0 \quad (16)$$

[1] A strategy is said to be *normalizing* if it obtains a normal form t of a term s, whenever it exists.
[2] A TRS is *left-normal* if in every rule $\ell \to r$ the constant and function symbols in the left-hand side ℓ precede (in the linear term notation) the variables.
[3] Along the paper, contracted redexes in reduction sequences are often underlined.

with $\mu(\mathsf{f}) = \mu(\mathsf{g}) = \mu(\mathsf{h}) = \mu(\mathsf{s}) = \{1\}$ [19, Example 19]. The following *peak*

$$s = \mathsf{f}(\mathsf{g}(x,x)) \quad \leftarrow \underrightarrow{\mathsf{f}(\overleftarrow{\mathsf{f}(x)})} \rightarrow \quad \mathsf{g}(\mathsf{f}(x),\mathsf{f}(x)) = t \tag{17}$$

can be produced with *CSR*: the innermost redex ($\overleftarrow{\text{overlined}}$ with a *left-arrow*) can be rewritten due to $\mu(f) = \{1\}$; the outermost redex ($\underrightarrow{\text{underlined}}$ with a *right-arrow*) can also be rewritten as root reduction is always allowed in *CSR*. In both cases, rule (12) applies. Although s and t are *joinable* with respect to unrestricted rewriting:

$$\begin{array}{ll} s = \underline{\mathsf{f}(\mathsf{g}(x,x))} \quad \rightarrow_{(12)} \mathsf{g}(\mathsf{g}(x,x),\mathsf{g}(x,x)) & \text{and} \\ t = \mathsf{g}(\underline{\mathsf{f}(x)},\mathsf{f}(x)) \rightarrow_{(12)} \mathsf{g}(\mathsf{g}(x,x),\underline{\mathsf{f}(x)}) \rightarrow_{(12)} \mathsf{g}(\mathsf{g}(x,x),\mathsf{g}(x,x)) & \end{array}$$

(where the notation $\rightarrow_{(12)}$ makes explicit the use of rule (12) in the rewriting step) the second sequence gets *broken* with *CSR*: the last step, reducing the redex $\mathsf{f}(x)$ in the *second* argument of g is disallowed due to $\mu(\mathsf{g}) = \{1\}$. Actually, $\mathsf{f}(\mathsf{g}(x,x))$ and $\mathsf{g}(\mathsf{f}(x),\mathsf{f}(x))$ are *not* joinable using *CSR* (see Example 50 below).

Unfortunately, there is no systematic treatment which is able to capture a situation like the one depicted in Example 2, thus leading to conclude that *CSR* is *not* confluent as happens for $\mathcal{R}$ and μ as above. In this paper, we fill this gap. Furthermore, we have implemented our results as part of the new tool CONFident [14], which can be found here:

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

As far as we know, this is the first tool which is able to prove confluence of *CSR*.

After some preliminaries in Section 2, Section 3 summarizes the state-of-the-art of techniques for proving confluence of *CSR* and discusses the situation illustrated in Example 2 to identify the source of problems. Section 4 introduces a characterization of local confluence of *CSR* by considering the μ-critical pairs introduced in [17] together with a new kind of *critical pairs* which can be obtained from a TRS $\mathcal{R}$ and a replacement map μ. Overall, we call them *extended μ-critical pairs*. Section 5 provides a logic-based characterization of *CSR* as provability in a first-order theory, recalls the notion of feasibility [12], and introduces the treatment of *CSR* computations using a grounding operator. These notions are used in Section 6 which investigates how to prove and disprove joinability of terms using *CSR* and their use in proofs of joinability of extended μ-critical pairs. Section 7 explains how the tool CONFident proves confluence of *CSR* and reports on our experimental evaluation of the tool. Section 8 discusses how to choose the replacement map μ to make μ-confluence easier to achieve. Section 9 discusses some related work. Section 10 concludes.

2. Preliminaries

Abstract reduction relations 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 $a\,\mathsf{R}\,b$; we say that b is an R-normal form of a (written $a\,\mathsf{R}^!\,b$), if $a\,\mathsf{R}^*b$ and b is an R-normal form. 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*); R is *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 $c\ \mathsf{R}^*d$. The following result, usually called *Newman's lemma*, is well-known (see, e.g., [15, Lemma 2.4], and also [2, Lemma 2.7.2]).[4]

Lemma 3. *A terminating relation is confluent if and only if it is locally confluent.*

Signatures, terms, positions We use the standard notations in term rewriting (see, e.g., [27]). In this paper, $\mathcal{X}$ denotes a countable set of *variables* and $\mathcal{F}$ denotes a *signature*, i.e., a set of *function symbols* $\{f,g,\ldots\}$, 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 symbol labeling the root of t is denoted as $root(t)$. The set of variables occurring in t is $\mathcal{V}ar(t)$. Terms are viewed as labeled trees in the usual way. *Positions* p 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 $\mathcal{P}os(t)$. The set of positions of a subterm s in t is denoted $\mathcal{P}os_s(t)$. The set of positions of non-variable symbols in t are denoted as $\mathcal{P}os_{\mathcal{F}}(t)$, and $\mathcal{P}os_{\mathcal{X}}(t)$ is the set of variable positions of t.

Unification A substitution σ is a mapping $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F},\mathcal{X})$ from variables into terms which is homomorphically extended to a mapping (also denoted σ) $\sigma : \mathcal{T}(\mathcal{F},\mathcal{X}) \to \mathcal{T}(\mathcal{F},\mathcal{X})$. It is standard to assume that substitutions σ satisfy $\sigma(x) = x$ except for a *finite* set of variables, usually called the *domain* of the substitution, and denoted $\mathcal{D}om(\sigma)$. Thus, we often write $\sigma = \{x_1 \mapsto t_1,\ldots,x_n \mapsto t_n\}$ to denote a substitution. Two terms s and t *unify* if there is a substitution σ (i.e., a *unifier*) such that $\sigma(s) = \sigma(t)$. If s and t unify, then there is a *most general unifier* (*mgu*) θ of s and t satisfying that, for any other unifier σ of s and t, there is a substitution τ such that, for all $x \in \mathcal{X}$, $\sigma(x) = \tau(\theta(x))$. Moreover, *mgu*'s are unique up to variable renaming.

[4] The original result by Newman is in [25], although quite a different terminology and notation is used.

Term rewriting A *rewrite rule* is an ordered pair (ℓ, r), written $\ell \to r$, with $\ell, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $\ell \notin \mathcal{X}$ and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)$. The *left-hand side* (*lhs*) of the rule is ℓ and r is the *right-hand side* (*rhs*). A *Term Rewriting System* (TRS) is a pair $\mathcal{R} = (\mathcal{F}, R)$ where R is a set of rewrite rules. We often write $\ell \to r \in \mathcal{R}$ to denote that $\ell \to r$ is a rule of $\mathcal{R}$.

Notation 4. *In the following, when rules $\ell \to r$ are given numeric labels (n), as in Examples 1 (with rules labeled (1)-(10)) and 2 (with rules labeled (12)-(16)), we often write $\ell_{(n)}$ and $r_{(n)}$ to refer the left- and right-hand sides of such rules.*

An instance $\sigma(\ell)$ of the left-hand side ℓ of a rule $\ell \to r$ in a TRS $\mathcal{R}$ is called a *redex*. A TRS $\mathcal{R}$ is *left-linear* if for all rules $\ell \to r \in \mathcal{R}$, ℓ is a linear term.

A term $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ rewrites to t (at position p), written $s \overset{p}{\to}_{\mathcal{R}} t$ (or just $s \overset{p}{\to} t$, $s \to_{\mathcal{R}}$, or even $s \to t$), if $s|_p = \sigma(\ell)$ and $t = s[\sigma(r)]_p$, for some rule $\ell \to r \in R$, $p \in \mathcal{P}os(s)$ and substitution σ. We often say that the redex $s|_p = \sigma(\ell)$ is *contracted* by the rewriting step.

2.1. Context-sensitive rewriting

In this section we provide some background on *CSR*, following [20]. Given a signature $\mathcal{F}$, a *replacement map* is a mapping μ satisfying that, for all symbols f in $\mathcal{F}$, $\mu(f) \subseteq \{1, \dots, ar(f)\}$. The set of replacement maps for the signature $\mathcal{F}$ is $M_{\mathcal{F}}$ (or $M_{\mathcal{R}}$ for a TRS $\mathcal{R} = (\mathcal{F}, R)$). Replacement maps are compared as follows: $\mu \sqsubseteq \mu'$ if for all $f \in \mathcal{F}$, $\mu(f) \subseteq \mu'(f)$; we often say that μ is *more restrictive* than μ'. We also write $\mu \sqsubset \mu'$ iff $\mu \sqsubseteq \mu'$ and $\mu \neq \mu'$ and say that μ is *strictly* more restrictive than μ'. Extreme cases of replacement maps are $\mu_\perp$, which disallows replacements in all arguments of function symbols: $\mu_\perp(f) = \emptyset$ for all $f \in \mathcal{F}$, and $\mu_\top$, which restricts no replacement: $\mu_\top(f) = \{1, \dots, k\}$ for all k-ary symbols $f \in \mathcal{F}$.

The set $\mathcal{P}os^\mu(t)$ of *μ-replacing (or* active*) positions* of t is $\mathcal{P}os^\mu(t) = \{\Lambda\}$, if $t \in \mathcal{X}$, and $\mathcal{P}os^\mu(t) = \{\Lambda\} \cup \{i.p \mid i \in \mu(f), p \in \mathcal{P}os^\mu(t_i)\}$, if $t = f(t_1, \dots, t_k)$. The set of *non-μ-replacing* (or *frozen*) positions of t is $\overline{\mathcal{P}os^\mu}(t) = \mathcal{P}os(t) - \mathcal{P}os^\mu(t)$. Positions of *active* non-variable symbols in t are denoted as $\mathcal{P}os^\mu_{\mathcal{F}}(t)$. Given terms s, t, we let $\overline{\mathcal{P}os^\mu_s}(t) = \overline{\mathcal{P}os^\mu}(t) \cap \mathcal{P}os_s(t)$, i.e., $\overline{\mathcal{P}os^\mu_s}(t)$ denotes the set of frozen positions of subterm s in t.

Given a term t, $\mathcal{V}ar^\mu(t)$ (resp. $\mathcal{V}ar^{\not\mu}(t)$) is the set of variables occurring at active (resp. frozen) positions in t: $\mathcal{V}ar^\mu(t) = \{x \in \mathcal{V}ar(t) \mid \exists p \in \mathcal{P}os^\mu(t), x = t|_p\}$ and $\mathcal{V}ar^{\not\mu}(t) = \{x \in \mathcal{V}ar(t) \mid \exists p \in \overline{\mathcal{P}os^\mu}(t), x = t|_p\}$. In general, $\mathcal{V}ar^\mu(t)$ and $\mathcal{V}ar^{\not\mu}(t)$ are not disjoint.

Example 5. Consider $\mathcal{R}$ and μ in Example 2 and the right-hand side $\mathsf{g}(x, x)$ of rule (12). We have $\mathcal{V}ar^\mu(\mathsf{g}(x, x)) = \mathcal{V}ar^{\not\mu}(\mathsf{g}(x, x)) = \{x\}$.

A pair $(\mathcal{R}, \mu)$ where $\mathcal{R}$ is a TRS and $\mu \in M_{\mathcal{R}}$ is often called a CS-TRS. *Context-sensitive rewriting* (*CSR*) is the restriction of rewriting obtained when a replacement map μ is used to specify the redex positions that can be contracted. A term s μ-rewrites to t, written $s \overset{p}{\hookrightarrow}_{\mathcal{R},\mu} t$ (or $s \hookrightarrow_{\mathcal{R},\mu} t$, $s \hookrightarrow_\mu t$, or even $s \hookrightarrow t$), if $s \overset{p}{\to}_{\mathcal{R}} t$ and p is active in s (i.e., $p \in \mathcal{P}os^\mu(s)$).

Example 6. Again, for $\mathcal{R}$ and μ in Example 2 we have $\mathsf{g}(\underline{\mathsf{f}(x)}, \mathsf{f}(x)) \overset{1}{\hookrightarrow}_{\mathcal{R},\mu} \mathsf{g}(\mathsf{g}(x, x), \mathsf{f}(x))$ because $1 \in \mu(\mathsf{g})$ and hence $1 \in \mathcal{P}os^\mu(\mathsf{g}(\mathsf{f}(x), \mathsf{f}(x)))$. However, $\mathsf{g}(\mathsf{g}(x, x), \underline{\mathsf{f}(x)}) \not\hookrightarrow \mathsf{g}(\mathsf{g}(x, x), \overline{\mathsf{g}(x, x)})$ because $2 \notin \mu(\mathsf{g})$.

If $\hookrightarrow_{\mathcal{R},\mu}$ is (locally) confluent (resp. terminating), we say that $\mathcal{R}$ is (locally) μ-confluent (resp. μ-terminating). The $\hookrightarrow_{\mathcal{R},\mu}$-normal forms are called μ-normal forms (of $\mathcal{R}$). Two terms s and t are μ-joinable (written $s \downarrow_\mu t$) if there is a term u such that $s \hookrightarrow^* u$ and $t \hookrightarrow^* u$.

In the following, for the purpose of analyzing (μ-)confluence, we assume that no TRS contains rules of the form $\ell \to \ell$ introducing no change (or divergence) in terms by using rewriting.

3. Analysis of confluence of context-sensitive rewriting

Given terms s, t, and t', the situation $t \leftarrow s \to t'$ is often called a *peak* [4, Section 4.1]. Two rules $\ell \to r$ and $\ell' \to r'$ sharing no variable (rename if necessary), a nonvariable position $p \in \mathcal{P}os_{\mathcal{F}}(\ell)$ (often called *critical* position), and θ being the *mgu* of $\ell|_p$ and ℓ', define a *critical pair* $\langle s, t \rangle = \langle \theta(\ell)[\theta(r')]_p, \theta(r) \rangle$.[5] The set of critical pairs of a TRS $\mathcal{R}$ is denoted $\mathsf{CP}(\mathcal{R})$. Left-linear TRSs $\mathcal{R}$ without critical pairs are called *orthogonal*.

Each critical pair $\langle \theta(\ell)[\theta(r')]_p, \theta(r) \rangle$ represents peaks of potentially diverging one-step rewritings starting from $\sigma(\ell)$ for some substitution σ refining θ, i.e., $\sigma = \tau \circ \theta$ of θ for some substitution τ:

[5] As remarked in the paragraph below [2, Definition 6.2.1], critical pairs obtained from rules $\ell \to r$ and $\ell' \to r'$ which are renamed versions of the same rule can be safely ignored if the critical position p is the root position, i.e., $p = \Lambda$. In fact, other authors directly *exclude* them from the set of critical pairs of a TRS, see, e.g., [27, Definition 4.2.1].

$$u = \sigma(\ell)[\sigma(r')]_p \leftarrow \sigma(\ell) \rightarrow \sigma(r) = v \tag{18}$$

This is called a *critical peak*. There can be infinitely many critical peaks, corresponding to all possible substitutions σ as above. Since rewriting is closed under substitution application, joinability of all critical pairs implies joinability of all critical peaks, as each critical peak is an instance of some critical pair.

In *CSR*, besides defining the analogous notion of μ-critical pair (where only *active* critical positions $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$ are considered), what is tested is μ-joinability.

Definition 7 *(μ-critical pair).* [20, Definition 8.5] Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. A critical pair $\langle\theta(\ell)[\theta(r')]_p, \theta(r)\rangle \in \mathsf{CP}(\mathcal{R})$ is a μ-critical pair of $\mathcal{R}$ if $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$. The set of μ-critical pairs of $\mathcal{R}$ is $\mathsf{CP}(\mathcal{R},\mu)$. A μ-critical pair $\langle s,t\rangle$ is μ-joinable if $s \downarrow_{\mu} t$ holds.

Example 8. Consider the TRS $\mathcal{R}$ [20, Example 8.10]

$$\mathsf{f}(x) \rightarrow \mathsf{g}(\mathsf{h}(x), x) \tag{19}$$

$$\mathsf{f}(x) \rightarrow x \tag{20}$$

$$\mathsf{g}(x, x) \rightarrow x \tag{21}$$

$$\mathsf{h}(x) \rightarrow x \tag{22}$$

together with $\mu(\mathsf{f}) = \mu(\mathsf{h}) = \emptyset$ and $\mu(\mathsf{g}) = \{1,2\}$. With rule (19) and a renamed version of (20), where x' is used instead of x, $p = \Lambda$ as the critical position, and the *mgu* $\theta = \{x' \mapsto x\}$ as the most-general unifier, we obtain the following μ-critical pair, which is the only one in $\mathsf{CP}(\mathcal{R},\mu)$:

$$\langle s,t\rangle = \langle \mathsf{g}(\mathsf{h}(x), x), x\rangle \tag{23}$$

This μ-critical pair is μ-joinable:

$$s = \mathsf{g}(\underline{\mathsf{h}(x)}, x) \hookrightarrow_{\mathcal{R},\mu} \underline{\mathsf{g}(x,x)} \hookrightarrow_{\mathcal{R},\mu} x = t$$

Some critical pairs $\pi \in \mathsf{CP}(\mathcal{R})$ are not in $\mathsf{CP}(\mathcal{R},\mu)$ due to the replacement restrictions: the critical position in π is frozen. Thus, it is possible that $\mathsf{CP}(\mathcal{R},\mu)$ is empty even though $\mathsf{CP}(\mathcal{R})$ is not.

Example 9. Consider the left-linear TRS $\mathcal{R}$

$$\mathsf{g}(x, \mathsf{a}) \rightarrow \mathsf{c}(x) \tag{24}$$

$$\mathsf{a} \rightarrow \mathsf{b} \tag{25}$$

with $\mu(\mathsf{c}) = \emptyset$ and $\mu(\mathsf{g}) = \{1\}$. Although the left-hand side a of rule (25) overlaps the left-hand side $\mathsf{g}(x,\mathsf{a})$ of rule (24) at position 2, this position is *frozen* in $\mathsf{g}(x,\mathsf{a})$. Hence, $\mathcal{R}$ has no μ-critical pair: $\mathsf{CP}(\mathcal{R},\mu) = \emptyset$.

Unfortunately, in contrast to term rewriting (Huet's Critical Pair Theorem [15, Lemma 3.1]), having no μ-critical pair does *not* mean that $\hookrightarrow_{\mu}$ is (locally) confluent.

Example 10. Consider $\mathcal{R}$ and μ in Example 9. As remarked in Example 9, $\mathsf{CP}(\mathcal{R},\mu)$ is empty. However, the following peak

$$\mathsf{g}(\mathsf{b},\mathsf{a}) \hookleftarrow \mathsf{g}(\mathsf{a},\mathsf{a}) \hookrightarrow \mathsf{c}(\mathsf{a}) \tag{26}$$

is *not* μ-joinable, as $\mathsf{c}(\mathsf{a})$ is a μ-normal form and the only μ-reduction step on $\mathsf{g}(\mathsf{b},\mathsf{a})$ is $\underline{\mathsf{g}(\mathsf{b},\mathsf{a})} \hookrightarrow_{\mu} \mathsf{c}(\mathsf{b})$, leading to a *different* μ-normal form.

The situation depicted in Examples 2 and 10 is captured as follows.

Definition 11 *(LHRV-condition [19,20]).* Let μ be a replacement map. A rule $\ell \rightarrow r$ has *left-homogeneous μ-replacing variables*, written $\mathsf{LHRV}(\ell \rightarrow r, \mu)$, if active variables in the left-hand side ℓ are not frozen neither in ℓ nor in r:

$$\mathcal{V}ar^{\mu}(\ell) \cap (\mathcal{V}ar^{\not\mu}(\ell) \cup \mathcal{V}ar^{\not\mu}(r)) = \emptyset \tag{27}$$

A TRS $\mathcal{R}$ has left-homogeneous μ-replacing variables, written $\mathsf{LHRV}(\mathcal{R},\mu)$, if $\mathsf{LHRV}(\ell \rightarrow r, \mu)$ holds for all rules $\ell \rightarrow r \in \mathcal{R}$.

In the following, when no confusion arises, given a rule $\ell \rightarrow r$ or a TRS $\mathcal{R}$ and a replacement map μ such that $\ell \rightarrow r$ (or $\mathcal{R}$) has left-homogeneous μ-replacing variables, we often say that the rule or TRS *satisfies the LHRV-condition.*

Example 12. For $\mathcal{R}$ and μ in Example 2, $\mathsf{LHRV}((12),\mu)$ (and $\mathsf{LHRV}(\mathcal{R},\mu)$) does *not* hold: since (12) is $\mathsf{f}(x) \rightarrow \mathsf{g}(x,x)$ and $\mu(\mathsf{f}) = \mu(\mathsf{g}) = \{1\}$, we note that variable x is active on the left-hand side but occurs frozen in the right-hand side.

Some particular cases are worth to be collected in the following result whose proof is immediate from Definition 11.

Proposition 13. *Let μ be a replacement map and $\ell \to r$ be a rule.*

1. *LHRV($\ell \to r, \mu_\bot$) and LHRV($\ell \to r, \mu_\top$) always hold.*
2. *LHRV($\ell \to r, \mu$) holds if $\mathcal{V}ar^{\mu}(\ell) = \emptyset$.*
3. *If ℓ is left-linear, then LHRV($\ell \to r, \mu$) holds iff $\mathcal{V}ar^{\mu}(\ell) \cap \mathcal{V}ar^{\not\mu}(r) = \emptyset$.*

Example 14. For the left-linear TRS $\mathcal{R}$ and μ in Example 9, consider rule (24), i.e., $\mathsf{g}(x, \mathsf{a}) \to \mathsf{c}(x)$. Then, LHRV((24), μ) does not hold: since $\mu(\mathsf{g}) = \{1\}$ and $\mu(\mathsf{c}) = \emptyset$, variable x is active on the left-hand side $\mathsf{g}(x, \mathsf{a})$ but frozen in the right-hand side $\mathsf{c}(x)$. With $\mu'(\mathsf{c}) = \mu'(\mathsf{g}) = \{1\}$, though, LHRV($\mathcal{R}, \mu'$) holds: Proposition 13(3) applies to rules (24) and (25) now.

We summarize currently existing results for (dis)proving confluence of *CSR* on the basis of the analysis of the μ-critical pairs in $\mathsf{CP}(\mathcal{R}, \mu)$:

Theorem 15. *[20, Theorems 8.7, 8.9, and 8.12] Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$.*

1. *If there is a non-μ-joinable μ-critical pair $\pi \in \mathsf{CP}(\mathcal{R}, \mu)$, then $\mathcal{R}$ is* not *(locally) μ-confluent.*
2. *If $\mathcal{R}$ is left-linear, LHRV($\mathcal{R}, \mu$) holds and $\mathsf{CP}(\mathcal{R}, \mu)$ is empty, then $\mathcal{R}$ is μ-*confluent.
3. *If $\mathcal{R}$ is μ-terminating, LHRV($\mathcal{R}, \mu$) holds, and all μ-critical pairs $\pi \in \mathsf{CP}(\mathcal{R}, \mu)$ are μ-*joinable*, then $\mathcal{R}$* is μ-confluent.

Example 16 *($\mathcal{R}$ in Example 8 is μ-confluent).* Consider $\mathcal{R}$ and μ as in Example 8. Since the only μ-critical pair in $\mathsf{CP}(\mathcal{R}, \mu)$, i.e., (23), is μ-joinable (as shown in the example) and $\mathcal{R}$ is μ-terminating (use MU-TERM), by Theorem 15.(3), $\mathcal{R}$ is μ-confluent.

Unfortunately, Theorem 15 *cannot* be used to conclude the non-μ-confluence of $\mathcal{R}$ in Examples 1, 2 or 9. In all cases, $\mathcal{R}$ has no μ-critical pair. Hence Theorem 15.(1) (the only result which can be used to conclude non-μ-confluence) does not apply. We investigate how to improve this situation.

4. Proving confluence of context-sensitive rewriting

In the following, rules $\ell \to r$ such that LHRV($\ell \to r, \mu$) holds are called LH$_\mu$-*positive* (and LH$_\mu$-*negative* otherwise). Thus, LHRV($\mathcal{R}, \mu$) holds if all rules in $\mathcal{R}$ are LH$_\mu$-positive. We investigate the role of LH$_\mu$-positive/negative rules in proofs of (non-)μ-confluence.

4.1. Variable peaks

In [5, Section 2.3], as part of their analysis of confluence of Conditional Term Rewriting Systems (CTRSs, see, e.g., [27, Chapter 7]), Dershowitz, Okada, and Sivakumar identify a cause of divergence in conditional reduction, they call a *variable peak*. When restricting the attention to unconditional rules, the notion is as follows. Consider (not necessarily distinct) rules $\ell \to r$ and $\ell' \to r'$ sharing no variable (rename if necessary), a variable x in ℓ at position p (i.e., $\ell|_p = x$), and a substitution σ such that $\sigma(x) = C[\sigma(\ell')]_{p'}$ for some context C with a hole at position p', containing an instance $\sigma(\ell')$ of ℓ' (since the rules share no variable we can use a single substitution σ). Conveniently, we call x the *critical variable* of the variable peak. Then, $\sigma(\ell) = \sigma(\ell)[C[\sigma(\ell')]_{p'}]_p$ can be rewritten in two different ways:

$$s = \sigma(\ell)[C[\sigma(r')]_{p'}]_p \leftarrow \underrightarrow{\sigma(\ell)[C[\overleftarrow{\sigma(\ell')}]_{p'}]_p} = \sigma(\ell) \to \sigma(r) = t \tag{28}$$

where the leftmost step is obtained by contracting in $\sigma(\ell)$ the inner redex $\sigma(\ell')$ of rule $\ell' \to r'$ at position $p.p'$. The rightmost step is obtained by directly applying rule $\ell \to r$ to contract redex $\sigma(\ell)$. Then, (28) is called a *variable peak*.

As noticed in [5], in unconditional rewriting such variable peaks are harmless, as they are *always joinable*, see Fig. 1, where $\sigma'(x) = C[\sigma(r')]_{p'}$ for x the critical variable of the variable peak, and $\sigma'(y) = \sigma(y)$ for all variables $y \neq x$. Note that

1. Reductions in the sequence $\sigma(\ell)[C[\sigma(r')]_{p'}]_p \to^* \sigma'(\ell)$ (in the leftmost part of the diagram) correspond to the possible contraction of redexes $\sigma(\ell')$ occurring at position $q.p'$, where q is a position of variable x in ℓ.
2. This can be done because term rewriting is closed under contexts, see, e.g., [27, Definition 5.2.1], and hence redexes at positions $q.p'$ can be rewritten without any problem.
3. Reductions in the sequence $\sigma(r) \to^* \sigma'(r)$ (in the rightmost part) correspond to the contraction of redexes $\sigma(\ell')$ occurring in terms $C[\sigma(\ell')]_{p'}$ at positions $q'.p'$, where q' is a position of variable x in r.

These are the reasons why in unconditional, unrestricted term rewriting, such variable peaks do not induce specific critical pairs representing such peaks whose joinability must be checked: they are always joinable.

$$\begin{tikzcd}
& \sigma(\ell) \arrow[dl] \arrow[dr] & \\
\sigma(\ell)[C[\sigma(r')]_{p'}]_p \arrow[d, "*"'] & & \sigma(r) \arrow[ddl, "*"] \\
\sigma'(\ell) \arrow[dr] & & \\
& \sigma'(r) &
\end{tikzcd}$$

Fig. 1. Joinability of variable peaks in unconditional rewriting.

4.2. Variable peaks in context-sensitive rewriting

As remarked by Dershowitz, Okada, and Sivakumar [5], when dealing with conditional rewriting the diagram in Fig. 1 may fail to hold due to the need of satisfying the conditional part of rules in the two different branches of the peak.

Remark 17 *(Satisfiability of conditions in conditional term rewriting).* In conditional rewriting, a conditional rule $\ell \to r \Leftarrow c$ applies to a term s to obtain another term t, written $s \to t$, if there is a substitution σ such that $s|_p = \sigma(\ell)$ for some position $p \in \mathcal{P}os(s)$ and the instantiation $\sigma(c)$ of the conditional part c by σ can be *satisfied* (and there are several ways how this can be done, see, e.g., [27, Definition 7.1.3]). Thus, the possibility of the rewriting step $s \to t$, where $t = s[\sigma(r)]_p$, crucially depends on the satisfiability of c by σ.

Dershowitz, Okada, and Sivakumar solve the problem concerning the satisfiability of the conditional part of conditional rules by requiring specific syntactical restrictions on the CTRSs (e.g., the use of normal, terminating, or *noetherian*, CTRSs, see [5, Definition 2] and the paragraph preceding [5, Section 2.1] for the definitions and [5, Section 4] for a discussion about confluence of such systems). No attempt to define and use 'variable critical pairs' is made, though.

Although we do not consider conditional rules, similar problems arise in *CSR*. Remind that we assume two (not necessarily distinct) rules $\ell \to r$ and $\ell' \to r'$ sharing no variables (rename if necessary). We consider a variable x which is active in ℓ, i.e., $x \in \mathcal{V}ar^{\mu}(\ell)$ and a position $p \in \mathcal{P}os^{\mu}_{x}(\ell)$ of an active occurrence of x in ℓ. Taking into account the situation depicted in Fig. 1, in an attempt to replace $\to$ and $\to^*$ by $\hookrightarrow_\mu$ and $\hookrightarrow^*_\mu$ to guarantee the μ-joinability of variable peaks, we may find the following problems:

1. If x occurs active in ℓ at position p, i.e., $\ell|_p = x$ and $p \in \mathcal{P}os^{\mu}(\ell)$, but there also are *frozen* occurrences of x at positions q of ℓ (i.e., $q \in \mathcal{P}os^{\not\mu}_{x}(\ell)$), then $\sigma(\ell)[C[\sigma(r')]_{p'}]_p \hookrightarrow^* \sigma'(\ell)$, contracting redexes $\sigma(\ell')$ in positions $q.p'$ of $\sigma(\ell)$, may become *impossible* because some redexes $\sigma(\ell')$ are frozen in $\sigma(\ell)[C[\sigma(r')]_{p'}]_p$, and the diagram fails to be closed.
2. If variable x is *frozen* at position q' of the right-hand side r of the rule, i.e., $q' \in \overline{\mathcal{P}os^{\mu}_{x}}(r)$, then positions $q'.p'$ of redexes $\sigma(\ell')$ in $\sigma(r)$ are frozen and cannot be μ-rewritten. Hence, the sequence $\sigma(r) \hookrightarrow^* \sigma'(r)$ contracting such redexes could be broken and, again, the diagram would not be closed.

At least one of these two situations (or both) may happen with LH_μ-negative rules $\ell \to r$. In [17], these problems were overcomed by restricting the attention to TRSs $\mathcal{R}$ and $\mu \in M_{\mathcal{R}}$ satisfying the LHRV-condition. Although this suffices to obtain *positive* answers about μ-confluence of TRSs (see Theorem 15.(2) and Theorem 15.(3)), it fails to capture the 'ability' of LH_μ-negative rules to jeopardize μ-confluence. The following definition approaches this goal.

Definition 18 *(LH_μ-variable peak).* Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $\ell \to r$ be an LH_μ-negative rule of $\mathcal{R}$ and $x \in \mathcal{V}ar^{\mu}(\ell)$ be such that $x \in \mathcal{V}ar^{\not\mu}(\ell) \cup \mathcal{V}ar^{\not\mu}(r)$. Let $p \in \mathcal{P}os^{\mu}_{x}(\ell)$, C be a term, $p' \in \mathcal{P}os^{\mu}(C)$, $\ell' \to r' \in \mathcal{R}$ such that $\mathcal{V}ar(\ell) \cap \mathcal{V}ar(\ell') = \emptyset$ (rename if necessary), and σ be a substitution such that $\sigma(x) = C[\sigma(\ell')]_{p'}$. Then,

$$s = \sigma(\ell)[C[\sigma(r')]_{p'}]_p \hookleftarrow \sigma(\ell) \hookrightarrow \sigma(r) = t \tag{29}$$

is called an LH_μ-variable peak.

Example 19. Consider $\mathcal{R}$ and μ in Example 9, and the LH_μ-negative rule (24), i.e., $\mathsf{g}(x,\mathsf{a}) \to \mathsf{c}(x)$. The peak (26) in Example 10, i.e., $\mathsf{g}(\mathsf{b},\mathsf{a}) \hookleftarrow \mathsf{g}(\mathsf{a},\mathsf{a}) \hookrightarrow \mathsf{c}(\mathsf{a})$, is an LH_μ-variable peak which is obtained from (24) using an arbitrary context C, the empty position Λ (which is always active in any term), and the rule (25), i.e., $\mathsf{a} \to \mathsf{b}$. In this case, σ in Definition 18 is given by $\sigma(x) = \mathsf{a}$.

4.3. New critical pairs from variable peaks

An LH_μ-negative rule $\ell \to r \in \mathcal{R}$ contains an active variable $x \in \mathcal{V}ar^\mu(\ell)$ at position $p \in \mathcal{P}os^\mu_x(\ell)$, together with frozen occurrences of x in ℓ or r. Accordingly, for each μ-rewriting steps $u \hookrightarrow_{\mathcal{R}} v$, an LH_μ-variable peak

$$\sigma(\ell)[v]_p \hookleftarrow \sigma(\ell) \hookrightarrow \sigma(r) \tag{30}$$

is obtained, where $\sigma(x) = u$. We introduce the following.

Definition 20 *(LH_μ-critical pair).* Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $\ell \to r \in \mathcal{R}$ be an LH_μ-negative rule and $p \in \mathcal{P}os^\mu_x(\ell)$ for some variable x of ℓ such that $x \in \mathcal{V}ar^{\not\mu}(\ell) \cup \mathcal{V}ar^{\not\mu}(r)$. Let y be a fresh variable, not occurring in ℓ or r. Then,

$$\langle \ell[y]_p, r\rangle \Leftarrow (x \hookrightarrow y) \tag{31}$$

is called an LH_μ-*critical pair*. Borrowing from the usual terminology of (ordinary) critical pairs, position p is called the *critical position* of the LH_μ-critical pair. $\mathsf{LHCP}(\mathcal{R},\mu)$ consists of all LH_μ-critical pairs of $\mathcal{R}$.

Note the following fact, which follows by Definitions 11 and 20.

Proposition 21. *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Then, LHRV$(\mathcal{R},\mu)$ holds if and only if* $\mathsf{LHCP}(\mathcal{R},\mu) = \emptyset$.

In the following, instead of $\langle s,t\rangle \Leftarrow (x \hookrightarrow y)$, we often use $\langle s,t\rangle \Leftarrow x \hookrightarrow y$, if no confusion arises. Also, as for (μ-)critical pairs $\langle s,t\rangle$, we often label LH_μ-critical pairs using labels π by writing $\pi : \langle s,t\rangle \Leftarrow x \hookrightarrow y$.

Remark 22 *(Peaks and conditions in LH_μ-critical pairs).* Given an LH_μ-critical pair $\pi : \langle s,t\rangle \Leftarrow x \hookrightarrow y$, we often call the components $\langle s,t\rangle$ and $x \hookrightarrow y$ the *peak* and *condition* of π, respectively. Note that s may contain several occurrences of variable x together with a *single* occurrence of y, but t contains no occurrence of y.

Example 23 *(Example 1 - LH_μ-critical pair).* For $\mathcal{R}$ and μ as in Example 1, the only LH_μ-negative rule is (10), i.e., $\mathsf{fact}(x) \to \mathsf{if}(\mathsf{zero}(x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(x)))$, where $x \in \mathcal{V}ar^\mu(\ell_{(10)}) \cap \mathcal{V}ar^{\not\mu}(r_{(10)})$. There is a single LH_μ-critical pair:

$$\langle \mathsf{fact}(y), \mathsf{if}(\mathsf{zero}(x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(x)) \times x)\rangle \Leftarrow x \hookrightarrow y \tag{32}$$

Example 24 *(Example 2 - LH_μ-critical pair).* Consider $\mathcal{R}$ and μ as in Example 2. The only LH_μ-negative rule in $\mathcal{R}$, is (12), i.e., $\mathsf{f}(x) \to \mathsf{g}(x,x)$, where $x \in \mathcal{V}ar^\mu(\ell_{(12)}) \cap \mathcal{V}ar^{\not\mu}(r_{(12)})$. There is a single LH_μ-critical pair:

$$\langle \mathsf{f}(y), \mathsf{g}(x,x)\rangle \Leftarrow x \hookrightarrow y \tag{33}$$

Example 25 *(Example 9 - LH_μ-critical pair).* For $\mathcal{R}$ and μ in Example 9, the only LH_μ-negative rule is (24), i.e., $\mathsf{g}(x,\mathsf{a}) \to \mathsf{c}(x)$, where $x \in \mathcal{V}ar^\mu(\ell_{(24)}) \cap \mathcal{V}ar^{\not\mu}(r_{(24)})$. We obtain a single LH_μ-critical pair:

$$\langle \mathsf{g}(y,\mathsf{a}), \mathsf{c}(x)\rangle \Leftarrow x \hookrightarrow y \tag{34}$$

The notation $\langle \ell[y]_p, r\rangle \Leftarrow x \hookrightarrow y$ suggests that LH_μ-critical pairs are somehow related to *conditional critical pairs* of CTRSs (see, e.g., [27, Definition 7.1.8.(1)]). Indeed, the *peak* component $\langle \ell[y]_p, r\rangle$ of the LH_μ-critical pair, together with a substitution σ, is expected to represent a variable peak as in (30) provided that the instance $\sigma(x) \hookrightarrow \sigma(y)$ of the *condition* $x \hookrightarrow y$ holds, i.e., $\sigma(x)$ μ-rewrites into $\sigma(y)$ in one step: $u = \sigma(x) \hookrightarrow_{\mathcal{R},\mu} \sigma(y) = v$. However, the conditional part $x \hookrightarrow y$ is always satisfiable by means of a substitution σ such that $\sigma(x) = \ell'$ and $\sigma(y) = r'$ for some (not necessarily distinct) rule $\ell' \to r' \in \mathcal{R}$. This is in sharp contrast with conditional critical pairs, that can be *infeasible*, see [27, Definition 7.1.8.(3)].

We define μ-joinability of LH_μ-critical pairs as follows.

Definition 26 *(μ-joinable LH_μ-critical pair).* Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $\pi : \langle s,t\rangle \Leftarrow x \hookrightarrow y$ be an LH_μ-critical pair. We say that π is μ-joinable if, for all substitutions σ such that $\mathcal{D}om(\sigma) = \{x,y\}$, $\sigma(x) \hookrightarrow_{\mathcal{R},\mu} \sigma(y)$ implies $\sigma(s) \downarrow_\mu \sigma(t)$.

Example 27 *(Example 2 - Non-μ-joinable LH_μ-critical pair).* For $\mathcal{R}$ and μ in Example 2, consider the LH_μ-critical pair (33), i.e., $\pi : \langle \mathsf{f}(y), \mathsf{g}(x,x)\rangle \Leftarrow x \hookrightarrow y$ is not μ-joinable: since $\underline{\mathsf{f}(x)} \hookrightarrow \mathsf{g}(x,x)$, with $\sigma = \{x \mapsto \mathsf{f}(x), y \mapsto \mathsf{g}(x,x)\}$, we obtain $\sigma(\mathsf{f}(y)) = \mathsf{f}(\mathsf{g}(x,x))$ and $\sigma(\mathsf{g}(x,x)) = \mathsf{g}(\mathsf{f}(x),\mathsf{f}(x))$ which have been claimed non-μ-joinable in Example 2. Thus, π is not μ-joinable (see also Example 58).

Example 28 *(Example 9 - Non-μ-joinable* LH_μ*-critical pair).* For $\mathcal{R}$ and μ in Example 9, consider the LH_μ-critical pair (34), i.e., $\pi : \langle \mathsf{g}(y,\mathsf{a}),\mathsf{c}(x)\rangle \Leftarrow x \hookrightarrow y$, is *not* μ-joinable: we have $\mathsf{a} \hookrightarrow \mathsf{b}$, but with $\sigma = \{x \mapsto \mathsf{a}, y \mapsto \mathsf{b}\}$, $\sigma(\mathsf{c}(x)) = \mathsf{c}(\mathsf{a})$ is a μ-normal form, and the only μ-rewriting step on $\sigma(\mathsf{g}(y,\mathsf{a})) = \mathsf{g}(\mathsf{b},\mathsf{a})$ is $\underline{\mathsf{g}(\mathsf{b},\mathsf{a})} \hookrightarrow_\mu \mathsf{c}(\mathsf{b})$, leading to a different μ-normal form. Thus, π is not μ-joinable (see also Example 59).

Now, we collect the 'old' μ-critical pairs and the new LH_μ-critical pairs in a single set which, as we prove below, can be used to characterize local μ-confluence.

Definition 29 *(Extended μ-critical pairs).* Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. The set

$$\mathsf{ECP}(\mathcal{R},\mu) = \mathsf{CP}(\mathcal{R},\mu) \cup \mathsf{LHCP}(\mathcal{R},\mu) \tag{35}$$

which *extends* $\mathsf{CP}(\mathcal{R},\mu)$ by adding the LH_μ-critical pairs is called the set of *extended μ-critical pairs* of $\mathcal{R}$.

4.4. Characterization of local confluence of CSR

The following result provides the first general characterization of local confluence of *CSR* on the basis of the analysis of extended μ-critical pairs.

Theorem 30 *(Local μ-confluence). Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Then, $\mathcal{R}$ is locally μ-confluent if and only if all pairs in* $\mathsf{ECP}(\mathcal{R},\mu)$ *are μ-joinable.*

Proof. For the *only if* part, we consider two kinds of μ-critical pairs in $\mathsf{ECP}(\mathcal{R},\mu)$. Let $\pi : \langle s,t\rangle = \langle \theta(\ell)[\theta(r')]_p, \theta(r)\rangle \in \mathsf{CP}(\mathcal{R},\mu)$ according to Definition 7. Since $s \hookleftarrow \theta(\ell) \hookrightarrow t$, by local μ-confluence, $s \downarrow_\mu t$, i.e., π is μ-joinable. Similarly, for LH_μ-variable pairs $\pi : \langle s,t\rangle \Leftarrow x \hookrightarrow_\mu y$, and substitutions σ such that $\sigma(x) \hookrightarrow \sigma(y)$, there is a peak (30). By local μ-confluence, $\sigma(s) \downarrow_\mu \sigma(t)$, i.e., π is μ-joinable.

For the *if* part, let s,t,t' be terms such that $s \overset{p}{\hookrightarrow} t$ and $s \overset{p'}{\hookrightarrow} t'$, that is, there are rules $\alpha : \ell \to r$, $\alpha' : \ell' \to r' \in \mathcal{R}$, active positions $p, p' \in \mathcal{P}os^\mu(s)$, and substitutions σ, σ' such that $s|_p = \sigma(\ell)$, $s|_{p'} = \sigma'(\ell')$, $t = s[\sigma(r)]_p$, and $t' = s[\sigma'(r')]_{p'}$. Let us consider two cases according to the positions p and p' of the two redexes.

1. The positions p and p' are disjoint. Then, we have $t|_{p'} = \sigma'(\ell')$, $p' \in \mathcal{P}os^\mu(t)$, $t'|_p = \sigma(\ell)$, and $p \in \mathcal{P}os^\mu(t')$. For $u = t[\sigma'(r')]_{p'} = t'[\sigma(r)]_p$, we have $t \hookrightarrow u$ and $t' \hookrightarrow u$.
2. The positions p and p' are *not* disjoint. Without loss of generality, we can assume $p \leq p'$ (the case $p' \leq p$ is analogous). Let $p' = p.q$ for some position q. We have

$$s = s[\sigma(\ell)[\sigma'(\ell')]_q]_p \overset{p'}{\hookrightarrow} s[\sigma(\ell)[\sigma'(r')]_q]_p = t' \tag{36}$$

$$s = s[\sigma(\ell)]_p \overset{p}{\hookrightarrow} s[\sigma(r)]_p = t \tag{37}$$

   We show that there is u such that $\sigma(r) \hookrightarrow^* u$ and $\sigma(\ell)[\sigma'(r')]_q \hookrightarrow^* u$. Note also that, since $q \in \mathcal{P}os^\mu(\sigma(\ell))$, we also have the following:

$$\sigma(\ell)[\sigma'(\ell')]_q \overset{q}{\hookrightarrow} \sigma(\ell)[\sigma'(r')]_q \tag{38}$$

$$\sigma(\ell) \overset{\Lambda}{\hookrightarrow} \sigma(r) \tag{39}$$

   If we prove μ-joinability of $\sigma(\ell)[\sigma'(r')]_q$ and $\sigma(r)$, then the conclusion $t \downarrow_\mu t'$ follows by closedness under active contexts [20, Section 4] applied to $s[\,]_p$. Now, we consider two cases. First, the case when *ℓ and ℓ' overlap* corresponds to the use of μ-critical pairs and is proved in [19, Theorem 4.13]. On the other hand, if *ℓ and ℓ' do not overlap*, then we can write $q = q_1.q_2$ for positions q_1 and q_2 such that $\ell|_{q_1} = x \in \mathcal{X}$ and $\sigma(x)|_{q_2} = \sigma'(\ell')$. Note that, since $\sigma'(\ell')$ is active in $\sigma(\ell)$, $q_1 \in \mathcal{P}os^\mu_x(\ell)$, i.e., x is active in ℓ, i.e., $x \in \mathcal{V}ar^\mu(\ell)$. Hence, $\sigma(x) = \sigma(x)[\sigma'(\ell')]_{q_2} \hookrightarrow \sigma(x)[\sigma'(r')]_{q_2}$. We consider two cases:
   (a) If x occurs active everywhere in ℓ and r, then $\sigma(\ell)[\sigma'(r')]_q \hookrightarrow^* \sigma''(\ell) \hookrightarrow \sigma''(r)$ for σ'' defined as $\sigma''(x) = \sigma'(r')$ and $\sigma''(y) = \sigma(y)$ if $y \neq x$. We also have $\sigma(r) \hookrightarrow^* \sigma''(r)$ because all occurrences of x in r are active.
   (b) If x, which is active in ℓ, also occurs in frozen positions in $\alpha : \ell \to r$, then α is LH_μ-negative and there is an LH_μ-critical pair $\pi : \langle \ell[y]_{q_1}, r\rangle \Leftarrow x \hookrightarrow y \in \mathsf{ECP}(\mathcal{R},\mu)$ where y is a fresh variable, not occurring in α. Since $\sigma(x) \hookrightarrow \sigma(x)[\sigma'(r')]_{q_2}$, we let the substitution τ defined on $\mathcal{V}ar(\ell) \cup \{y\}$ as follows: $\tau(y) = \sigma(x)[\sigma'(r')]_{q_2}$, and $\tau(z) = \sigma(z)$ if $z \neq y$. Note that $\tau(\ell[y]_{q_1}) = \sigma(\ell)[\sigma(x)[\sigma'(r')]_{q_2}]_{q_1} = \sigma(\ell)[\sigma'(r')]_q$ and $\tau(r) = \sigma(r)$. By μ-joinability of π, $\tau(\ell[y]_{q_1})$ and $\tau(r)$ are μ-joinable. Therefore, $\sigma(\ell)[\sigma'(r')]_q$ and $\sigma(r)$ are μ-joinable. Finally, we conclude μ-joinability of t and t'. $\square$

As a corollary of Lemma 3, we have the following.

Lemma 31. *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. If $\mathcal{R}$ is μ-terminating, then $\mathcal{R}$ is μ-confluent if and only if $\mathcal{R}$ is locally μ-confluent.*

4.5. Confluence and orthogonality in context-sensitive rewriting

Now, we obtain the following generalization of Theorem 15 where the LHRV-condition is not used anymore.

Theorem 32. *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$.*

1. *If there is a non-μ-joinable pair $\pi \in \mathsf{ECP}(\mathcal{R}, \mu)$, then $\mathcal{R}$ is not (locally) μ-confluent.*
2. *If $\mathcal{R}$ is left-linear and $\mathsf{ECP}(\mathcal{R}, \mu)$ is empty, then $\mathcal{R}$ is μ-confluent.*
3. *If $\mathcal{R}$ is μ-terminating and all pairs in $\mathsf{ECP}(\mathcal{R}, \mu)$ are μ-joinable, then $\mathcal{R}$ is μ-confluent.*

Proof. 1. Since μ-confluence implies local μ-confluence, it is immediate from Theorem 30.
2. It follows from Theorem 15.(2) and Proposition 21.
3. It follows from Theorem 30 and Lemma 31. □

Example 33 *($\mathcal{R}$ in Example 9 is not μ-confluent).* According to Example 28, by using Theorem 32.(1) we can conclude now that $\mathcal{R}$ in Example 9 is not locally μ-confluent, nor μ-confluent.

In [19, Definition 11] (and also [20, Definition 8.11]) a left-linear TRS without μ-critical pairs (i.e., such that $\mathsf{CP}(\mathcal{R}, \mu)$ is empty) was called *μ-orthogonal*. In sharp contrast with the unrestricted case, though, μ-orthogonality does *not* immediately imply μ-confluence, as the LHRV-condition is required (see [19, Theorem 6]).

Example 34. According to [19, Definition 11], $\mathcal{R}$ in Example 8 is μ-orthogonal. However, it is *not* μ-confluent (see Example 33).

In view of Theorem 32.(2), the following redefinition of the notion makes sense.

Definition 35 *(μ-orthogonal TRS).* Let $\mathcal{R}$ be a left-linear TRS and $\mu \in M_{\mathcal{R}}$. If $\mathsf{ECP}(\mathcal{R}, \mu)$ is empty, then $\mathcal{R}$ is called μ-orthogonal.

With the new Definition 35, $\mathcal{R}$ in Example 9 is no longer μ-orthogonal. However, for the new definition of μ-orthogonality, Theorem 15.(2) yields the desired correspondence with unrestricted rewriting.

Corollary 36. *μ-orthogonal TRSs are μ-confluent.*

Example 37 *($\mathcal{R}$ in Example 9 with μ' in Example 14 is μ'-confluent).* Consider the left-linear TRS $\mathcal{R}$ in Example 9 and μ' as in Example 14. As shown in Example 14, $\mathsf{LHRV}(\mathcal{R}, \mu')$-holds. Since $\mathcal{R}$ has no μ'-critical pair, by Corollary 36, $\mathcal{R}$ is μ'-confluent.

As remarked in the paragraph below Example 25, the structure of LH_{μ}-critical pairs, and, consequently, the definition of μ-joinability of LH_{μ}-critical pairs (Definition 26) has a *conditional* component. In Section 6 we show how to deal with the corresponding μ-joinability problems (for extended μ-critical pairs) by using logic-based methods. The following section prepares the way.

5. Logic-based analysis of context-sensitive computations

In this section we first provide a logic-based description of *CSR* as deduction with respect to a first-order theory (Section 5.1). Then we use such a description to apply the notion of *feasibility problem* to *CSR* (Section 5.2). Feasibility problems are sequences of atoms where all variables are assumed to be existentially quantified. Such problems are solved positively if appropriate substitutions are found to *satisfy* them. Since atoms in feasibility problems may contain variables which are more naturally considered to be universally quantified, in Section 5.3 we discuss the possibility of *replacing* such variables by new constant symbols so that universal quantification is no longer necessary and sequences of atoms are properly understood as feasibility problems. As remarked above, we make a practical use of all this in Section 6.

5.1. The first-order theory of context-sensitive rewriting

As discussed in [20, Section 4.1], given a TRS $\mathcal{R}$ and $\mu \in M_{\mathcal{R}}$, both $\hookrightarrow_{\mathcal{R},\mu}$ and $\hookrightarrow^*_{\mathcal{R},\mu}$ can be defined as provability of goals $s \to t$ and $s \to^* t$ (where $\to$ and $\to^*$ are viewed as predicate symbols) in a first-order theory $\overline{\mathcal{R}^\mu}$ associated to $\mathcal{R}$ and μ which consists of four components:

1. A sentence
$$(\forall x)\; x \to^* x \tag{40}$$
expressing *reflexivity* of the many-step context-sensitive rewrite relation $\hookrightarrow^*_{\mathcal{R},\mu}$.
2. A sentence
$$(\forall x, y, z)\; x \to y \wedge y \to^* z \Rightarrow x \to^* z \tag{41}$$
expressing *compatibility* of one-step and many-step relations $\hookrightarrow_{\mathcal{R},\mu}$ and $\hookrightarrow^*_{\mathcal{R},\mu}$.
3. For each k-ary function symbol $f \in \mathcal{F}$ and $i \in \mu(f)$, a sentence
$$(\forall \vec{x})(\forall y_i)\; x_i \to y_i \Rightarrow f(x_1, \ldots, x_i, \ldots, x_k) \to f(x_1, \ldots, y_i, \ldots, x_k) \tag{42}$$
where $\vec{x}$ is a sequence $x_1, \ldots, x_k$ of k distinct variables, enabling the *propagation* of μ-reduction steps $s_i \hookrightarrow_{\mathcal{R},\mu} t_i$ on active arguments s_i of f for $i \in \mu(f)$ in terms $f(s_1, \ldots, s_i, \ldots, s_k)$ so that $f(s_1, \ldots, s_i, \ldots, s_k) \hookrightarrow_{\mathcal{R},\mu} f(s_1, \ldots, t_i, \ldots, s_k)$ holds for all terms $s_1, \ldots, s_k$ and t_i.[6]
4. For each rule $\ell \to r$ in $\mathcal{R}$ a sentence
$$(\forall \vec{x})\; \ell \to r \tag{43}$$
where $\vec{x}$ is the sequence $x_1, \ldots, x_n$ of all variables occurring in the rule, representing the possibility of applying a rule $\ell \to r$ to any instantiation $\sigma(\ell)$ of ℓ by a substitution σ to obtain a μ-rewriting step $\sigma(\ell) \hookrightarrow_{\mathcal{R},\mu} \sigma(r)$.

Example 38. Consider the (terminating) orthogonal TRS $\mathcal{R}$
$$\mathsf{f}(x) \to \mathsf{g}(x) \tag{44}$$
$$\mathsf{g}(x) \to x \tag{45}$$
together with $\mu(\mathsf{f}) = \{1\}$ and $\mu(\mathsf{g}) = \emptyset$. The theory $\overline{\mathcal{R}^\mu}$ is
$$(\forall x)\;\; x \to^* x \tag{46}$$
$$(\forall x, y, z)\;\; x \to y \wedge y \to^* z \Rightarrow x \to^* z \tag{47}$$
$$(\forall x, y)\;\; x \to y \Rightarrow \mathsf{f}(x) \to \mathsf{f}(y) \tag{48}$$
$$(\forall x)\;\; \mathsf{f}(x) \to \mathsf{g}(x) \tag{49}$$
$$(\forall x)\;\; \mathsf{g}(x) \to x \tag{50}$$

Now, for all terms s and t we have
$$s \hookrightarrow_{\mathcal{R},\mu} t \text{ iff } \overline{\mathcal{R}^\mu} \vdash s \to t \quad \text{and} \quad s \hookrightarrow^*_{\mathcal{R},\mu} t \text{ iff } \overline{\mathcal{R}^\mu} \vdash s \to^* t$$
where '$\vdash$' denotes provability with any (sound and complete) first-order calculus.[7] Note that for $\vec{x} = x_1, \ldots, x_n$ the variables occurring in s and t, we equivalently have
$$s \hookrightarrow_{\mathcal{R},\mu} t \text{ iff } \overline{\mathcal{R}^\mu} \vdash (\forall \vec{x})\, s \to t \quad \text{and} \quad s \hookrightarrow^*_{\mathcal{R},\mu} t \text{ iff } \overline{\mathcal{R}^\mu} \vdash (\forall \vec{x})\, s \to^* t$$

[6] This is often called μ-monotonicity, see [20, Section 7.1].
[7] Resolution, Hilbert's axiomatic calculus, natural deduction, etc., see [6], for instance.

5.2. Feasibility sequences in context-sensitive rewriting

Following [12], given a set $\mathbb{P}$ of (binary) predicates $\bowtie\in\mathbb{P}$, we consider a $\mathbb{P}$-indexed set $\mathbb{T}=\{\mathsf{Th}_{\bowtie}\mid\bowtie\in\mathbb{P}\}$ of first-order theories $\mathsf{Th}_{\bowtie}$. 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 39 *(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 this paper, we consider *feasibility problems* given as sequences[8]

$$\gamma=s_1\bowtie_1 t_1,\ldots,s_n\bowtie_n t_n \tag{51}$$

of atoms $s_i\bowtie_i t_i$ where for all $1\leq i\leq n$, $\bowtie_i$ is a binary predicate symbol $\hookrightarrow$ or $\hookrightarrow^*$ representing one-step or many-step μ-reduction conditions.

Remark 40. According to Definition 39 and considering the definition of $\overline{\mathcal{R}^\mu}$ given in Section 5.1, the predicate symbols $\to$ and $\to^*$ (rather than $\hookrightarrow$ and $\hookrightarrow^*$) should be used in feasibility conditions (51), as $\overline{\mathcal{R}^\mu}$ uses predicates $\to$ and $\to^*$. However, we think that using atoms $s\hookrightarrow t$ and $s\hookrightarrow^* t$ in feasibility conditions hopefully improves readability of the paper.

Given a TRS $\mathcal{R}$ and $\mu\in M_{\mathcal{R}}$, we consider the theory $\overline{\mathcal{R}^\mu}$ to deal with both $\hookrightarrow$ and $\hookrightarrow^*$ predicates. Then, as a particularization of Definition 39, we use the following:

Definition 41. Let $\mathcal{R}$ be a TRS and $\mu\in M_{\mathcal{R}}$. A feasibility sequence γ of the form (51) is $\overline{\mathcal{R}^\mu}$-feasible if and only if there is a substitution σ such that, for all conditions $s\bowtie t$ in γ, $\overline{\mathcal{R}^\mu}\vdash\sigma(s)\bowtie\sigma(t)$ (i.e., $\sigma(s)\hookrightarrow_{\mathcal{R},\mu}\sigma(t)$ if $\bowtie$ is $\hookrightarrow$, or $\sigma(s)\hookrightarrow^*_{\mathcal{R},\mu}\sigma(t)$ if $\bowtie$ is $\hookrightarrow^*$) holds. Otherwise, γ is called $\overline{\mathcal{R}^\mu}$*-infeasible*.

The tool infChecker [12, Section 5] is able to (dis)prove such feasibility problems. In order to take benefit from feasibility sequences $s_1\bowtie_1 t_1,\ldots,s_n\bowtie_n t_n$, it is useful to first *remove* all variables in terms s_i and t_i for $1\leq i\leq n$ which are (implicitly) universally quantified and thus are not expected to be instantiated. The following section explores this approach which is used in Section 6 below.

5.3. Dealing with variables as fresh constants

In term rewriting, variables occurring in terms t_i in reduction sequences $t_1\to t_2\to\cdots$ are treated *as constants*; 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 formalize and exploit this fact to improve proofs of (non-)μ-joinability of terms.

Let $\mathcal{F}$ be a signature and $\mathcal{X}$ be a set of variables. For each variable $x\in\mathcal{X}$, we consider a new *constant* symbol c_x such that $c_x\notin\mathcal{F}\cup\mathcal{X}$. Let $C_{\mathcal{X}}=\{c_x\mid x\in\mathcal{X}\}$ and $\mathcal{F}_{\mathcal{X}}=\mathcal{F}\cup C_{\mathcal{X}}$. Note that there is a bijection between the set $\mathcal{T}(\mathcal{F},\mathcal{X})$ of terms with variables for the signature $\mathcal{F}$ and $\mathcal{T}(\mathcal{F}_{\mathcal{X}})$, the set 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. We call $t^{\downarrow}$ the *grounded* version of t and $\_^{\downarrow}$ is called the *grounding* operator.[9] Vice versa: given $t\in\mathcal{T}(\mathcal{F}_{\mathcal{X}})$, the *ungrounded* version $t^{\uparrow}\in\mathcal{T}(\mathcal{F},\mathcal{X})$ (obtained by the *ungrounding* operator $\_^{\uparrow}$) is obtained by replacing, for all $x\in\mathcal{X}$, each constant c_x in t by x.

Proposition 42. *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 43. *Let $\mathcal{R}=(\mathcal{F},R)$ be a TRS, $\mu\in M_{\mathcal{R}}$, and $s,t\in\mathcal{T}(\mathcal{F},\mathcal{X})$. Then, $s\hookrightarrow_{\mathcal{R},\mu}t$ if and only if $s^{\downarrow}\hookrightarrow_{\mathcal{R},\mu}t^{\downarrow}$.*

[8] Actually, we can think of f-sequences as *sets* of f-conditions, as the order of the f-conditions, or their repetition, does not change their (in)feasibility.

[9] We use $\downarrow$ as a superscript to denote the *grounding* operation as in $t^{\downarrow}$, hopefully not leading to confusion with the infix use of $\downarrow$ as joinability operator, as in $s\downarrow t$.

Proof. By induction on the structure of s. As for the *only if* part, if s is a variable x, then x is a μ-normal form, and $s^{\downarrow} = c_x$ also is, as $c_x \notin \mathcal{F}$. If $s = f(s_1, \ldots, s_k)$ for some $k \geq 0$ (i.e., we include here the case when s is a constant), then we consider two cases according to $p \in \mathcal{P}os^{\mu}(s)$: (i) if $p = \Lambda$, then $s = \sigma(\ell)$ and $t = \sigma(r)$ for some $\ell \to r \in \mathcal{R}$ and substitution σ. Then, $s^{\downarrow} = \sigma^{\downarrow}(\ell) \overset{\Lambda}{\hookrightarrow} \sigma^{\downarrow}(r) = t^{\downarrow}$. (ii) if $p = i.p'$ for some $i \in \mu(f)$ and $p' \in \mathcal{P}os^{\mu}(s_i)$, then, $s_i \overset{p'}{\hookrightarrow} t_i$ and, by the induction hypothesis, $s_i^{\downarrow} \overset{p'}{\hookrightarrow} t_i^{\downarrow}$. Thus, by definition of $\_^{\downarrow}$, and since the marking process does not affect the active/frozen status of positions, we obtain $s^{\downarrow} = f(s_1^{\downarrow}, \ldots, s_{i-1}^{\downarrow}, s_i^{\downarrow}, \ldots, s_k^{\downarrow}) \overset{p}{\hookrightarrow} f(s_1^{\downarrow}, \ldots, s_{i-1}^{\downarrow}, t_i^{\downarrow}, \ldots, s_k^{\downarrow}) = t^{\downarrow}$. The *if* part is similar starting with $s^{\downarrow}$ and $t^{\downarrow}$ which are ground terms in $\mathcal{T}(\mathcal{F}_{\mathcal{X}})$ and then proceeding as above using $\_^{\uparrow}$ instead of $\_^{\downarrow}$, using Proposition 42. □

It is also useful to perform a *partial grounding* of terms t where all variables occurring in t are grounded, *except* those collected in a finite set $V \subseteq \mathcal{X}$. Thus, in the following, given a term t and $V \subseteq \mathcal{X}$, the term $t^{\downarrow_{\overline{V}}}$ is as t except that all occurrences of variables $x \in \mathcal{V}ar(t) - V$ are replaced by c_x. Note that $\mathcal{V}ar(t^{\downarrow_{\overline{V}}}) \subseteq V$. Furthermore, $t^{\downarrow} = t^{\downarrow_{\overline{\emptyset}}}$. We have the following result, whose proof is analogous to Proposition 43.

Proposition 44. *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{R}}$, $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $V \subseteq \mathcal{X}$. Then, $s \hookrightarrow_{\mathcal{R},\mu} t$ if and only if $s^{\downarrow_{\overline{V}}} \hookrightarrow_{\mathcal{R},\mu} t^{\downarrow_{\overline{V}}}$.*

6. Proving and disproving μ-joinability of extended μ-critical pairs

As a corollary of Proposition 43, we have the following:

Corollary 45. *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{R}}$, and $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Then, s and t are μ-joinable if and only if $s^{\downarrow}$ and $t^{\downarrow}$ are μ-joinable.*

This result is useful to deal with μ-joinability of arbitrary terms, being part of critical pairs or not. Joinability of terms is easily transformed into a feasibility problem.

Proposition 46. *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$, and $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $z \in \mathcal{X}$. Then, s and t are μ-joinable if and only if $s^{\downarrow} \hookrightarrow^* z, t^{\downarrow} \hookrightarrow^* z$ is $\overline{\mathcal{R}^{\mu}}$-feasible.*

Proof. By Corollary 45, s and t are μ-joinable if and only if $s^{\downarrow}$ and $t^{\downarrow}$ are. If $s^{\downarrow}$ and $t^{\downarrow}$ are μ-joinable, then there is a term u such that $s^{\downarrow} \hookrightarrow^*_{\mathcal{R},\mu} u$ and $t^{\downarrow} \hookrightarrow^*_{\mathcal{R},\mu} u$. Then, $s^{\downarrow} \hookrightarrow^* z, t^{\downarrow} \hookrightarrow^* z$ is $\overline{\mathcal{R}^{\mu}}$-feasible by using $\sigma(z) = u$. As for the if part, if $s^{\downarrow} \hookrightarrow^* z, t^{\downarrow} \hookrightarrow^* z$ is $\overline{\mathcal{R}^{\mu}}$-feasible, then there is a substitution σ such that $\sigma(s^{\downarrow}) = s^{\downarrow} \hookrightarrow^*_{\mathcal{R},\mu} \sigma(z)$ and $\sigma(t^{\downarrow}) = t^{\downarrow} \hookrightarrow^*_{\mathcal{R},\mu} \sigma(z)$, i.e., $s^{\downarrow}$ and $t^{\downarrow}$ are μ-joinable, and hence s and t also are. □

In the following, we develop the use of these results to deal with extended μ-critical pairs.

6.1. μ-critical pairs

Corollary 45 immediately applies to (dis)prove μ-joinability of μ-critical pairs.

Corollary 47. *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{R}}$, and $\pi : \langle s, t\rangle \in \mathsf{CP}(\mathcal{R}, \mu)$. Then, π is μ-joinable if and only if $s^{\downarrow}$ and $t^{\downarrow}$ are μ-joinable.*

Remark 48 *(Joinability of terms and critical pairs).* Proposition 43 and Corollary 45 can also be used to prove joinability of terms and critical pairs in unrestricted rewriting: term rewriting can be seen as a particular case of *CSR* where $\mu = \mu_{\top}$, i.e., $\to_{\mathcal{R}} = \hookrightarrow_{\mathcal{R},\mu_{\top}}$.

Example 49 *(Example 8 - μ-joinability of (23) as feasibility).* Consider $\mathcal{R}$ and μ as in Example 8 and the μ-critical pair (23), i.e., $\langle \mathsf{g}(\mathsf{h}(x), x), x\rangle$. According to Proposition 46, the feasibility problem which characterizes the μ-joinability of (23) is

$$\mathsf{g}(\mathsf{h}(c_x), c_x) \hookrightarrow^* z,\ c_x \hookrightarrow^* z \tag{52}$$

The feasibility of (52) can be automatically proved by infChecker. This proves μ-joinability of (23).

The use of the grounding operator $\_^{\downarrow}$ is essential to obtain an equivalent treatment of μ-joinability of μ-critical pairs as $\overline{\mathcal{R}^{\mu}}$-feasibility: by definition, in feasibility problems *all* variables can be instantiated; however, variables in terms s and t whose μ-joinability by μ-rewriting are being tested should *not* be instantiated, as done in rewriting sequences. By first grounding terms s and t in a μ-critical pair $\langle s, t\rangle$, we avoid such undesired instantiations. Otherwise, wrong results can be obtained.

Example 50. Consider $\mathcal{R}$ and μ as in Example 2. Let $s = \mathsf{f}(\mathsf{g}(x,x))$ and $t = \mathsf{g}(\mathsf{f}(x),\mathsf{f}(x))$ as in the example. Note that $\mathsf{f}(\mathsf{g}(x,x)) \hookrightarrow^* z, \mathsf{g}(\mathsf{f}(x),\mathsf{f}(x)) \hookrightarrow^* z$ is $\overline{\mathcal{R}^\mu}$-feasible: with $\sigma = \{x \mapsto 0, z \mapsto 0\}$, we obtain

$$\sigma(s) = \mathsf{f}(\underline{\mathsf{g}(0,0)}) \hookrightarrow \underline{\mathsf{f}(0)} \hookrightarrow \underline{\mathsf{g}(0,0)} \hookrightarrow 0 \tag{53}$$

$$\sigma(t) = \mathsf{g}(\underline{\mathsf{f}(0)},\mathsf{f}(0)) \hookrightarrow \mathsf{g}(\underline{\mathsf{g}(0,0)},\mathsf{f}(0)) \hookrightarrow \underline{\mathsf{g}(0,\mathsf{f}(0))} \hookrightarrow 0 \tag{54}$$

However, as remarked in the example, s and t are *not* μ-joinable. This problem is avoided by testing feasibility of $s^\downarrow \hookrightarrow^* z, t^\downarrow \hookrightarrow^* z$, i.e., of

$$\mathsf{f}(\mathsf{g}(c_x,c_x)) \hookrightarrow^* z,\ \mathsf{g}(\mathsf{f}(c_x),\mathsf{f}(c_x)) \hookrightarrow^* z \tag{55}$$

instead. For instance, it is not difficult to see that sequences (53) and (54) are not possible if 0 is replaced everywhere by the new constant c_x. Furthermore, the $\overline{\mathcal{R}^\mu}$-infeasibility of (55) can be automatically proved by infChecker.

6.2. LH_μ-critical pairs

In order to adapt Corollary 47 to LH_μ-pairs $\langle s,t\rangle \Leftarrow x \hookrightarrow y$, we should *not* make the new variables x and y ground: they may need to be appropriately instantiated to make the peak component $\langle s,t\rangle$ μ-joinable. In order to illustrate this observation, the next example shows that there are LH_μ-pairs where $s^\downarrow$ and $t^\downarrow$ are not μ-joinable but π is *still* μ-joinable.

Example 51. Consider $\mathcal{R}$ and μ in Example 38 and the LH_μ-critical pair

$$\langle \mathsf{f}(y),\mathsf{g}(x)\rangle \Leftarrow x \hookrightarrow y \tag{56}$$

Note that $\mathsf{f}(c_y)$ and $\mathsf{g}(c_x)$ are *not* μ-joinable: $\mathsf{g}(c_x)$ is a μ-normal form and the only μ-rewriting step on $\mathsf{f}(c_y)$ is $\underline{\mathsf{f}(c_y)} \hookrightarrow_{\mathcal{R},\mu} \mathsf{g}(c_y)$, leading to a *different* μ-normal form $\mathsf{g}(c_y)$. According to Proposition 46, this can also be proved as the $\overline{\mathcal{R}^\mu}$-infeasibility of $\mathsf{f}(c_y) \hookrightarrow^* z, \mathsf{g}(c_x) \hookrightarrow^* z$, which can be proved by infChecker. However, as shown in Example 53 below, (56) can be proved μ-joinable.

In the following result (Proposition 52), we show that μ-joinability of LH_μ-critical pairs can be handled as *provability* of a logical formula in the first-order theory $\overline{\mathcal{R}^\mu}$. The result implements the observation in Example 51 showing that, when dealing with LH_μ-critical pairs $\langle s,t\rangle \Leftarrow x \hookrightarrow y$, we should *not* replace variables x and y in s and t by c_x and c_y, respectively: the instantiation of x and y should be controlled by the conditional part $x \hookrightarrow y$ instead. All other variables in s and t, though, can be grounded.

Proposition 52. *Let $\mathcal{R}$ be a TRS, $\mu \in M_{\mathcal{R}}$, and $\pi : \langle \ell[y]_p, r\rangle \Leftarrow x \hookrightarrow y \in \mathsf{LHCP}(\mathcal{R},\mu)$ be an LH_μ-critical pair. If $\overline{\mathcal{R}^\mu} \vdash (\forall x)(\forall y)(\exists z)\, x \to y \Rightarrow \ell^{\downarrow\overline{\{x\}}}[y]_p \to^* z \wedge r^{\downarrow\overline{\{x\}}} \to^* z$ holds, then π is μ-joinable.*

Proof. If $\overline{\mathcal{R}^\mu} \vdash (\forall x)(\forall y)(\exists z)\, x \to y \Rightarrow \ell^{\downarrow\overline{\{x\}}}[y]_p \to^* z \wedge r^{\downarrow\overline{\{x\}}} \to^* z$ holds, then, for each instantiation σ of variables x and y such that $\overline{\mathcal{R}^\mu} \vdash \sigma(x) \to \sigma(y)$ holds (i.e., $\sigma(x) \hookrightarrow_{\mathcal{R},\mu} \sigma(y)$ holds), there is a term u such that $\overline{\mathcal{R}^\mu} \vdash \sigma(\ell^{\downarrow\overline{\{x\}}}[y]_p) \to^* u \wedge \sigma(r^{\downarrow\overline{\{x\}}}) \to^* u$ holds, i.e., both $\sigma(\ell^{\downarrow\overline{\{x\}}}[y]_p) \hookrightarrow^*_{\mathcal{R},\mu} u$ and $\sigma(r^{\downarrow\overline{\{x\}}}) \hookrightarrow^*_{\mathcal{R},\mu} u$ hold. Hence, $\sigma(\ell^{\downarrow\overline{\{x\}}}[y]_p)$ and $\sigma(r^{\downarrow\overline{\{x\}}})$ are μ-joinable. By Proposition 44, π is μ-joinable. □

Example 53 *($\mathcal{R}$ in Example 38 is μ-confluent).* Continuing Example 51, consider $\mathcal{R}$ and μ in Example 38. Note that, since $\mathcal{R}$ is terminating, it is μ-terminating. We have $\mathsf{ECP}(\mathcal{R},\mu) = \mathsf{LHCP}(\mathcal{R},\mu) = \{(56)\} = \{\langle \mathsf{f}(y),\mathsf{g}(x)\rangle \Leftarrow x \hookrightarrow y\}$. By using the theorem prover Prover9 [23], it is possible to show that

$$\overline{\mathcal{R}^\mu} \vdash (\forall x)(\forall y)(\exists z)\, x \to y \Rightarrow \mathsf{f}(y) \to^* z \wedge \mathsf{g}(x) \to^* z$$

holds. By Proposition 52, π is μ-joinable. Since $\mathcal{R}$ is μ-terminating, by Theorem 32).(3), it is μ-confluent.

Note that proving μ-confluence of $\mathcal{R}$ in Example 38, as done in Example 53, is *not* possible with Theorem 15 as the LHRV-condition is always required for that purpose and $\mathcal{R}$ does not fulfill the LHRV-condition because $\mathsf{LHCP}(\mathcal{R},\mu) \neq \emptyset$ (see Proposition 21). As for proofs of non-μ-joinability of LH_μ-critical pairs, we have:

Proposition 54. *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $\pi : \langle s,t\rangle \Leftarrow x \hookrightarrow y$ be an LH_μ-critical pair. Let $z \in \mathcal{X}$ be such that $z \notin \{x, y\}$. If*

$$x \hookrightarrow y, s^{\downarrow\overline{\{x,y\}}} \hookrightarrow^* z, t^{\downarrow\overline{\{x\}}} \to^* z \tag{57}$$

is $\overline{\mathcal{R}^\mu}$-infeasible, then π is not μ-joinable.

Proof. By contradiction. If π is joinable, then for all substitutions σ with $\mathcal{D}om(\sigma) = \{x, y\}$, if $\sigma(x) \hookrightarrow_{\mathcal{R}} \sigma(y)$ holds, then there is a term u such that $\sigma(s) \hookrightarrow^*_{\mathcal{R}} u$ and $\sigma(t) \hookrightarrow^*_{\mathcal{R}} u$. Since σ instantiates no variable in s and t except x and y, we can also say that $\sigma(s^{\downarrow \overline{\{x,y\}}}) \hookrightarrow^*_{\mathcal{R}} u$ and $\sigma(t^{\downarrow \overline{\{x\}}}) \hookrightarrow^*_{\mathcal{R}} u$ (note that, by definition of LH_μ-critical pair, y cannot occur in t). Since, given a rule $\ell \to r \in \mathcal{R}$, the substitution $\sigma(x) = \ell$ and $\sigma(y) = r$ trivially satisfies $\sigma(x) \hookrightarrow_{\mathcal{R}} \sigma(y)$, if we additionally let $\sigma(z) = u$, then $x \hookrightarrow y, s^{\downarrow \overline{\{x,y\}}} \hookrightarrow^* z, t^{\downarrow \overline{\{x\}}} \to^* z$ is $\overline{\mathcal{R}^\mu}$-feasible, leading to a contradiction. $\square$

Unfortunately, Proposition 54 cannot be reversed to characterize μ-joinability of LH_μ-critical pairs as $\overline{\mathcal{R}^\mu}$-feasibility: there are non-μ-joinable LH_μ-critical pairs whose feasibility problem (57) can be solved.

Example 55. Consider $\mathcal{R}$ and μ as in Example 2 and $\pi : \langle \mathsf{f}(y), \mathsf{g}(x,x)\rangle \Leftarrow x \hookrightarrow y$ in Example 27. The sequence

$$x \hookrightarrow y, \mathsf{f}(y) \hookrightarrow^* z, \mathsf{g}(x,x) \hookrightarrow^* z \tag{58}$$

is $\overline{\mathcal{R}^\mu}$-feasible: with $\sigma(x) = \ell_{(14)} = \mathsf{h}(0)$ and $\sigma(y) = r_{(14)} = 0$, we have $\sigma(x) \hookrightarrow \sigma(y)$, $\sigma(\mathsf{f}(y)) = \mathsf{f}(0)$ and $\sigma(\mathsf{g}(x,x)) = \mathsf{g}(\mathsf{h}(0), \mathsf{h}(0))$. Since $\mathsf{f}(0) \hookrightarrow \underline{\mathsf{g}(0,0)} \hookrightarrow 0$ and $\mathsf{g}(\underline{\mathsf{h}(0)}, \mathsf{h}(0)) \hookrightarrow \underline{\mathsf{g}(0, \mathsf{h}(0))} \hookrightarrow 0$, we have that (58) is feasible, but π is *not* μ-joinable (see Example 58).

Example 55 also shows that Proposition 54 can be difficult to use in practice: depending on the considered μ-rewriting step fulfilling the feasibility condition $x \hookrightarrow y$, it is possible that the remainder of the condition holds or not. Thus, infeasibility of (58), i.e., the *absence* of substitutions σ that make all feasibility conditions to hold in $\overline{\mathcal{R}^\mu}$, *cannot* be proved because, as shown in Example 55, there is a substitution that satisfies (58). Still, π in Example 55 is *not* μ-joinable.

The following result provides a different approach where such considered steps are restricted to those directly obtained from the rules $\ell \to r$ of the TRS, which obviously permit a μ-rewriting step $\ell \hookrightarrow_{\mathcal{R},\mu} r$.

Proposition 56. *Let $\mathcal{R}$ be a TRS, $\mu \in M_{\mathcal{R}}$ and $\pi : \langle s, t\rangle \Leftarrow x \hookrightarrow y \in \mathsf{LHCP}(\mathcal{R}, \mu)$. Let $\ell \to r \in \mathcal{R}$ and σ given by $\sigma(x) = \ell^\downarrow$, $\sigma(y) = r^\downarrow$, and $\sigma(z) = c_z$ if $z \notin \{x, y\}$. If $\sigma(s)$ and $\sigma(t)$ are not μ-joinable, then π is not μ-joinable.*

Proof. Since, by definition of σ, we have that $\sigma(x) \hookrightarrow \sigma(y)$ holds, by Definition 26 and using Corollary 45, the fact that $\sigma(s)$ and $\sigma(t)$ are not μ-joinable implies that π is not μ-joinable. $\square$

Example 57 *($\mathcal{R}$ in Example 1 is not μ-confluent).* Consider $\mathcal{R}$ and μ as in Example 1 and

$$\pi : \langle \mathsf{fact}(y), \mathsf{if}(\mathsf{zero}(x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(x)) \times x)\rangle \Leftarrow x \hookrightarrow y$$

in Example 23. With $\sigma(x) = \ell^\downarrow_{(2)} = 0 + c_x$ and $\sigma(y) = r^\downarrow_{(2)} = c_x$, we obtain $\sigma(\mathsf{fact}(y)) = \mathsf{fact}(c_x)$ and

$$\sigma(\mathsf{if}(\mathsf{zero}(x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(x)) \times x)) = \mathsf{if}(\mathsf{zero}(0 + c_x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(0 + c_x)) \times (0 + c_x)).$$

Since the only μ-rewriting step on $s = \mathsf{fact}(c_x)$ is

$$s = \underline{\mathsf{fact}(c_x)} \quad \hookrightarrow \quad \mathsf{if}(\mathsf{zero}(c_x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(c_x)) \times c_x)$$

leading to a μ-normal form, and $t = \mathsf{if}(\mathsf{zero}(0 + c_x), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(0 + c_x)) \times (0 + c_x))$ only admits the following μ-rewriting step

$$\begin{aligned} t &= \mathsf{if}(\mathsf{zero}(\underline{0 + c_x}), \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(0 + c_x)) \times (0 + c_x)) \\ &\hookrightarrow \underline{\mathsf{if}(\mathsf{zero}(c_x)}, \mathsf{s}(0), \mathsf{fact}(\mathsf{p}(0 + c_x)) \times (0 + c_x)) \end{aligned}$$

leading to a *different* μ-normal form, s and t are *not* μ-joinable and by Proposition 56, π is not μ-joinable. By Theorem 32, $\mathcal{R}$ is not μ-confluent.

Example 58 *($\mathcal{R}$ in Example 2 is not μ-confluent).* Continuing the previous discussion in Example 50, consider $\mathcal{R}$ and μ as in Example 2 and $\pi : \langle \mathsf{f}(y), \mathsf{g}(x,x)\rangle \Leftarrow x \hookrightarrow y$ in Example 27. With $\sigma(x) = \ell^\downarrow_{(12)} = \mathsf{f}(c_x)$ and $\sigma(y) = r^\downarrow_{(12)} = \mathsf{g}(c_x, c_x)$, we obtain $\sigma(\mathsf{f}(y)) = \mathsf{f}(\mathsf{g}(c_x, c_x))$ and $\sigma(\mathsf{g}(x,x)) = \mathsf{g}(\mathsf{f}(c_x), \mathsf{f}(c_x))$. By Proposition 46, in order to prove their (non-)μ-joinability, we should check the sequence

$$\mathsf{f}(\mathsf{g}(c_x, c_x)) \hookrightarrow^* z,\ \mathsf{g}(\mathsf{f}(c_x), \mathsf{f}(c_x)) \hookrightarrow^* z$$

which can be proved infeasible by using infChecker. By Proposition 56, π is not μ-joinable. By Theorem 32, $\mathcal{R}$ is not μ-confluent.

Example 59 *($\mathcal{R}$ in Example 9 is not μ-confluent).* Consider $\mathcal{R}$ and μ as in Example 9 and $\pi : \langle \mathsf{g}(y, \mathsf{a}), \mathsf{c}(x)\rangle \Leftarrow x \hookrightarrow y$ in Example 28. With $\sigma(x) = \ell^{\downarrow}_{(25)} = \mathsf{a}$ and $\sigma(y) = r^{\downarrow}_{(25)} = \mathsf{b}$, we obtain $\sigma(\mathsf{g}(y, \mathsf{a})) = \mathsf{g}(\mathsf{b}, \mathsf{a})$ and $\sigma(\mathsf{c}(x)) = \mathsf{c}(\mathsf{a})$. By Proposition 46, in order to prove their (non-)μ-joinability, we should check the sequence

$$\mathsf{g}(\mathsf{b}, \mathsf{a}) \hookrightarrow^* z,\ \mathsf{c}(\mathsf{a}) \hookrightarrow^* z$$

which can be proved infeasible by using infChecker. By Proposition 56, π is *not* μ-joinable. By Theorem 32, $\mathcal{R}$ is not μ-confluent.

7. 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 at the URL showed in the introduction. The input format is an extended version of the Confluence Competition (CoCo) format [24], which is the official format used in the *confluence* (CR) category. The input is a CS-TRS $\mathcal{R}$ in TPDB format,[10] where *replacement maps* μ can be specified as well as part of the `STRATEGY CONTEXTSENSITIVE` section of the format: each function symbol f with $\mu(f) = \{i_1, \dots, i_m\}$ is given an entry $(f\ i_1\ \cdots\ i_m)$. However, only k-ary symbols f whose replacement map $\mu(f)$ imposes some replacement restrictions, i.e., such that $\mu(f) \neq \{1, \dots, k\}$, need to be specified: function symbols f of arity k with no entry in the `STRATEGY CONTEXTSENSITIVE` section are assumed to have no replacement restriction, i.e., $\mu(f) = \{1, \dots, k\}$.

Example 60. The TRS $\mathcal{R}$ and μ in Example 2, where $\mu(\mathsf{f}) = \mu(\mathsf{g}) = \mu(\mathsf{h}) = \mu(\mathsf{s}) = \{1\}$ is entered as follows:

```
(VAR x y)
(STRATEGY CONTEXTSENSITIVE
  (g 1)
)
(RULES
  f(x) -> g(x,x)
  g(0,x) -> 0
  g(s(x),y) -> s(x)
  h(0) -> 0
  h(s(x)) -> 0
)
```

Note that, since f and h are monadic and $\mu(\mathsf{f}) = \mu(\mathsf{h}) = \{1\}$, no special entry in the `STRATEGY CONTEXTSENSITIVE` section is required.

The implementation of CONFident (see [28] for missing details) is based on a divide and conquer schema where, given an input problem, a set of techniques is orderly applied to it according to a predefined strategy. The techniques can simplify the problem, reduce it into a set of simpler problems or just give a positive or negative answer. Our proof strategy is based on experimentation: we first apply techniques that simplify the problems and reduce them into simpler problems (e.g., remove unnecessary rules $t \to t$ for some term t). When all simplification techniques have been applied, we analyze the problem in order to extract good properties that guide the strategy (left-linearity, μ-termination, etc.). Then we calculate its extended μ-critical pairs and check them for μ-joinability using the results in Section 6, calling infChecker to prove (in)feasibility, or by applying faster methods in specific situations (e.g., two different μ-normal forms are immediately qualified as non-μ-joinable). If proofs of μ-termination are required, they are obtained from MU-TERM. When required, provability of first-order formulas (as in Example 53) is checked with Prover9. If the final answer is YES or NO, the tool displays a report in plain text. Otherwise, MAYBE is returned.

7.1. Experimental evaluation

Since termination of *CSR* is one of the subcategories of the International Termination and Complexity Competition (termCOMP), see e.g., [8, Section 3.1], in order to test our tool we have used the collection of CS-TRSs which is available for such a competition, which consists of 108 examples. The timeout limit to provide an answer is the usual one of CoCo: 60 seconds. Table 1 summarizes the obtained results. The table shows the outcome of two versions of the tool: the first row shows the results for a version of the tool which only uses μ-critical pairs (i.e., pairs in $\mathsf{CP}(\mathcal{R}, \mu)$) together with Theorem 15. The second row displays the results when extended critical pairs (i.e., $\mathsf{ECP}(\mathcal{R}, \mu)$) and Theorem 32 are used. Note that version ECPs disproves μ-confluence of 49 examples which could not be handled by the CPs version. Furthermore, it proves μ-confluence of 2 examples which could not be handled with CPs only. In particular, we are able to prove

[10] See http://zenon.dsic.upv.es/muterm/?page_id=31.

Table 1
Proofs of μ-confluence with CPs and ECPs.

	Yes	No	Maybe	Solved (Yes+No)	Total
CPs	25	30	53	55	108
ECPs	27	79	2	106	108

Table 2
μ-confluence and $\mu^{can}_{\mathcal{R}}$-confluence of our example TRSs.

$\mathcal{R}$ and μ in	LHRV	μ-confluent?	LHRV($\mathcal{R}, \mu^{can}_{\mathcal{R}}$)	$\mu^{can}_{\mathcal{R}}$-confluent?
Example 1	No	No (Example 57)	Yes	Yes
Example 2	No	No (Example 58)	Yes	Yes
Example 8	No	No (Example 16)	Yes	No
Example 9	No	No (Example 33)	Yes	No
Example 9 & μ' in Example 14	Yes	Yes (Example 37)	Yes	No
Example 38	No	Yes (Example 53)	Yes	Yes

μ-confluence of TRSs not satisfying the LHRV-condition (e.g., $\mathcal{R}$ and μ in Example 38). Full details of the proofs can be found in the benchmarks section of the web site of the tool.

8. Discussion: how to achieve the LHRV-condition?

In general, given a TRS $\mathcal{R}$ one can freely associate a replacement map μ to impose replacement restrictions on symbols f occurring in the rules of $\mathcal{R}$. However, if the LHRV-condition does not hold, $\mathcal{R}$ is often non-μ-confluent. Thus, the following question naturally arises:

How to achieve the LHRV-condition?

First note that, as a consequence of Proposition 13.(1), the use of $\mu_\perp$ guarantees the LHRV-condition for any TRS $\mathcal{R}$. In general, though, $\mu_\perp$ is a very restrictive choice of replacement map as it forbids all reductions below the root of terms.

A more natural requirement on $\mu(f)$ for each symbol f is that the replacement restrictions imposed by a replacement map μ do *not* prevent the rules defining f from being applied to *calls* $f(t_1, \ldots, t_k)$. As discussed in [20, Section 5], this is requirement is captured by the *canonical replacement map*.

8.1. Canonical replacement map and LHRV-condition

Given a TRS $\mathcal{R}$, the canonical replacement map $\mu^{can}_{\mathcal{R}}$ is systematically associated to $\mathcal{R}$ as

the most restrictive replacement map ensuring that, for all rules $\ell \to r \in \mathcal{R}$, the non-variable subterms of ℓ are all active [20, Section 5].

Example 61. For $\mathcal{R}$ in Example 1, we have $\mu^{can}_{\mathcal{R}}(\mathsf{s}) = \mu^{can}_{\mathcal{R}}(\mathsf{fact}) = \emptyset$, and $\mu^{can}_{\mathcal{R}}(f) = \{1\}$ for any other non-constant symbol f. It is not difficult to see that *all variables in the left-hand sides of the rules in $\mathcal{R}$ are* frozen *now* (with respect to $\mu^{can}_{\mathcal{R}}$).

As a consequence of Proposition 13.(2), if all variables occur frozen in left-hand sides ℓ of rules $\ell \to r \in \mathcal{R}$, then the LHRV-condition holds. When using $\mu^{can}_{\mathcal{R}}$, variables will often remain frozen in the left-hand sides of rules as only non-variable subterms must be active. Actually, all TRSs $\mathcal{R}$ considered in this paper satisfy LHRV($\mathcal{R}, \mu^{can}_{\mathcal{R}}$). Table 2 reports on the LHRV-condition and confluence of *CSR* for these examples, regarding both μ in the corresponding example (with pointers to the examples of the paper where μ-confluence is proved or disproved), and $\mu^{can}_{\mathcal{R}}$ (with proofs or refutations of $\mu^{can}_{\mathcal{R}}$-confluence obtained by CONFident in all cases). However, using the canonical replacement map does *not* guarantee the LHRV-condition.

Example 62. Consider the following TRS $\mathcal{R}_{\mathsf{half}}$ defining a function half, which computes the quotient of integer division of natural numbers (in Peano's notation) by 2:

$$\mathsf{half}(0) \to 0 \tag{59}$$

$$\mathsf{half}(\mathsf{s}(0)) \to 0 \tag{60}$$

$$\mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)) \tag{61}$$

Table 3
μ-confluence vs. $\mu^{can}_{\mathcal{R}}$-confluence (termCOMP TRSs).

Rep. map	#LHRV	Yes	No	Maybe	Solved (Yes+No)	Total
μ	40	27	79	2	106	108
$\mu^{can}_{\mathcal{R}}$	61	52	50	2	102	108

The canonical replacement map for $\mathcal{R}_{\mathsf{half}}$ is $\mu^{can}_{\mathcal{R}_{\mathsf{half}}}(\mathsf{half}) = \mu^{can}_{\mathcal{R}_{\mathsf{half}}}(\mathsf{s}) = \{1\}$. Note that $\mu^{can}_{\mathcal{R}_{\mathsf{half}}}$ imposes no replacement restriction on any symbol. Trivially, $\mathsf{LHRV}(\mathcal{R}_{\mathsf{half}}, \mu^{can}_{\mathcal{R}_{\mathsf{half}}})$ holds. Actually, we can think of $(\mathcal{R}, \mu^{can}_{\mathcal{R}_{\mathsf{half}}})$ as a TRS, i.e., $\hookrightarrow_{\mathcal{R}_{\mathsf{half}},\mu^{can}_{\mathcal{R}_{\mathsf{half}}}}$ and $\to_{\mathcal{R}_{\mathsf{half}}}$ coincide. Therefore, since $\mathcal{R}_{\mathsf{half}}$ is orthogonal, it is confluent, and also $\mu^{can}_{\mathcal{R}_{\mathsf{half}}}$-confluent.

Now consider the TRS $\mathcal{R}_{\mathsf{first}}$ defining a function first, which returns a sublist with the first n components of a list:

$$\mathsf{first}(0, xs) \to [\,] \tag{62}$$

$$\mathsf{first}(\mathsf{s}(n), x : xs) \to x : \mathsf{first}(n, xs) \tag{63}$$

We have $\mu^{can}_{\mathcal{R}_{\mathsf{first}}}(:) = \mu^{can}_{\mathcal{R}_{\mathsf{first}}}(\mathsf{s}) = \emptyset$, and $\mu^{can}_{\mathcal{R}_{\mathsf{first}}}(\mathsf{first}) = \{1, 2\}$. Since $\mathcal{R}_{\mathsf{first}}$ is orthogonal, it has no $\mu^{can}_{\mathcal{R}_{\mathsf{first}}}$-critical pair. Since $\mathsf{LHRV}(\mathcal{R}_{\mathsf{first}}, \mu^{can}_{\mathcal{R}_{\mathsf{first}}})$ holds, there is no $\mathsf{LH}_{\mu^{can}_{\mathcal{R}_{\mathsf{first}}}}$-critical pair. Hence, $\mathcal{R}_{\mathsf{first}}$ is $\mu^{can}_{\mathcal{R}_{\mathsf{first}}}$-orthogonal (see Definition 35). By Corollary 36, $\mathcal{R}_{\mathsf{first}}$ is $\mu^{can}_{\mathcal{R}_{\mathsf{first}}}$-confluent.

Now consider the TRS $\mathcal{R}$ obtained as the union of $\mathcal{R}_{\mathsf{half}}$ and $\mathcal{R}_{\mathsf{first}}$, i.e., $\mathcal{R} = \mathcal{R}_{\mathsf{half}} \cup \mathcal{R}_{\mathsf{first}}$. We have $\mu^{can}_{\mathcal{R}}(:) = \emptyset$, $\mu^{can}_{\mathcal{R}}(\mathsf{half}) = \mu^{can}_{\mathcal{R}}(\mathsf{s}) = \{1\}$, and $\mu^{can}_{\mathcal{R}}(\mathsf{first}) = \{1, 2\}$. Since $\mathcal{R}$ is also orthogonal, there is no $\mu^{can}_{\mathcal{R}}$-critical pair. However, we note that (63) is $\mathsf{LH}_{\mu^{can}_{\mathcal{R}}}$-negative: $n \in \mathcal{V}ar^{\mu^{can}_{\mathcal{R}}}(\ell_{(63)}) \cap \mathcal{V}ar^{\not\mu^{can}_{\mathcal{R}}}(r_{(63)})$. Then, there is an LH_{μ}-critical pair:

$$\pi : \langle \mathsf{first}(\mathsf{s}(y), x : xs), x : \mathsf{first}(n, xs)\rangle \Leftarrow n \hookrightarrow y$$

According to Proposition 56, with (59) as the rule $\ell \to r$ in the proposition, since $\mathsf{first}(\mathsf{s}(0), c_x : c_{xs})$ and $c_x : \mathsf{first}(\mathsf{half}(0), c_{xs})$ are not $\mu^{can}_{\mathcal{R}}$-joinable (this can be proved using Proposition 46 and infChecker), we conclude that π is not $\mu^{can}_{\mathcal{R}}$-joinable. By Theorem 32.(1), $\mathcal{R}$ is not $\mu^{can}_{\mathcal{R}}$-confluent.

Example 62 shows that, although the use of the canonical replacement map could 'naturally' guarantee the LHRV-condition for TRSs, like $\mathcal{R}_{\mathsf{half}}$ and $\mathcal{R}_{\mathsf{first}}$, defining specific functions which could be useful for some purposes, when trying to use them together, as part of a 'collecting' TRS $\mathcal{R}$, the LHRV-condition may fail to hold. Moreover, as discussed in [20, Section 8], the LHRV-condition is *not* necessarily preserved when restricting or relaxing the replacement map μ. This is in sharp contrast with μ-termination, which is preserved if the replacement map is made more restrictive.

Example 63. Consider $\mathcal{R}$ and μ in Example 1. As remarked in [20, Example 1.1], $\mathcal{R}$ is μ-terminating. CONFident is able to show that $\mathcal{R}$ is *not* μ-confluent (see also Example 57). If we let μ' be as μ except $\mu'(\mathsf{fact}) = \emptyset$, then $\mathcal{R}$ is μ'-terminating (as μ' is *more restrictive* than μ), but it is now μ'-confluent (CONFident can prove it). However, if we restrict further, e.g., by letting μ'' be as μ' except $\mu''(\mathsf{add}) = \{1\}$, then $\mathcal{R}$ is still μ''-terminating, but it is not μ''-confluent now (again, CONFident proves it).

8.2. Canonical replacement map as 'default' replacement map for TRSs?

Despite the discussion in the previous section, Table 2 is encouraging: it suggests that a systematical use of $\mu^{can}_{\mathcal{R}}$ is good to achieve the LHRV-condition and also to obtain confluence of *CSR*. At a purely experimental level, this is somehow supported by the benchmarks summarized in Table 3, where we consider again the TRSs $\mathcal{R}$ in termCOMP's CS-TRS collection (first row), but we also use the canonical replacement map $\mu^{can}_{\mathcal{R}}$ instead of the specified replacement map now (second row). The second column of the table (labeled #LHRV) shows the number of systems where the LHRV condition holds (with respect to μ or $\mu^{can}_{\mathcal{R}}$). We can see that the use of $\mu^{can}_{\mathcal{R}}$ 'reinforces' the LHRV-condition and also confluence of *CSR*: whereas only 27 examples are proved confluent for the TRSs $\mathcal{R}$ with the original replacement maps μ,[11] when the canonical replacement map is considered, 52 TRSs are proved $\mu^{can}_{\mathcal{R}}$-confluent. The increase in the number of examples that can be proved confluent is similar to the increase in the number of TRSs fulfilling the LHRV-condition.

Since using the canonical replacement map has important computational advantages (see [20, Sections 5 and 9]), it is a natural choice already. The discussion above provides additional, although 'heuristic', arguments to start with $\mu^{can}_{\mathcal{R}}$ when using *CSR*.

[11] Most of the 108 CS-TRSs included in the termCOMP benchmark collection have been taken from examples used in papers related to termination of *CSR* along the last 25 years. Typically, they are obtained from *nonterminating* TRSs $\mathcal{R}$ by adding a replacement map μ to (try to) make them μ-terminating. The particular choice of μ usually tries to illustrate the application of some technique to prove or disprove termination of *CSR*. Achieving other computational properties of *CSR* (e.g., μ-confluence), was not in the agenda.

9. Related work

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 connection has not been used in verification of computational properties like confluence, which is the main focus of this paper.

As remarked above, research on confluence of CSR goes back to [17] and [19] (journal version of the conference paper [17]). In [20, Section 8], besides surveying existing results on (local) confluence of *CSR*, some applications of μ-confluence in proofs of confluence properties of rewriting are reported ([20, Section 8.5 and also Section 9.4]) and some results are provided. In particular, new results about *ground* confluence of *CSR*, i.e., confluence of *CSR* where rewrite sequences start with a ground term, were proved ([20, Theorem 8.14]). Although [20, Theorem 8.14] does *not* require the LHRV-condition, restrictions like termination, ground confluence, and completely definedness of $\mathcal{R}$ are required instead.

Remark 64 *(μ-confluence and ground μ-confluence).* The TRS $\mathcal{R}$ and μ in Example 2 is ground μ-confluent (see [20, Example 8.15]). However, as shown in Example 2, $\mathcal{R}$ is not μ-confluent.[12]

In this respect, an important contribution of this paper is providing a homogeneous treatment of the LHRV-condition, usually required in virtually all previous results on confluence of *CSR*, but whose *absence* had not been exploited to (try to) conclude non-μ-confluence to date. Theorem 30 permits to 'embed' the LHRV-condition as a new class of μ-critical pairs to explore its role in determining local μ-confluence using μ-joinability. Theorem 32 generalizes the old results (summarized in Theorem 15) to deal with arbitrary TRSs, without requiring the LHRV-condition. Example 53 shows that, besides dramatically improving the ability to disprove μ-confluence (as witnessed by our benchmarks), this also has a 'positive' impact in proofs of μ-confluence.

Recently, Andrianarivelo and Réty [1] have used the analysis of confluence of *CSR* in [17] to investigate confluence of *prefix-constrained rewrite systems* [16], where restrictions on rewriting steps can be specified using automata techniques, being *CSR* a particular case of prefix-constrained rewriting. It is possible that our new results could be of some use to improve the analysis of confluence of prefix-constrained rewriting or similar systems.

10. Conclusions and future work

We have defined the notion of LH$_\mu$-critical pair which captures how variables which occur active in the left-hand side of a rule but also frozen somewhere else in the rule may affect confluence. The μ-critical pairs introduced in [17] (see Definition 7) together with LH$_\mu$-critical pairs (Definition 20) conform the *extended μ-critical pairs* (Definition 29) whose μ-joinability characterizes local μ-confluence (Theorem 30) and, together with μ-termination, also μ-confluence (Theorem 32). We have also redefined the notion of μ-orthogonality of TRSs (Definition 35), so that it implies μ-confluence, as in the usual results for unrestricted rewriting (Corollary 36).

We have characterized μ-joinability of μ-critical pairs $\langle s,t\rangle$ as the μ-joinability of the 'grounded' version $\langle s^{\downarrow},t^{\downarrow}\rangle$ obtained by replacing in s and t all occurrences of variables x by constants c_x (Corollary 47). We have also investigated how to deal with these μ-joinability checkings of μ-critical pairs as feasibility problems (Proposition 46). We have given sufficient conditions for proving and disproving μ-joinability of LH$_\mu$-critical pairs (Propositions 52, 54, and 56). In the last two cases, we have shown how to treat such (non-)μ-joinability problems as (in)feasibility problems which can be automatically handled using existing tools like infChecker. We have implemented our techniques as part of the confluence tool CONFident, which is available for use as a web application.

The experimental evaluation of the tool is encouraging: on the external benchmark suite we used (108 examples of CS-TRSs from the *CSR* subcategory of termCOMP) we are able to prove/disprove the μ-confluence of 106 of the examples (more than 98%). On the same collection of TRSs $\mathcal{R}$ but systematically using the canonical replacement map $\mu^{can}_{\mathcal{R}}$, we are able to handle 102 examples (more than 94%). The experiments also support the idea that using the canonical replacement map $\mu^{can}_{\mathcal{R}}$ often reinforces the LHRV-condition and also confluence of *CSR*, as discussed in Section 8. This provides more arguments to use the canonical replacement map in computations with *CSR*, as already discussed in [20].

In 2021, the Steering Committee of the *Confluence Competition* decided to launch a new subcategory of the competition devoted to confluence of *CSR*.[13] We envisage the participation of CONFident in the forthcoming edition of the competition. Although we are not aware of any other tool which is able to prove confluence of *CSR*, this situation will hopefully change thanks to this opportunity, which will probably promote new research in confluence of *CSR*. In this sense, an important aspect to be explored in the future is *modularity* of confluence of *CSR*. Modularity issues for *CSR* are underexplored to date. With the exception of [10], concerning modularity of termination of *CSR*, we are not aware of any other development. Therefore, this is an important subject for future work.

[12] In [19, Example 19] the μ-confluence of $\mathcal{R}$ is wrongly claimed. Only ground μ-confluence of $\mathcal{R}$ holds.

[13] See http://project-coco.uibk.ac.at/2022/categories/csr.php.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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

References

[1] Nirina Andrianarivelo, Pierre Réty, Confluence of prefix-constrained rewrite systems, in: Hélène Kirchner (Ed.), 3rd International Conference on Formal Structures for Computation and Deduction, FSCD 2018, July 9-12, 2018, Oxford, UK, in: LIPIcs, vol. 108, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 6:1–6:15.
[2] Franz Baader, Tobias Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.
[3] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, Carolyn L. Talcott, All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic, Lecture Notes in Computer Science, vol. 4350, Springer, 2007.
[4] Nachum Dershowitz, Jean-Pierre Jouannaud, Rewrite systems, in: Jan van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, Elsevier and MIT Press, 1990, pp. 243–320.
[5] Nachum Dershowitz, Mitsuhiro Okada, G. Sivakumar, Confluence of conditional rewrite systems, in: Stéphane Kaplan, Jean-Pierre Jouannaud (Eds.), Conditional Term Rewriting Systems, 1st International Workshop, Orsay, France, July 8–10, 1987, Proceedings, in: Lecture Notes in Computer Science, vol. 308, Springer, 1987, pp. 31–44.
[6] Melvin Fitting, First-Order Logic and Automated Theorem Proving, second edition, Graduate Texts in Computer Science, Springer, 1996.
[7] Jürgen Giesl, Cornelius Aschermann, Marc Brockschmidt, Fabian Emmes, Florian Frohn, Carsten Fuhs, Jera Hensel, Carsten Otto, Martin Plücker, Peter Schneider-Kamp, Thomas Ströder, Stephanie Swiderski, René Thiemann, Analyzing program termination and complexity automatically with AProVE, J. Autom. Reason. 58 (1) (2017) 3–31, https://doi.org/10.1007/s10817-016-9388-y.
[8] Jürgen Giesl, Albert Rubio, Christian Sternagel, Johannes Waldmann, Akihisa Yamada, The termination and complexity competition, in: Dirk Beyer, Marieke Huisman, Fabrice Kordon, Bernhard Steffen (Eds.), Tools and Algorithms for the Construction and Analysis of Systems - 25 Years of TACAS: TOOLympics, Held as Part of ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings, Part III, in: Lecture Notes in Computer Science, vol. 11429, Springer, 2019, pp. 156–166.
[9] Joseph A. Goguen, José Meseguer, Models and equality for logical programming, in: Hartmut Ehrig, Robert A. Kowalski, Giorgio Levi, Ugo Montanari (Eds.), 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), in: Lecture Notes in Computer Science, vol. 250, Springer, 1987, pp. 1–22.
[10] Bernhard Gramlich, Salvador Lucas, Modular termination of context-sensitive rewriting, in: Proceedings of the 4th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, October 6-8, 2002, Pittsburgh, PA, USA (Affiliated with PLI 2002), ACM, 2002, pp. 50–61.
[11] Bernhard Gramlich, Salvador Lucas, Generalizing Newman's lemma for left-linear rewrite systems, in: Frank Pfenning (Ed.), Term Rewriting and Applications, 17th International Conference, RTA 2006, Seattle, WA, USA, August 12–14, 2006, Proceedings, in: Lecture Notes in Computer Science, vol. 4098, Springer, 2006, pp. 66–80.
[12] Raúl Gutiérrez, Salvador Lucas, Automatically proving and disproving feasibility conditions, in: Nicolas Peltier, Viorica Sofronie-Stokkermans (Eds.), Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II, in: Lecture Notes in Computer Science, vol. 12167, Springer, 2020, pp. 416–435.
[13] Raúl Gutiérrez, Salvador Lucas, MU-TERM: verify termination properties automatically (system description), in: Nicolas Peltier, Viorica Sofronie-Stokkermans (Eds.), Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II, in: Lecture Notes in Computer Science, vol. 12167, Springer, 2020, pp. 436–447.
[14] Raúl Gutiérrez, Salvador Lucas, Miguel Vítores, Confluence of conditional rewriting in logic form, in: Mikołaj Bojańczy, Chandra Chekuri (Eds.), 41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021), in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 213, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2021, pp. 44:1–44:18, https://drops.dagstuhl.de/opus/volltexte/2021/15555.
[15] Gérard P. Huet, Confluent reductions: abstract properties and applications to term rewriting systems: abstract properties and applications to term rewriting systems, J. ACM 27 (4) (1980) 797–821, https://doi.org/10.1145/322217.322230.
[16] Florent Jacquemard, Yoshiharu Kojima, Masahiko Sakai, Term rewriting with prefix context constraints and bottom-up strategies, in: Amy P. Felty, Aart Middeldorp (Eds.), Automated Deduction - CADE-25 - 25th International Conference on Automated Deduction, Berlin, Germany, August 1–7, 2015, Proceedings, in: Lecture Notes in Computer Science, vol. 9195, Springer, 2015, pp. 137–151.
[17] Salvador Lucas, Context-sensitive computations in confluent programs, in: Herbert Kuchen, S. Doaitse Swierstra (Eds.), PLILP, in: Lecture Notes in Computer Science, vol. 1140, Springer, 1996, pp. 408–422.
[18] Salvador Lucas, Termination of context-sensitive rewriting by rewriting, in: Friedhelm Meyer auf der Heide, Burkhard Monien (Eds.), ICALP, in: Lecture Notes in Computer Science, vol. 1099, Springer, 1996, pp. 122–133.
[19] Salvador Lucas, Context-sensitive computations in functional and functional logic programs, J. Funct. Logic Program. 1998 (1) (1998), http://danae.uni-muenster.de/lehre/kuchen/JFLP/articles/1998/A98-01/A98-01.html.
[20] Salvador Lucas, Context-sensitive rewriting, ACM Comput. Surv. 53 (4) (2020) 78, https://doi.org/10.1145/3397677.
[21] Salvador Lucas, Applications and extensions of context-sensitive rewriting, J. Log. Algebraic Methods Program. 121 (2021) 100680, https://doi.org/10.1016/j.jlamp.2021.100680.
[22] Salvador Lucas, Derivational complexity and context-sensitive rewriting, J. Autom. Reason. 65 (8) (2021) 1191–1229, https://doi.org/10.1007/s10817-021-09603-1.
[23] William McCune, Prover9 & Mace4, Technical report, 2005–2010, http://www.cs.unm.edu/~mccune/prover9/.
[24] A. Middeldorp, J. Nagele, K. Shintani, CoCo 2019: report on the eight confluence competition, Int. J. Softw. Tools Technol. Transf. 23 (2021) 905–916, https://doi.org/10.1007/s10009-021-00620-4.
[25] M.H.A. Newman, On theories with a combinatorial definition of "equivalence", Ann. Math. 43 (2) (1942) 223–243, http://www.jstor.org/stable/1968867.
[26] Michael J. O'Donnell, Computing in Systems Described by Equations, Lecture Notes in Computer Science, vol. 58, Springer, 1977.
[27] Enno Ohlebusch, Advanced Topics in Term Rewriting, Springer, 2002.
[28] Miguel Vítores, Confident: a tool for confluence analysis of rewriting systems (master thesis), April 2022.