Certified Rule Labeling*

Julian Nagele and Harald Zankl

Institute of Computer Science, University of Innsbruck, Austria
{julian.nagele|harald.zankl}@uibk.ac.at

Abstract

The rule labeling heuristic aims to establish confluence of (left-)linear term rewrite systems via decreasing diagrams. We present a formalization of a confluence criterion based on the interplay of relative termination and the rule labeling in the theorem prover Isabelle. Moreover, we report on the integration of this result into the certifier CeTA, facilitating the checking of confluence certificates based on decreasing diagrams for the first time. The power of the method is illustrated by an experimental evaluation on a (standard) collection of confluence problems.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, F.4 Mathematical Logic and Formal Languages

Keywords and phrases term rewriting, confluence, decreasing diagrams, certification

Digital Object Identifier 10.4230/LIPIcs.RTA.2015.x

Category Regular Research Paper

1 Introduction

Confluence is an important property of rewrite systems as it ensures unique normal forms. The recent achievements in confluence research have enabled a competition[1] where automated tools try to establish/refute confluence. As the proofs produced by these tools are often complicated and large, there is interest in checking them within a trustable certifier.

Decreasing diagrams [15] provide a complete characterization of confluence for abstract rewrite systems whose convertibility classes are countable. As a criterion for abstract rewrite systems, they can be applied to first- and higher-order rewriting, including term rewriting and the λ-calculus. In this paper we build upon the recent Isabelle formalization of decreasing diagrams (see [28, 29]) and specialize it from abstract rewriting to term rewriting. Moreover, we formalize the rule labeling and present a mechanized proof of the following result (see [31, Corollary 16]):

▶ **Theorem 1.** *A left-linear term rewrite system is confluent if its duplicating rules terminate relative to its other rules and all its critical peaks are decreasing for the rule labeling.*

This result is an adequate candidate for a formalization because of the following reasons. On the one hand, regarding the aspect of automation, it is easily implementable as the relative termination requirement can be outsourced to external (relative) termination provers and the rule labeling heuristic has already been implemented successfully [1, 7]. Furthermore, it is a powerful criterion as demonstrated by an experimental evaluation in Section 6. On the other hand, regarding the aspect of formalization, it is challenging because it involves the combination of different labeling functions (in the sense of [31]). Hence, in our formalization

* This research is supported by FWF (Austrian Science Fund) project P27528.
[1] http://coco.nue.riec.tohoku.ac.jp/2014

26th International Conference on Rewriting Techniques and Applications (RTA'15).
Editor: Maribel Fernández; pp. 1–16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Theorem 1 is not established directly, but obtained as a corollary of more general results. This paves the way for reusing the formalization described here when tackling the remaining criteria in [31].

We based our formalization on the **Isa**belle **F**ormalization **o**f **R**ewriting (IsaFoR) [27] and extended it by the theories `Decreasing_Diagrams2.thy` and `Rule_Labeling_Impl.thy`, which amount to approximately 3500 lines of Isabelle in Isar style. IsaFoR contains executable check functions for each formalized proof technique together with formal proofs that whenever such a check is accepted, the technique is applied correctly. Then Isabelle's code-generation facility is used to obtain a trusted Haskell program, i.e., the certifier CeTA, which is capable of checking proof certificates in CPF [22] (certification problem format).[2] We suitably extended CPF to represent proofs according to Theorem 1 and implemented dedicated check functions in our formalization, enabling CeTA to inspect, i.e., certify such confluence proofs. Typically, these proofs are generated by automated confluence tools. (See Footnote 1 for details.)

A preliminary result of our formalization has already been proved useful in the latest edition of the confluence competition (CoCo 2014), where CeTA certified confluence proofs for *linear* rewrite systems based on the rule labeling (among others). The main challenge in lifting the result from linear to left-linear rewrite systems has not been the relative termination requirement per se, which vacuously holds in the linear case, but the interplay of the relative termination condition with the rule labeling, which is crucial in the the proof of Theorem 1, albeit in the statement of the result these concepts are clearly separated. Besides, to establish decreasingness of variable peaks (involving non-right-linear rules) more details about the joining sequences were needed than the existing theories in IsaFoR provided.

The remainder of this paper is organized as follows. Preliminaries are introduced in the next section. The interplay of several labeling functions favors the notion of *extended local decreasingness* [7], which is proved to imply *local decreasingness* in Section 3, where also the connection to the existing formalization of decreasing diagrams for abstract rewrite systems [28, 29] is established. Afterwards, Section 4 lifts extended local decreasingness from abstract rewriting to results for term rewriting that are parametrized by a labeling. Section 5 instantiates these results with concrete labeling functions to obtain corollaries that ensure confluence. Section 6 presents an experimental evaluation, before we conclude in Section 7.

The full formalization is available from the URL in Footnote 2.

2 Preliminaries

We assume familiarity with rewriting [25] and decreasing diagrams [15]. Basic knowledge of Isabelle [14] is not essential but experience with an interactive theorem prover might be helpful.

Let $\mathcal{F}$ be a signature and $\mathcal{V}$ a set of variables disjoint from $\mathcal{F}$. By $\mathcal{T}(\mathcal{F},\mathcal{V})$, we denote the set of terms over $\mathcal{F}$ and $\mathcal{V}$. Positions are strings of positive natural numbers, i.e., elements of $\mathbb{N}_+^*$. We write $q \leqslant p$ if $qq' = p$ for some position q', in which case $p\backslash q$ is defined to be q'. Furthermore $q < p$ if $q \leqslant p$ and $q \neq p$. Finally, $q \parallel p$ if neither $q \leqslant p$ nor $p < q$. Positions are used to address subterm occurrences. The set of positions of a term t is defined as $\mathcal{P}\mathsf{os}(t) = \{\epsilon\}$ if t is a variable and as $\mathcal{P}\mathsf{os}(t) = \{\epsilon\} \cup \{iq \mid 1 \leqslant i \leqslant n \text{ and } q \in \mathcal{P}\mathsf{os}(t_i)\}$ if $t = f(t_1,\dots,t_n)$. The subterm of t at position $p \in \mathcal{P}\mathsf{os}(t)$ is defined as $t|_p = t$ if $p = \epsilon$ and as $t|_p = t_i|_q$ if $p = iq$ and $t = f(t_1,\dots,t_n)$. We write $s[t]_p$ for the result of replacing the occurrence of

[2] IsaFoR/CeTA and CPF are available at `http://cl-informatik.uibk.ac.at/software/ceta/`.

$s|_p$ with t in s. The set of function symbol positions $\mathcal{P}\mathsf{os}_{\mathcal{F}}(t)$ is $\{p \in \mathcal{P}\mathsf{os}(t) \mid t|_p \notin \mathcal{V}\}$ and $\mathcal{P}\mathsf{os}_{\mathcal{V}}(t) = \mathcal{P}\mathsf{os}(t) \setminus \mathcal{P}\mathsf{os}_{\mathcal{F}}(t)$.

A rewrite rule is a pair of terms (l, r), written $l \to r$.[3] A rewrite rule $l \to r$ is duplicating if $|l|_x < |r|_x$ for some $x \in \mathcal{V}$. Here the expression $|t|_x$ indicates the number of occurrences of the variable x in term t. A term rewrite system (TRS) is a signature together with a set of rewrite rules over this signature. In the sequel, signatures are left implicit. By $\mathcal{R}_{\mathsf{d}}$ and $\mathcal{R}_{\mathsf{nd}}$, we denote the duplicating and non-duplicating rules of a TRS $\mathcal{R}$, respectively. A rewrite relation is a binary relation on terms that is closed under contexts and substitutions. For a TRS $\mathcal{R}$ we define $\to_{\mathcal{R}}$ (often written as $\to$) to be the smallest rewrite relation that contains $\mathcal{R}$. As usual $\to^{=}$, $\to^{+}$, and $\to^{*}$ denote the reflexive, transitive, and reflexive and transitive closure of $\to$, respectively, while $\to^{n}$ denotes the n-fold composition of $\to$.

A relative TRS $\mathcal{R}/\mathcal{S}$ is a pair of TRSs $\mathcal{R}$ and $\mathcal{S}$ with the induced rewrite relation $\to_{\mathcal{R}/\mathcal{S}} = \to_{\mathcal{S}}^{*} \cdot \to_{\mathcal{R}} \cdot \to_{\mathcal{S}}^{*}$. Sometimes we identify a TRS $\mathcal{R}$ with the relative TRS $\mathcal{R}/\varnothing$ and vice versa. A TRS $\mathcal{R}$ is terminating (relative to a TRS $\mathcal{S}$) if $\to_{\mathcal{R}}$ ($\to_{\mathcal{R}/\mathcal{S}}$) is well-founded.

A critical overlap $(l_1 \to r_1, p, l_2 \to r_2)_\mu$ of a TRS $\mathcal{R}$ consists of variants $l_1 \to r_1$ and $l_2 \to r_2$ of rewrite rules in $\mathcal{R}$ without common variables, a position $p \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l_2)$, and a most general unifier μ of l_1 and $l_2|_p$. From a critical overlap $(l_1 \to r_1, p, l_2 \to r_2)_\mu$ we obtain a critical peak $l_2\mu[r_1\mu]_p \leftarrow l_2\mu \to r_2\mu$ and a critical pair $l_2\mu[r_1\mu]_p \leftarrow\!\!\rightarrow r_2\mu$.

If $l \to r \in \mathcal{R}$, p is a position, and σ is a substitution we call the triple $\pi = \langle p, l \to r, \sigma\rangle$ a redex pattern, and write p_π, l_π, r_π, σ_π for its position, left-hand side, right-hand side, and substitution, respectively. We write $\to^{\pi}$ (or $\to^{p_\pi, l_\pi \to r_\pi, \sigma_\pi}$) for a rewrite step at position p_π using the rule $l_\pi \to r_\pi$ and the substitution σ_π. A redex pattern π matches a term t if $t|_{p_\pi} = l_\pi\sigma_\pi$, which is then called a redex.

Let π_1 and π_2 be redex patterns that match a common term. They are called parallel, written $\pi_1 \parallel \pi_2$, if $p_{\pi_1} \parallel p_{\pi_2}$. If $P = \{\pi_1, \ldots, \pi_n\}$ is a set of pairwise parallel redex patterns matching a term t, we denote by $t \twoheadrightarrow^{P} t'$ the parallel rewrite step from t to t' by P, i.e., $t \to^{\pi_1} \cdot \ldots \cdot \to^{\pi_n} t'$.

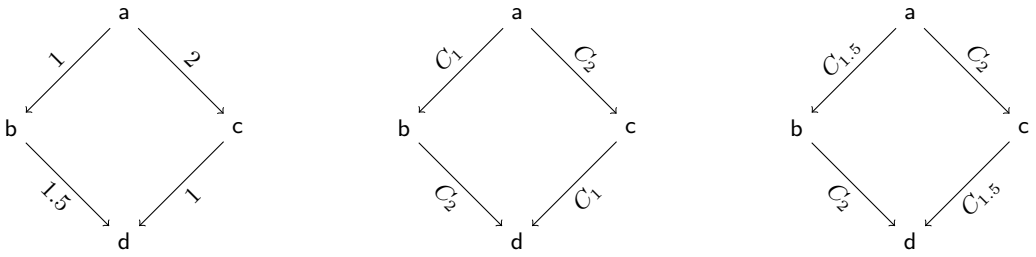
In IsaFoR, an abstract rewrite system (ARS) is a binary relation $\to$ where the domain is left implicit in the type. Let I be an index set. We write $\{\to_\alpha\}_{\alpha\in I}$ to denote the ARS $\to$ where $\to$ is the union of $\to_\alpha$ for all $\alpha \in I$. Let $\{\to_\alpha\}_{\alpha\in I}$ be an ARS and let $>$ and $\geqslant$ be relations on I. Two relations $>$ and $\geqslant$ are called compatible if $\geqslant \cdot > \cdot \geqslant \subseteq >$. Given a relation $\succcurlyeq$ we write $\to_{\curlyvee\alpha_1 \cdots \alpha_n}$ for the union of $\to_\beta$ where $\alpha_i \succcurlyeq \beta$ for some $1 \leqslant i \leqslant n$. Similarly, $\curlyvee S$ is the set of all β such that $\alpha \succcurlyeq \beta$ for some $\alpha \in S$. We call α and β *extended locally decreasing* (for $>$ and $\geqslant$) if ${}_\alpha\!\leftarrow \cdot \to_\beta \subseteq \to_{\curlyvee\alpha}^{*} \cdot \to_{\curlyvee/\beta}^{=} \cdot \to_{\curlyvee\alpha\beta}^{*} \cdot {}_{\curlyvee\alpha\beta}\!\overset{*}{\leftarrow} \cdot {}_{\curlyvee/\alpha}\!\overset{=}{\leftarrow} \cdot {}_{\curlyvee\beta}\!\overset{*}{\leftarrow}$. If there exist a well-founded order $>$ and a preorder $\geqslant$, such that $>$ and $\geqslant$ are compatible, and α and β are extended locally decreasing for all $\alpha, \beta \in I$ then the ARS $\{\to_\alpha\}_{\alpha\in I}$ is *extended locally decreasing* (for $>$ and $\geqslant$). We call an ARS *locally decreasing* (for $>$) if it is extended locally decreasing for $>$ and $=$, where the latter is the identity relation. In the sequel, we often refer to extended locally decreasing as well as to locally decreasing just by decreasing, whenever the context clarifies which concept is meant or the exact meaning is irrelevant.

3 Abstract Rewriting

This section is concerned with the formalization of the following result from [7, Theorem 2]:

▶ **Lemma 2.** *Every extended locally decreasing ARS is confluent.* ◀

[3] We do not require the common *variable conditions*, i.e., the restriction that l is not a variable and all variables in r are contained in l.



(a) Extended locally decreasing **(b)** Locally decreasing **(c)** Locally decreasing

Figure 1 (Extended) locally decreasing peaks.

The results for decreasing diagrams formalized in [28] differ from the above lemma for two reasons. Firstly, [28] establishes results for local decreasingness instead of extended local decreasingness. Secondly, in contrast to the formulation of the lemma above it does not represent the ARS as a family of rewrite relations (i.e., $\{\to_\alpha\}_{\alpha\in I}$) but considers a single labeled relation where a triple (a, α, b) expresses that $(a, b) \in \to_\alpha$.

Given an ARS that is extended locally decreasing for $>$ and $\geqslant$, the proof in [7] constructs a single order $\succ$ on sets of labels and establishes local decreasingness of the ARS for $\succ$. Our formalization goes along the lines with the proposed proof (see below). It turned out that the representation of the ARS as a family of relations is essential to follow the proof in [7]. Hence establishing equivalence of a single labeled ARS with a family of rewrite relations is needed to employ the formalization of [28] in the proof of Lemma 2. This equivalence looks trivial at first sight, but as each representation comes with a different formalization of rewrite steps, also related concepts such as local peaks, joining sequences, and local decreasingness, must be mapped. We refer the interested reader to the formalization and do not present the technical details here.

The remainder of this section sketches the formalization of the next lemma (following [7]).

▶ **Lemma 3.** *Every extended locally decreasing ARS is locally decreasing.*

To prepare for its proof we consider sets of labels.

▶ **Definition 4.** Let C_α denote the set $\{\alpha' \mid \alpha \geqslant \alpha' \text{ and } \alpha \not> \alpha'\}$ and let $\mathcal{C}$ be the set of all C_α. For $C, D \in \mathcal{C}$ let $C \succ D$ if there exist α and β with $C = C_\alpha$, $D = C_\beta$, and $\alpha > \beta$. By $\to_C$, we denote the union of $\to_\alpha$ for all $\alpha \in C$.

The idea is to establish $\{\to_\alpha\}_{\alpha\in I} = \{\to_C\}_{C\in\mathcal{C}}$ and conclude local decreasingness of the ARS $\{\to_C\}_{C\in\mathcal{C}}$ based on extended local decreasingness of the ARS $\{\to_\alpha\}_{\alpha\in I}$. The next example demonstrates some peculiarities of this approach.

▶ **Example 5.** Consider the ARS $\{\to_\alpha\}_{\alpha\in\{1,1.5,2\}}$ with $\to_1 = \{(\mathsf{a},\mathsf{b}),(\mathsf{c},\mathsf{d})\}$, $\to_{1.5} = \{(\mathsf{b},\mathsf{d})\}$, and $\to_2 = \{(\mathsf{a},\mathsf{c})\}$. This ARS is extended locally decreasing for $>_\mathbb{N}$ and $\geqslant_\mathbb{Q}$, as depicted in Figure 1(a). We have $\mathcal{C} = \{C_2, C_{1.5}, C_1\}$ with $C_2 = \{2, 1.5\}$, $C_{1.5} = \{1.5, 1\}$, and $C_1 = \{1\}$. E.g. $1.5 \in C_2$ since $2 \geqslant_\mathbb{Q} 1.5$ but $2 \not>_\mathbb{N} 1.5$. Consequently, $\to_{C_2} = \{(\mathsf{a},\mathsf{c}),(\mathsf{b},\mathsf{d})\}$, $\to_{C_{1.5}} = \{(\mathsf{b},\mathsf{d}),(\mathsf{a},\mathsf{b}),(\mathsf{c},\mathsf{d})\}$, and $\to_{C_1} = \{(\mathsf{a},\mathsf{b}),(\mathsf{c},\mathsf{d})\}$. To establish local decreasingness of the related ARS $\{\to_C\}_{C\in\mathcal{C}}$ the peak $\mathsf{b} \;{}_{C_1}\!\!\leftarrow \mathsf{a} \to_{C_2} \mathsf{c}$ (emerging from $\mathsf{b} \;{}_1\!\!\leftarrow \mathsf{a} \to_2 \mathsf{c}$) must be considered, which can be closed in a locally decreasing fashion via $\mathsf{b} \to_{C_2} \mathsf{d} \;{}_{C_1}\!\!\leftarrow \mathsf{c}$ (based on $\mathsf{b} \to_{1.5} \mathsf{d} \;{}_1\!\!\leftarrow \mathsf{c}$), as in Figure 1(b). However, the construction also admits the peak $\mathsf{b} \;{}_{C_{1.5}}\!\!\leftarrow \mathsf{a} \to_{C_2} \mathsf{c}$, for which there is no peak $\mathsf{b} \;{}_{1.5}\!\!\leftarrow \mathsf{a} \to_2 \mathsf{c}$ in the original ARS, as it does not contain the step $\mathsf{b} \;{}_{1.5}\!\!\leftarrow \mathsf{a}$. Still, this peak can be closed locally decreasing, cf. Figure 1(c).

The following properties are crucial:

▶ **Lemma 6.** *Let $>$ be a well-founded order and $\geqslant$ a preorder compatible with $>$.*

1. *Then $\succ$ is a well-founded order.*
2. *If $\gamma \geqslant \gamma'$, $\delta \geqslant \delta'$, and $x \to^*_{\curlyvee\gamma'\delta'} y$ then $x \to^*_{\curlyvee\gamma\delta} y$.*
3. *If $\gamma \geqslant \gamma'$ and $x \to^=_{\curlywedge\gamma'} y$ then $x \to^=_{\curlywedge\gamma} y$.*
4. *If $x \to^*_{\curlyvee\gamma\delta} y$ then $x \to^*_{\curlyvee C_\gamma C_\delta} y$.*
5. *If $x \to^=_{\curlywedge\gamma} y$ then $x \to^=_{C_\gamma} y$ or $x \to_{\curlyvee C_\gamma} y$.*

Proof. Items (1–3) follow from the properties of the orders. Items (4) and (5) are established as in [7]. ◀

Item (1) of Lemma 6, i.e., well-foundedness of $\succ$ is not proved explicitly in [7]. Moreover items (2) and (3) are missing in [7]. Their need becomes apparent in the following proof. In [8] (the journal version of [7]) extended local decreasingness is avoided by employing the *predecessor labeling*. Then a rewrite step comes with a set of labels, which is typically not computable and hence inappropriate for certification.

Proof of Lemma 3. We assume the ARS $\{\to_\alpha\}_{\alpha\in I}$ is extended locally decreasing for $>$ and $\geqslant$ and establish local decreasingness of the ARS $\{\to_C\}_{C\in\mathcal{C}}$ for $\succ$ by showing

$$\xleftarrow[C]{} \cdot \xrightarrow[D]{} \subseteq \xrightarrow[\curlyvee C]{*} \cdot \xrightarrow[D]{=} \cdot \xrightarrow[\curlyvee CD]{*} \cdot \xleftarrow[\curlyvee CD]{*} \cdot \xleftarrow[C]{=} \cdot \xleftarrow[\curlyvee D]{*} \tag{1}$$

for $C, D \in \mathcal{C}$.[4] By definition of C and D, there exist α and β with $C = C_\alpha$ and $D = C_\beta$, i.e., ${}_C\!\leftarrow \cdot \to_D = {}_{C_\alpha}\!\leftarrow \cdot \to_{C_\beta} = \bigcup_{\alpha'\in C_\alpha, \beta'\in C_\beta} {}_{\alpha'}\!\leftarrow \cdot \to_{\beta'}$. We note that from $y \,{}_{C_\alpha}\!\leftarrow x$ in general we may not infer $y \,{}_\alpha\!\leftarrow x$, but rather $y \,{}_{\alpha'}\!\leftarrow x$ for some $\alpha' \in C_\alpha$ (cf. Example 5). Similarly $x \to_{C_\beta} z$ implies $x \to_{\beta'} z$ for some $\beta' \in C_\beta$. Consequently, the extended local decreasingness assumption cannot be applied to α and β (as conveyed in [7]) but must be applied to α' and β' (as sketched in Example 5), i.e.,

$$\xleftarrow[\alpha']{} \cdot \xrightarrow[\beta']{} \subseteq \xrightarrow[\curlyvee\alpha']{*} \cdot \xrightarrow[\curlywedge\beta']{=} \cdot \xrightarrow[\curlyvee\alpha'\beta']{*} \cdot \xleftarrow[\curlyvee\alpha'\beta']{*} \cdot \xleftarrow[\curlywedge\alpha']{=} \cdot \xleftarrow[\curlyvee\beta']{*}$$

Then we establish

$$\xrightarrow[\curlyvee\alpha']{*} \cdot \xrightarrow[\curlywedge\beta']{=} \cdot \xrightarrow[\curlyvee\alpha'\beta']{*} \cdot \xleftarrow[\curlyvee\alpha'\beta']{*} \cdot \xleftarrow[\curlywedge\alpha']{=} \cdot \xleftarrow[\curlyvee\beta']{*} \subseteq \xrightarrow[\curlyvee\alpha]{*} \cdot \xrightarrow[\curlywedge\beta]{=} \cdot \xrightarrow[\curlyvee\alpha\beta]{*} \cdot \xleftarrow[\curlyvee\alpha\beta]{*} \cdot \xleftarrow[\curlywedge\alpha]{=} \cdot \xleftarrow[\curlyvee\beta]{*}$$

using Lemma 6(2-3), from which the desired

$$\xleftarrow[C_\alpha]{} \cdot \xrightarrow[C_\beta]{} \subseteq \xrightarrow[\curlyvee C_\alpha]{*} \cdot \xrightarrow[C_\beta]{=} \cdot \xrightarrow[\curlyvee C_\alpha C_\beta]{*} \cdot \xleftarrow[\curlyvee C_\alpha C_\beta]{*} \cdot \xleftarrow[C_\alpha]{=} \cdot \xleftarrow[\curlyvee C_\beta]{*}$$

is obtained using Lemma 6(4–5). Depending on the case of Lemma 6(5) that applies, the reflexive step either stays, if e.g. $\to^=_{\curlywedge\beta}$ becomes $\to^=_{C_\beta}$, or is merged with the subsequent sequence having smaller labels, if e.g. $\to^=_{\curlywedge\beta}$ becomes $\to_{\curlyvee C_\beta}$, establishing the property (1). The proof concludes by the equivalence $\{\to_\alpha\}_{\alpha\in I} = \{\to_C\}_{C\in\mathcal{C}}$, as in [7]. ◀

[4] In [7] the (stronger) property ${}_C\!\leftarrow \cdot \to_D \subseteq \to^*_{\curlyvee C} \cdot \to^=_{\curlyvee D} \cdot \to^*_{\curlyvee CD} \cdot {}_{\curlyvee CD}\!\overset{*}{\leftarrow} \cdot {}_{\curlyvee C}\!\overset{=}{\leftarrow} \cdot {}_{\curlyvee D}\!\overset{*}{\leftarrow}$ is claimed, but as this is obviously impossible we anticipate a typo there.

4 Term Rewriting

This section builds upon the result for ARSs from the previous section to prepare for confluence criteria for TRSs, such as Theorem 1. To support confluence results besides Theorem 1, in the formalization we did not follow the easiest way, i.e., suit the definitions and lemmas directly towards Theorem 1. Rather, we adopted the approach from [31], where all results are established via *labeling functions* (satisfying some abstract properties). Apart from avoiding a monolithic proof, this has the advantage that similar proofs need not be repeated for different labeling functions but it suffices to establish that the concrete labeling functions satisfy some abstract conditions. Then decreasingness is established in three steps. The first step comprises joinability results for local peaks (Section 4.1). The second step (Section 4.2) formulates abstract conditions with the help of *labeling functions* that admit a finite characterization of decreasingness of local peaks. Finally, based on the previous two steps, the third step (Section 5) then obtains confluence results by instantiating the abstract labeling functions with concrete ones, e.g. the rule labeling. So only the third step needs to be adapted when formalizing new labeling functions, as steps one and two are unaffected.

4.1 Local Peaks

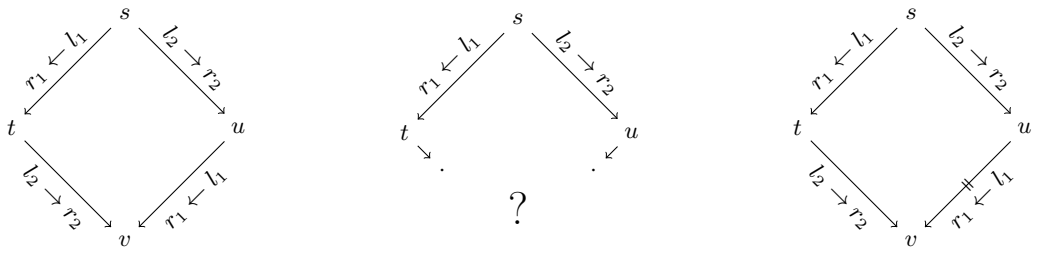
As IsaFoR already supported Knuth-Bendix' criterion (see [21]), it contained results for joinability of local peaks and the critical pair theorem (the terms obtained by a local peak in a left-linear TRS are joinable or an instance of a critical pair). However, large parts of the existing formalization could not be reused directly as the established results lacked information required for ensuring decreasingness. For instance, to obtain decreasingness for the rule labeling (cf. Section 5) in case of a variable peak, the rewrite rules employed in the joining sequences are crucial, but the existing formalization only states that such a local peak is joinable. On the other hand, the existing notion of critical pairs from IsaFoR could be reused as the foundation for critical peaks. Since the computation of critical pairs requires a formalized unification algorithm, extending IsaFoR admitted focusing on the tasks related to decreasingness.

Local peaks can be characterized based on the positions of the diverging rewrite steps. Either the positions are parallel, called a parallel peak, or one position is above the other. In the latter situation we further distinguish whether the lower position is at a function position, called a function peak, or at/below a variable position of the other rule's left-hand side, called a variable peak. More precisely, for a local peak

$$t = s[r_1\sigma_1]_p \leftarrow s[l_1\sigma_1]_p = s = s[l_2\sigma_2]_q \rightarrow s[r_2\sigma_2]_q = u \tag{2}$$

there are three possibilities (modulo symmetry):
(a) $p \parallel q$ (parallel peak),
(b) $q \leqslant p$ and $p\backslash q \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l_2)$ (function peak),
(c) $q \leqslant p$ and $p\backslash q \notin \mathcal{P}\mathsf{os}_{\mathcal{F}}(l_2)$ (variable peak).

For the situation of a left-linear TRS these cases are visualized in Figure 2. It is easy to characterize parallel, function, and variable peaks in Isabelle (cf. Listing 1) but it requires tedious notation. The information of a rewrite step $s \to_{\mathcal{R}}^{p,l\to r,\sigma} t$ is represented in IsaFoR as `(s,t)` ∈ `rstep_r_p_s` $\mathcal{R}$ `(l,r)` `p` σ. As the definition of function and variable peaks is asymmetric the five cases of local peaks can be reduced to the above three by mirroring those peaks. Then local peaks can be characterized as in Listing 2. Next we elaborate on the three cases.



(a) Parallel peak **(b)** Function peak **(c)** Variable peak

Figure 2 Three kinds of local peaks.

```
definition local_peaks where "local_peaks R =
 {((s,rl1,p,σ1,t),(s,rl2,q,σ2,u)) | s t u rl1 rl2 p q σ1 σ2.
 ((s,t) ∈ rstep_r_p_s R rl1 p σ1 ∧ (s,u) ∈ rstep_r_p_s R rl2 q σ2)}"

definition parallel_peak where "parallel_peak R pk = (
 pk ∈ local_peaks R ∧ (let ((s,rl1,p,σ1,t),(s,rl2,q,σ2,u)) = pk in
 p ⊥ q))"

definition function_peak where "function_peak R pk = (
 pk ∈ local_peaks R ∧ (let ((s,rl1,p,σ1,t),(s,rl2,q,σ2,u)) = pk in
 ∃r.((p<#>r = q) ∧ r ∈ poss (fst rl1) ∧ is_Fun ((fst rl1) |_ r))))"

definition variable_peak where "variable_peak R pk = (
 pk ∈ local_peaks R ∧ (let ((s,rl1,p,σ1,t),(s,rl2,q,σ2,u)) = pk in
 ∃r.((p<#>r = q) ∧ ¬(r ∈ poss (fst rl1) ∧ is_Fun ((fst rl1) |_ r)))))"
```

Listing 1 Characterization of local peaks.

```
lemma local_peaks_cases:
  assumes "pk ∈ local_peaks R"
  shows "parallel_peak R pk ∨ variable_peak R pk ∨ function_peak R pk
   ∨ variable_peak R (snd pk, fst pk) ∨ function_peak R (snd pk, fst pk)"
```

Listing 2 Cases of local peaks.

Case 1: Parallel Peaks

Figure 2(a) shows the shape of a local peak where the steps take place at parallel positions. For a peak $t \ {}^{\pi_1}\!\leftarrow s \rightarrow^{\pi_2} u$ with $\pi_1 \parallel \pi_2$ we established that $t \rightarrow^{\pi_2} v \ {}^{\pi_1}\!\leftarrow u$, i.e., the steps drawn at opposing sides in the diagram are corresponding, that is, they apply the same rule/substitution at the same position. The proof is straightforward and based on a decomposition of the terms into a context and the redex.

Case 2: Function Peaks

In general joining function peaks may involve rules not present in the divergence (as indicated by the question mark in Figure 2(b)). To reduce the duty of joining (infinitely many) function

peaks to joining the (in case of a finite TRS finitely many) critical peaks, we established that every function peak is an instance of a critical peak.

▶ **Lemma 7.** *Let* $t \;{}^{p,l_1\to r_1,\sigma_1}\!\!\leftarrow\; s \;\to^{q,l_2\to r_2,\sigma_2}\; u$ *with* $qq' = p$, *and* $q' \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(l_2)$. *Then there are a context* C, *a substitution* τ, *and a critical peak* $l_2\mu[r_1\mu]_{q'} \leftarrow l_2\mu \to r_2\mu$ *such that* $s = C[l_2\mu\tau]$, $t = C[(l_2\mu[r_1\mu]_{q'})\tau]$, *and* $u = C[r_2\mu\tau]$. ◀

We remark that this fact was already present (multiple times) in IsaFoR, but concealed in larger proofs, e.g. the formalization of orthogonality [12], and never stated explicitly.

As IsaFoR does not enforce that the variables of a rewrite rule's right-hand side are contained in its left-hand side, such rules are just also included in the critical peak computation.

Case 3: Variable Peaks

Variable overlaps (Figure 2(c)) can again be joined by the rules involved in the diverging step.[5] We only consider the case if $l_2 \to r_2$ is left-linear, as our main result assumes left-linearity. More precisely, if q' is the unique position in $\mathcal{P}\mathsf{os}_{\mathcal{V}}(l_2)$ such that $qq' \leqslant p$, $x = l_2|_{q'}$, and $|r_2|_x = n$ then we have $t \to_{l_2\to r_2} v$, which is similar to the case for parallel peaks, as the redex $l_2\sigma$ becomes $l_2\tau$ but is not destroyed, and $u \to^n_{l_1\to r_1} v$. To obtain this result we reason via parallel rewriting. The notion of parallel rewriting already supported by IsaFoR (employed to prove that orthogonal systems are confluent) does not keep track of e.g. the applied rules. Thus we augmented IsaFoR by a new version of parallel steps, which record the information (position, rewrite rule, substitution) of each rewrite step, i.e., the rewrite relation is decorated with the contracted redex patterns:

$$\frac{}{x \overset{\varnothing}{\twoheadrightarrow\!\!\!\!|\;\;} x} \qquad \frac{l \to r \in \mathcal{R}}{l\sigma \overset{\{\langle\epsilon, l\to r,\sigma\rangle\}}{\twoheadrightarrow\!\!\!\!|\;\;} r\sigma} \qquad \frac{s_1 \overset{P_1}{\twoheadrightarrow\!\!\!\!|\;\;} t_1 \quad \cdots \quad s_n \overset{P_n}{\twoheadrightarrow\!\!\!\!|\;\;} t_n}{f(s_1,\ldots,s_n) \overset{(1P_1)\cup\cdots\cup(nP_n)}{\twoheadrightarrow\!\!\!\!|\;\;} f(t_1,\ldots,t_n)}$$

Here for a set of redex patterns $P = \{\pi_1,\ldots,\pi_m\}$ by iP we denote $\{i\pi_1,\ldots,i\pi_m\}$ with $i\pi = \langle ip, l \to r, \sigma\rangle$ for $\pi = \langle p, l \to r, \sigma\rangle$. To use this parallel rewrite relation for closing variable peaks we established the following auxiliary results.

▶ **Lemma 8.** *The following properties of the parallel rewrite relation hold:*

1. *For all* s *we have* $s \twoheadrightarrow\!\!\!\!|^{\varnothing} s$.
2. *If* $s \twoheadrightarrow\!\!\!\!|^{\varnothing} t$ *then* $s = t$.
3. *If* $s \twoheadrightarrow\!\!\!\!|^{P} t$ *and* $q \in \mathcal{P}\mathsf{os}(u)$ *then* $u[s]_q \twoheadrightarrow\!\!\!\!|^{qP} u[t]_q$.
4. *We have* $s \to^{\pi} t$ *if and only if* $s \twoheadrightarrow\!\!\!\!|^{\{\pi\}} t$.
5. *If* $\sigma(x) \to^{\pi} \tau(x)$ *and* $\sigma(y) = \tau(y)$ *for all* $y \in \mathcal{V}$ *with* $y \neq x$ *then* $t\sigma \twoheadrightarrow\!\!\!\!|^{P} t\tau$ *with* $l_{\pi'} \to r_{\pi'} = l_\pi \to r_\pi$ *for all* $\pi' \in P$.
6. *If* $s \twoheadrightarrow\!\!\!\!|^{\{\pi\}\cup P} t$ *then there is a* u *with* $s \twoheadrightarrow\!\!\!\!|^{\{\pi\}} u \twoheadrightarrow\!\!\!\!|^{P} t$.
7. *If* $s \twoheadrightarrow\!\!\!\!|^{\{\pi_1,\ldots,\pi_n\}} t$ *then* $s \to^{\pi_1} \cdots \to^{\pi_n} t$.

Proof. In principle the results follow from the definitions using straightforward induction proofs. However, the additional bookkeeping, required to correctly propagate the information attached to the rewrite relation, makes them considerably more involved than for the existing, agnostic notion of parallel rewriting. ◀

Now for reasoning about variable peaks as above we decompose $u = s[r_2\sigma]_q$ and $v = s[r_2\tau]_q$ where $\sigma(y) = \tau(y)$ for all $y \in \mathcal{V} \setminus \{x\}$ and $\sigma(x) \to_{p\setminus qq', l_1\to r_1} \tau(x)$. From the latter by item (5)

[5] This includes rules having a variable as left-hand side.

```
inductive_set seq for R where
 "(s,[]) ∈ seq R" |
 "(s,t) ∈ rstep_r_p_s R rl p σ
   ⟹ (t,ts) ∈ seq R ⟹ (s,(s,rl,p,σ,t)#ts) ∈ seq R"
```

Listing 3 Rewrite sequences.

we obtain $r_2\sigma \twoheadrightarrow^P r_2\tau$, where all redex patterns in P use $l_1 \to r_1$. Then by item (3) we get $s[r_2\sigma]_q \twoheadrightarrow^{qP} s[r_2\tau]_q$ and finally $s[r_2\sigma]_q \to^n_{l_1\to r_1} s[r_2\tau]_q$ with $n = |qP| = |P|$ by item (7).

4.2 Local Decreasingness

The aim of this section is a confluence result (cf. Corollary 14) based on decreasingness of the critical peaks. Abstract conditions, via the key notion of a labeling, will ensure that parallel peaks and variable peaks are decreasing. Furthermore these conditions imply that decreasingness of the critical peaks implies decreasingness of the function peaks.

For establishing (extended) local decreasingness, a label must be attached to rewrite steps. To facilitate checking, the formalization makes the rewrite sequences (cf. Listing 3) explicit, i.e., they involve the intermediate terms, applied rules, etc. based on `rstep_r_p_s`. Furthermore, labels are computed by a *labeling (function)*, having (local) information about the rewrite step (such as source and target term, applied rewrite rule, position, and substitution) it is expected to label. For reasons of readability in this presentation we employ the mathematical notation (e.g., $\to^*$, etc.) with all information implicit but remark that the formalization works on rewrite sequences with explicit information (as in Listing 3).

▶ **Definition 9.** A *labeling* is a function ℓ from rewrite steps to a set of labels such that for all contexts C and substitutions σ the following properties are satisfied:

- If $\ell(s \to^{\pi_1} t) > \ell(u \to^{\pi_2} v)$ then $\ell(C[s\sigma] \to^{C[\pi_1\sigma]} C[t\sigma]) > \ell(C[u\sigma] \to^{C[\pi_2\sigma]} C[v\sigma])$
- If $\ell(s \to^{\pi_1} t) \geqslant \ell(u \to^{\pi_2} v)$ then $\ell(C[s\sigma] \to^{C[\pi_1\sigma]} C[t\sigma]) \geqslant \ell(C[u\sigma] \to^{C[\pi_2\sigma]} C[v\sigma])$

Here $C[\pi\sigma]$ denotes $\langle qp, l \to r, \tau\sigma\rangle$ for $\pi = \langle p, l \to r, \tau\rangle$ and $C|_q = \Box$.

In presence of a labeling, rewrite sequences can be labeled at any time. This avoids lifting many notions (such as rewrite steps, local peaks, rewrite sequences, etc.) and results from rewriting to labeled rewriting.

Following [28], we separate (local) diagrams (where rewriting is involved) from decreasingness (where only the labels are involved). In the next definition a labeling is extended to peaks and rewrite sequences via the equations: $\ell(t \,{}^{\pi}\!\leftarrow s) = \ell(s \to^{\pi} t)$, $\ell(t \to^0 t) = \varnothing$, and $\ell(s \to^{\pi} t \to^* u) = \{\ell(s \to^{\pi} t)\} \cup \ell(t \to^* u)$.

▶ **Definition 10.** A local peak $t \,{}^{\pi_1}\!\leftarrow s \to^{\pi_2} u$ is *extended locally decreasing* (for ℓ) if it can be completed into a local diagram $t \to^* t' \to^= t'' \to^* v \,{}^*\!\leftarrow u'' \,{}^=\!\leftarrow u' \,{}^*\!\leftarrow u$ such that its labels are extended locally decreasing, i.e.,
$\ell(t \to^* t') \subseteq \vee\ell(t \,{}^{\pi_1}\!\leftarrow s)$, $\ell(t' \to^= t'') \subseteq \vee\!/\ell(s \to^{\pi_2} u)$, $\ell(t'' \to^* v) \subseteq \vee\ell(t \,{}^{\pi_1}\!\leftarrow s \to^{\pi_2} u)$ and
$\ell(u' \,{}^*\!\leftarrow u) \subseteq \vee\ell(s \to^{\pi_2} u)$, $\ell(u'' \,{}^=\!\leftarrow u') \subseteq \vee\!/\ell(t \,{}^{\pi_1}\!\leftarrow s)$, $\ell(v \,{}^*\!\leftarrow u'') \subseteq \vee\ell(t \,{}^{\pi_1}\!\leftarrow s \to^{\pi_2} u)$.

The corresponding predicate in IsaFoR is given in Listing 4 where extended local decreasingness (`eld`) of a local peak `pk` is expressed via the existence of rewrite sequences `jl` and `jr` that join the divergence caused by the local peak `pk` in the shape of a local diagram (`ld_trs`) and the labels of the underlying rewrite sequences are extended locally decreasing (`eld_seq`). Here `r` is the pair of relations $(>, \geqslant)$.

```
definition eld where "eld R ℓ r pk =
 (∃ jl jr. (ld_trs R pk jl jr ∧ eld_seq ℓ r pk jl jr))"
```

Listing 4 Extended local decreasingness.

Then a function peak is extended locally decreasing if the critical peaks are.

▶ **Lemma 11.** *Let ℓ be a labeling and let all critical peaks of a TRS $\mathcal{R}$ be extended locally decreasing for ℓ. Then every function peak of $\mathcal{R}$ is extended locally decreasing for ℓ.*

Proof. As every function peak is an instance of a critical peak (see Lemma 7), the result follows from ℓ being a labeling (Definition 9). ◀

The notion of compatibility (between a TRS and a labeling) admits a finite characterization of extended local decreasingness.

▶ **Definition 12.** Let ℓ be a labeling. We call ℓ *compatible* with a TRS $\mathcal{R}$ if all parallel peaks and all variable peaks of $\mathcal{R}$ are extended locally decreasing for ℓ.

The key lemma then establishes that if ℓ is compatible with a TRS, then all local peaks are extended locally decreasing.

▶ **Lemma 13.** *Let ℓ be a labeling which is compatible with a TRS $\mathcal{R}$. If the critical peaks of $\mathcal{R}$ are extended locally decreasing for ℓ, then all local peaks of $\mathcal{R}$ are extended locally decreasing for ℓ.*

Proof. The cases of variable and parallel peaks are taken care of by compatibility. The case of function peaks follows from the assumption in connection with Lemma 11. The symmetric cases for function and variable peaks are resolved by mirroring the local diagrams. ◀

Representing a TRS $\mathcal{R}$ over the signature $\mathcal{F}$ and variables $\mathcal{V}$ as the ARS over objects $\mathcal{T}(\mathcal{F},\mathcal{V})$ and relations $\bigcup_\alpha\{(s,t) \mid s \to^\pi t \text{ and } \ell(s \to^\pi t) = \alpha\}$, Lemma 2 immediately applies to TRSs. To this end extended local decreasingness formulated via explicit rewrite sequences (with labeling functions) has to be mapped to extended local decreasingness on families of (abstract rewrite) relations; we omit the technical details here.

Finally, we obtain the following result.

▶ **Corollary 14.** *Let ℓ be a labeling compatible with a TRS $\mathcal{R}$. If the critical peaks of $\mathcal{R}$ are extended locally decreasing for ℓ then $\mathcal{R}$ is confluent.* ◀

Concrete confluence criteria are then obtained as instances of the above result. In the case of Theorem 1 by showing that the relative termination assumption in combination with the rule labeling implies the desired preconditions.

5 Applications

In this section we instantiate Corollary 14 to obtain concrete confluence results. Afterwards we discuss the design of the certificates, checkable by CeTA.

5.1 Rule Labeling

The rule labeling [16] is parametrized by an index mapping $i\colon \mathcal{R} \to \mathbb{N}$, which associates to every rewrite rule a natural number.

▶ **Definition 15.** The function $\ell^i(s \to^\pi t) = i(l_\pi \to r_\pi)$ is called *rule labeling*. Labels due to the rule labeling are compared by $>_\mathbb{N}$ and $\geqslant_\mathbb{N}$.

The rule labeling admits a confluence criterion based on the results established so far.

▶ **Lemma 16.**

1. *The rule labeling is a labeling.*
2. *Parallel peaks are extended locally decreasing for the rule labeling.*
3. *Variable peaks of a linear TRS are extended locally decreasing for the rule labeling.*
4. *The rule labeling is compatible with a linear TRS.*
5. *A linear TRS is confluent if its critical peaks are extended locally decreasing for the rule labeling.*

Proof. Item (1) follows from Definition 9. For (2) and (3) we employ the analysis of parallel and variable peaks from Section 4.1, respectively. Item (4) is then a consequence of (2) and (3). Finally, (5) amounts to an application of Corollary 14. ◀

Eventually, we remark that for the rule labeling extended local decreasingness implies local decreasingness, as $\geqslant_\mathbb{N}$ is the reflexive closure of $>_\mathbb{N}$.

5.2 Relative Termination

That a locally confluent terminating left-linear TRS is confluent can be established in the flavor of Lemma 16. The restriction to left-linearity arises from the lack of considering non-left-linear variable peaks in Section 4.1. As the analysis of such a peak would not give further insights we pursue another aim in this section, i.e., the mechanized proof of Theorem 1.

It is well known that the rule labeling ℓ^i is in general not compatible with left-linear TRSs, cf. [8]. Thus, to obtain extended local decreasingness for variable peaks the additional relative termination assumption is exploited. To this end we use the *source labeling*, which labels each rewrite step by its source, i.e., $\ell^{\mathsf{src}}(s \to^\pi t) = s$. Here, labels due to the source labeling are compared by the orders $\to^+_{\mathcal{R}_\mathsf{d}/\mathcal{R}_\mathsf{nd}}$ and $\to^*_\mathcal{R}$. The relative termination assumption of Theorem 1 makes all variable peaks of a left-linear TRS extended locally decreasing for the source labeling.

Following [31], the aim is to establish that the lexicographic combination $\ell^{\mathsf{src}} \times \ell^i$ is compatible with a left-linear TRS. To employ the rule labeling we have to introduce a weaker version of compatibility.

▶ **Definition 17.** A diagram of the shape $t \;{}_\alpha\!\leftarrow s \to^{l_2 \to r_2}_\beta u$, $t \to_{\diagdown\!/\beta} v \;{}^n_{\diagdown\!/\alpha}\!\leftarrow u$ is called *weakly extended locally decreasing* if $n \leqslant 1$ whenever r_2 is linear. We call a labeling ℓ *weakly compatible* with a TRS $\mathcal{R}$ if parallel and variable peaks are weakly extended locally decreasing for ℓ.

While weak extended local decreasingness could also be defined in the spirit of extended local decreasingness (with a more complicated join sequence) the chosen formulation eases the definition and simplifies proofs.

Based on the peak analysis of Section 4.1 the following results are established (Such properties must be proved for each labeling function.):

▶ **Lemma 18.** *Let $\mathcal{R}$ be a left-linear TRS.*
1. *Parallel peaks are weakly extended locally decreasing for the rule labeling.*
2. *Variable peaks of $\mathcal{R}$ are weakly extended locally decreasing for the rule labeling.*
3. *The rule labeling is weakly compatible with $\mathcal{R}$.* ◀

Similar results are established for the source labeling.

▶ **Lemma 19.** *Let $\mathcal{R}$ be a left-linear TRS whose duplicating rules terminate relative to the other rules.*
1. *The source labeling is a labeling.*
2. *Parallel peaks are extended locally decreasing for the source labeling.*
3. *Variable peaks of $\mathcal{R}$ are extended locally decreasing for the source labeling.*
4. *The source labeling is compatible with $\mathcal{R}$.* ◀

Using this lemma, we proved the following results for the lexicographic combination of the source labeling with another labeling.

▶ **Lemma 20.** *Let $\mathcal{R}$ be a left-linear TRS whose duplicating rules terminate relative to the other rules, and ℓ a labeling weakly compatible with $\mathcal{R}$.*
1. *Then $\ell^{\mathsf{src}} \times \ell$ is a labeling.*
2. *Then $\ell^{\mathsf{src}} \times \ell$ is compatible with $\mathcal{R}$.* ◀

For reasons of readability we have left the orders $>$ and $\geqslant$ that are required for (weak) compatibility implicit and just mention that the lexicographic extension (as detailed in [31]) preserves the required properties. Finally, we prove the main result of this paper.

Proof of Theorem 1. From Lemma 18(3) in combination with Lemma 20 we obtain that $\ell^{\mathsf{src}} \times \ell^i$ is a labeling compatible with a left-linear TRS, provided the relative termination assumption is satisfied. By assumption, the critical peaks are (extended locally) decreasing for the rule labeling ℓ^i. As along a rewrite sequence labels with respect to ℓ^{src} never increase, the critical peaks are extended locally decreasing for $\ell^{\mathsf{src}} \times \ell^i$. We conclude the proof by an application of Corollary 14. ◀

Hence, actually a stronger result than Theorem 1 has been mechanized, as $\ell^{\mathsf{src}} \times \ell^i$ might show more critical peaks decreasing than ℓ^i alone.

5.3 Certificates

Next we discuss the design of the certificates for confluence proofs via the rule labeling, i.e., how they are represented in CPF, and the executable checker to verify them. A minimal certificate could just claim that the considered rewrite system can be shown decreasing via the rule labeling. However, this is undecidable, even for locally confluent systems [8]. Hence in the certificate the index function i as well as (candidates for) the joining sequences for each critical pair have to be provided. Note that the labels in the joining sequences are not required for the certificate, since CeTA has to check, i.e., compute them anyway. The same holds for the critical peaks.

As the confluence tools that generate certificates might use different renamings than CeTA when computing critical pairs, the joining sequences given in the certificate are subject to a variable renaming. Thus, after computing all critical peaks, CeTA has to look for joining sequences in the certificate modulo renaming of variables.

Now, to verify whether the sequences of labels obtained from the joining sequences by applying the given index function fulfill the extended local decreasingness condition, we need

method	success	CoCo 2013	CoCo 2014	CeTA 2.19
(weak) orthogonality	4	✓	✓	✓
Knuth-Bendix	26	✓	✓	✓
strong closedness	28	✓	✓	✓
Lemma 16(5)	41	✗	✓	✓
Theorem 1	46	✗	✗	✓
$\sum$		45	56	58

Table 1 Experimental results for 148 TRSs from CoCo 2014.

to provide means to decide the following: given two natural numbers α and β and a sequence σ of natural numbers, is there a split $\sigma = \sigma_1\sigma_2\sigma_3$ such that $\sigma_1 \subseteq \vee\alpha$, $\sigma_2 \subseteq \vee\!\!/\beta$ with length of σ_2 at most one, and $\sigma_3 \subseteq \vee\alpha\beta$? To this end our checker employs a simple, greedy approach. That is, we pick the maximal prefix of σ with labels smaller α as σ_1. If the next label is less or equal to β we take it as σ_2 and otherwise we take the empty sequence for σ_2. Finally, the remainder of the sequence is σ_3. A straightforward case analysis shows that this approach is complete, i.e., otherwise no such split exists.

To certify applications of Theorem 1, additionally the relative termination condition has to be checked. Luckily, CeTA already supports a wide range of relative termination techniques, so that here we just needed to make use of existing machinery.

6 Experiments

For experiments we considered the 148 TRSs selected for CoCo 2014 and used the confluence tool CSI [30] to obtain certificates in CPF for confluence proofs. Note that ACP [2] can also produce certificates in CPF, but at the moment they are a subset of the ones reported by CSI. All generated certificates have been certified by CeTA. Note that CeTA can also certify various methods for non-confluence [12]. The largest certificate (for Cops #60) has 760 KB and lists 182 candidate joins for showing the 34 critical peaks decreasing. The certificate is checked within 1.1 seconds. We remark that no confluence tool besides CSI has solved Cops #60 so far, stressing the importance of a certified proof.

Next we elaborate on the impact of the new contributions. Experimental results for various criteria supported by CeTA are shown in Table 1.[6] The CeTA version from CoCo 2013 incorporated (weak) orthogonality [18], Knuth-Bendix' criterion [11], and strong closedness [9]. Due to the formalization described in this paper now also Theorem 1 is supported (column CeTA 2.19). As already employed for CoCo 2014, we included the data for Theorem 1 restricted to linear TRSs, i.e., Lemma 16(5). On our testbed Theorem 1 can establish confluence of more systems than all earlier methods together (46 vs. 45) and admits about 25% increase in power (58 vs. 45) when used in combination with the other criteria.

7 Conclusion

Finally we discuss related work, comment on the existing formalization of rewriting in IsaFoR, and conclude with a short summary.

[6] Details are available from `http://cl-informatik.uibk.ac.at/experiments/2015/rta3`.

7.1 Related Work

Formalizing confluence criteria has a long history in λ-calculus. Huet [10] proved a stronger variant of the parallel moves lemma in Coq. Isabelle/HOL was used in [13] to prove the Church-Rosser property of β, η, and $\beta\eta$. For β-reduction the standard Tait/Martin-Löf proof as well as Takahashi's proof [24] were formalized. The first mechanically verified proof of the Church-Rosser property of β-reduction was done using the Boyer-Moore theorem prover [20]. The formalization in Twelf [17] was used to formalize the confluence proof of a specific higher-order rewrite system in [23].

Next we discuss related work for term rewriting. Newman's lemma (for abstract rewrite systems) and Knuth and Bendix' critical pair theorem (for first-order rewrite systems) have been proved in [19] using ACL. An alternative proof of the latter in PVS, following the higher-order structure of Huet's proof, is presented in [6]. PVS is also used in the formalization of the lemmas of Newman and Yokouchi in [5]. Knuth and Bendix' criterion has also been formalized in Coq [4] and Isabelle/HOL [26]. The strong closedness condition of Huet [9] has been formalized by the first author in Isabelle [12] where reasoning similar to the one in Section 4.1 is used to (strongly) close variable and parallel peaks. However, for strong closedness it suffices to construct a common reduct while for our setting every rewrite step has to be made explicit in order to compute the labels and show local decreasingness.

7.2 Assessment

Next we discuss the usefulness of existing formalizations for this work. The existing machinery of IsaFoR admitted invaluable support. We regard our efforts to establish an annotated version of parallel rewriting not as a shortcoming of IsaFoR, but as a useful extension to it. On the contrary, we could employ many results from IsaFoR without further ado, e.g., completeness of the unification algorithm (to compute critical peaks), plain rewriting (to connect parallel steps with single steps), and the support for relative termination. Although [28] does not provide the main result for decreasing diagrams of abstract rewrite systems that are represented via families (as needed for Lemma 2), the amount of work to use this result has been modest, justifying the usability of [28]. That Lemma 7 occurred several times in IsaFoR can be explained as follows. Note that also in textbook proofs (e.g. [3]) this result is not made explicit but established in the scope of a larger proof, probably due to its nasty formulation. Still, in later proofs the result is used as if it would have been established explicitly. In IsaFoR these proofs have been duplicated, but as formalization papers typically come with code refactoring these deficiencies have been fixed. Note that the duplicated proofs have actually never been published.

Next, changes to the setting in [31] are addressed. The concepts of an L-labeling and an LL-labeling from [31] have been unified to the notion of a labeling *compatible* with a TRS while weak-LL-labelings are represented via *weakly compatible* labelings here. This admits the formulation of the abstract conditions such that a labeling ensures confluence (cf. Corollary 14) independent from the TRS being linear or left-linear. We anticipate that the key result for closing variable peaks for the left-linear case (cf. Section 4.1) does not rely on the annotated version of parallel rewriting, but as [31] also supports labelings based on parallel rewriting the developed machinery might be useful for future certification efforts. The formalization described in this paper covers a significant amount of the results presented in [31]. As explained, additional concepts (e.g., the annotated version of parallel rewriting) were formalized to already prepare towards the remaining criteria. However, for some results that are not covered yet (e.g. persistency), we anticipate that already formalizing the preliminaries requires significant effort.

7.3 Summary

In this paper we presented the formalization of a result establishing confluence of left-linear term rewrite systems based on relative termination and the rule labeling. While our formalization admits stronger results (in order to prepare for further results from [31]), we targeted Theorem 1, whose statement (in contrast to its proof) does not require the complex interplay of relative termination and the rule labeling, admitting the use of external termination provers. Our formalization also admits the (original) criterion for the rule labeling (cf. Lemma 16(5)). As this criterion applies to linear systems only, the involved analysis of non-right-linear variable peaks is not needed. The same holds for (the interplay with) the relative termination condition and the notion of extended local decreasingness (the rule labeling does not benefit from a preorder). Hence the proof of Theorem 1 is significantly more involved than the one of Lemma 16(5).

Acknowledgments

We thank Bertram Felgenhauer, Christian Sternagel, and René Thiemann for discussion and the reviewers for helpful comments.

References

1 T. Aoto. Automated confluence proof by decreasing diagrams based on rule-labelling. In *Proc. 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics*, pages 7–16, 2010.

2 T. Aoto, J. Yoshida, and Y. Toyama. Proving confluence of term rewriting systems automatically. In *Proc. 20th International Conference on Rewriting Techniques and Applications*, volume 5595 of *Lecture Notes in Computer Science*, pages 93–102, 2009.

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

4 E. Contejean, P. Courtieu, J. Forest, O. Pons, and X. Urbain. Automated certified proofs with CiME3. In *Proc. 22nd International Conference on Rewriting Techniques and Applications*, volume 10 of *Leibniz International Proceedings in Informatics*, pages 21–30, 2011.

5 A.L. Galdino and M. Ayala-Rincón. A formalization of Newman's and Yokouchi's lemmas in a higher-order language. *Journal of Formalized Reasoning*, 1(1):39–50, 2008.

6 A.L. Galdino and M. Ayala-Rincón. A formalization of the Knuth-Bendix(-Huet) critical pair theorem. *Journal of Automated Reasoning*, 45(3):301–325, 2010.

7 N. Hirokawa and A. Middeldorp. Decreasing diagrams and relative termination. In *Proc. 5th International Joint Conference on Automated Reasoning*, volume 6173 of *Lecture Notes in Artificial Intelligence*, pages 487–501, 2010.

8 N. Hirokawa and A. Middeldorp. Decreasing diagrams and relative termination. *Journal of Automated Reasoning*, 47(4):481–501, 2011.

9 G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.

10 G. Huet. Residual theory in lambda-calculus: A formal development. *Journal of Functional Programming*, 4(3):371–394, 1994.

11 D.E. Knuth and P.B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

12 J. Nagele and R. Thiemann. Certification of confluence proofs using CeTA. In *Proc. 3rd International Workshop on Confluence*, pages 19–23, 2014.

13 T. Nipkow. More Church-Rosser proofs. *Journal of Automated Reasoning*, 26(1):51–66, 2001.

14 T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, 2002.

15 V. van Oostrom. Confluence by decreasing diagrams. *Theoretical Computer Science*, 126(2):259–280, 1994.

16 V. van Oostrom. Confluence by decreasing diagrams – converted. In *Proc. 19th International Conference on Rewriting Techniques and Applications*, volume 5117 of *Lecture Notes in Computer Science*, pages 306–320, 2008.

17 F. Pfenning. A proof of the Church-Rosser theorem and its representation in a logical framework. Technical Report CMU-CS-92-186, School of Computer Science, Carnegie Mellon University, 1992.

18 B.K. Rosen. Tree-manipulating systems and Church-Rosser theorems. *Journal of the ACM*, 20(1):160–187, 1973.

19 J.-L. Ruiz-Reina, J.-A. Alonso, M.-J. Hidalgo, and F.-J. Martín-Mateos. Formal proofs about rewriting using ACL2. *Annals of Mathematics and Artificial Intelligence*, 36(3):239–262, 2002.

20 N. Shankar. A mechanical proof of the Church-Rosser theorem. *Journal of the ACM*, 35(3):475–522, 1988.

21 C. Sternagel and R. Thiemann. Formalizing Knuth-Bendix orders and Knuth-Bendix completion. In *Proc. 24th International Conference on Rewriting Techniques and Applications*, volume 21 of *Leibniz International Proceedings in Informatics*, pages 287–302, 2013.

22 C. Sternagel and R. Thiemann. The certification problem format. In *Proc. 11th International Workshop on User Interfaces for Theorem Provers*, volume 167 of *Electronic Proceedings in Theoretical Computer Science*, pages 61–72, 2014.

23 K. Støvring. Extending the extensional lambda calculus with surjective pairing is conservative. *Logical Methods in Computer Science*, 2(2:1):1–14, 2006.

24 M. Takahashi. Parallel reductions in λ-calculus. *Information and Computation*, 118(1):120–127, 1995.

25 Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

26 R. Thiemann. Certification of confluence proofs using CeTA. In *Proc. 1st International Workshop on Confluence*, page 45, 2012.

27 R. Thiemann and C. Sternagel. Certification of termination proofs using CeTA. In *Proc. 22nd International Conference on Theorem Proving in Higher Order Logics*, volume 5674 of *Lecture Notes in Computer Science*, pages 452–468, 2009.

28 H. Zankl. Confluence by decreasing diagrams – formalized. In *Proc. 24th International Conference on Rewriting Techniques and Applications*, volume 21 of *Leibniz International Proceedings in Informatics*, pages 352–367, 2013.

29 H. Zankl. Decreasing diagrams. *Archive of Formal Proofs*, November 2013. Formal proof development, `http://afp.sf.net/entries/Decreasing-Diagrams.shtml`.

30 H. Zankl, B. Felgenhauer, and A. Middeldorp. CSI – A confluence tool. In *Proc. 23rd International Conference on Automated Deduction*, volume 6803 of *Lecture Notes in Artificial Intelligence*, pages 499–505, 2011.

31 H. Zankl, B. Felgenhauer, and A. Middeldorp. Labelings for decreasing diagrams. *Journal of Automated Reasoning*, 54(2):101–133, 2015.