Recording Completion for Finding and Certifying Proofs in Equational Logic*

Thomas Sternagel, René Thiemann, Harald Zankl
Institute of Computer Science
University of Innsbruck, Austria

Christian Sternagel
School of Information Science
JAIST, Japan

1 Introduction

Solving the word problem requires to decide whether an equation $s \approx t$ follows from an equational system (ES) $\mathcal{E}$. By Birkhoff's theorem this is equivalent to the existence of a conversion $s \leftrightarrow_{\mathcal{E}}^* t$. Knuth-Bendix completion [5] (if successful) gives a decision procedure: If an ES $\mathcal{E}$ is transformed into an equivalent convergent term rewrite system (TRS) $\mathcal{R}$, then $s \leftrightarrow_{\mathcal{E}}^* t$ iff the $\mathcal{R}$-normal forms of s and t coincide. (Note that completion does not construct such conversions explicitly.)

Example 1. For $\mathcal{E} = \{\mathsf{ff} \approx \mathsf{f}, \mathsf{ggf} \approx \mathsf{g}\}$ (where f and g are unary function symbols, for which we find it convenient to abbreviate $\mathsf{f}(\mathsf{g}(\mathsf{f}(x)))$ to fgf etc.) a possible choice of $\mathcal{R}$ is $\{\mathsf{ff} \to \mathsf{f}, \mathsf{gf} \to \mathsf{g}, \mathsf{gg} \to \mathsf{g}\}$. Since $\mathsf{fgf} \to_{\mathcal{R}}^* \mathsf{fg} \,{}_{\mathcal{R}}^*\!\leftarrow \mathsf{fgg}$, we have that $\mathsf{fgf} \approx \mathsf{fgg}$ follows from $\mathcal{E}$.

When we want to answer/certify whether $s \leftrightarrow_{\mathcal{E}}^* t$, we face the following situation: (1) It is hard to find a conversion but easy to certify a given one. (2) Under the assumption that Knuth-Bendix completion is successful, it is easy to decide the existence of a conversion (just rewrite s and t to $\mathcal{R}$-normal forms) but hard to certify this decision (e.g., by certifying that $\mathcal{E}$ and $\mathcal{R}$ are equivalent).

In this paper we introduce *recording completion*, which overcomes both problems. Recording completion keeps a history that allows us to reconstruct how the rules in $\mathcal{R}$ have been derived from the equations in $\mathcal{E}$. Then, from a join $s \to_{\mathcal{R}}^* \cdot \,{}_{\mathcal{R}}^*\!\leftarrow t$ a conversion $s \leftrightarrow_{\mathcal{E}}^* t$ can be reconstructed. Furthermore, recording completion enables the certification of completion proofs, i.e., to check that $\mathcal{R}$ and $\mathcal{E}$ are equivalent. Using equivalence together with confluence and termination certificates, it is also possible to certify that a conversion $s \leftrightarrow_{\mathcal{E}}^* t$ does *not* exist.

In addition to formalizing all required theorems like the critical pair theorem and soundness of completion, we have proven two new results: For finite completion proofs, i.e., where the completion procedure stops successfully after a finite number of steps, the strict encompassment condition (in the collapse-rule of Figure 1) is not required. Moreover, an infinite set of variables is essential for the critical pair theorem as well as modularity of confluence [8].

2 Proof Construction via Recording Completion

We extend the inference rules of completion [1] by a *history component* which allows us to infer how rules in $\mathcal{R}$ have been derived from equations in $\mathcal{E}$. The construction of a conversion $s \leftrightarrow_{\mathcal{E}}^* t$ (if possible at all) is then executed in three phases: (**record**) The inference rules of recording completion (see Figure 1) are applied to the ES $\mathcal{E}$. Upon success, a convergent TRS $\mathcal{R}$ (equivalent to $\mathcal{E}$) and a history $\mathcal{H}$ (recording how the rules in $\mathcal{R}$ have been derived) are computed. (**compare**) If the previous phase is successful, the test for $s \to_{\mathcal{R}}^* \cdot \,{}_{\mathcal{R}}^*\!\leftarrow t$ is performed. (**recall**) If the previous phase is successful, we construct $s \leftrightarrow_{\mathcal{E}}^* t$ from $s \to_{\mathcal{R}}^* \cdot \,{}_{\mathcal{R}}^*\!\leftarrow t$.

*Supported by Austrian Science Fund (FWF): P22467, P22767, J3202, and a grant by Hypo Tirol Bank.

$$\frac{(\mathcal{E},\mathcal{R},\mathcal{H})}{(\mathcal{E}\cup\{m\colon s\approx t\},\mathcal{R},\mathcal{H}\cup\{m\colon s \overset{j}{\leftarrow} u \overset{k}{\rightarrow} t\})}\ (\mathsf{deduce}) \qquad \text{if } s\ {}_{\mathcal{R}}\overset{j}{\leftarrow} u \overset{k}{\rightarrow}_{\mathcal{R}} t$$

$$\frac{(\mathcal{E}\cup\{i\colon s\approx t\},\mathcal{R},\mathcal{H})}{(\mathcal{E},\mathcal{R}\cup\{i\colon s\to t\},\mathcal{H})}\ (\mathsf{orient_l}) \qquad \text{if } s > t$$

$$\frac{(\mathcal{E}\cup\{i\colon s\approx t\},\mathcal{R},\mathcal{H}\cup\{i\colon s \overset{j}{\circ_1} u \overset{k}{\circ_2} t\})}{(\mathcal{E},\mathcal{R}\cup\{i\colon t\to s\},\mathcal{H}\cup\{i\colon t\,(\overset{k}{\circ_2})^{-1}\,u\,(\overset{j}{\circ_1})^{-1}\,s\})}\ (\mathsf{orient_r}) \qquad \text{if } t > s$$

$$\frac{(\mathcal{E}\cup\{i\colon s\approx t\},\mathcal{R},\mathcal{H})}{(\mathcal{E}\cup\{m\colon u\approx t\},\mathcal{R},\mathcal{H}\cup\{m\colon u \overset{l}{\leftarrow} s \overset{i}{\rightarrow} t\})}\ (\mathsf{simplify_l}) \qquad \text{if } s \overset{l}{\rightarrow}_{\mathcal{R}} u$$

$$\frac{(\mathcal{E}\cup\{i\colon s\approx t\},\mathcal{R},\mathcal{H})}{(\mathcal{E}\cup\{m\colon s\approx u\},\mathcal{R},\mathcal{H}\cup\{m\colon s \overset{i}{\rightarrow} t \overset{l}{\rightarrow} u\})}\ (\mathsf{simplify_r}) \qquad \text{if } t \overset{l}{\rightarrow}_{\mathcal{R}} u$$

$$\frac{(\mathcal{E}\cup\{i\colon s\approx s\},\mathcal{R},\mathcal{H}\cup\{i\colon s\circ_1 v\circ_2 s\})}{(\mathcal{E},\mathcal{R},\mathcal{H})}\ (\mathsf{delete})$$

$$\frac{(\mathcal{E},\mathcal{R}\cup\{i\colon s\to t\},\mathcal{H})}{(\mathcal{E},\mathcal{R}\cup\{m\colon s\to u\},\mathcal{H}\cup\{m\colon s \overset{i}{\rightarrow} t \overset{j}{\rightarrow} u\})}\ (\mathsf{compose}) \qquad \text{if } t \overset{j}{\rightarrow}_{\mathcal{R}} u$$

$$\frac{(\mathcal{E},\mathcal{R}\cup\{i\colon s\to t\},\mathcal{H})}{(\mathcal{E}\cup\{m\colon u\approx t\},\mathcal{R},\mathcal{H}\cup\{m\colon u \overset{j}{\leftarrow} s \overset{i}{\rightarrow} t\})}\ (\mathsf{collapse}) \qquad \text{if } s \overset{j}{\rightarrow}_{\mathcal{R}} u$$

Figure 1: The inference rules of *recording completion*.

In the sequel we give more details for each of the phases.

Record. The *record* phase uses the inference rules from Figure 1 where every rule/equation is annotated by a unique index i. Here, $\overset{i}{\rightarrow}_{\mathcal{R}}$ denotes an $\mathcal{R}$-reduction using the rule with index i. The inference rules are similar to the standard rules except for the following two differences: In the collapse-rule we dropped the condition of strict encompassment. Since we only consider finite runs, this condition is no longer required for soundness (cf. Theorem 1). Furthermore, there is a new history component $\mathcal{H}$ whose entries are of the form $i : s \overset{j}{\circ_1} u \overset{k}{\circ_2} t$ where i is the index of the entry, j and k are indices of equations/rules, s, u, and t are terms, and $\circ_1, \circ_2 \in \{\leftarrow, \rightarrow, \approx\}$.

Let us take a closer look at the extended inference rules. For deduce the peak $s \overset{j}{\leftarrow}_{\mathcal{R}} u \overset{k}{\rightarrow}_{\mathcal{R}} t$ that triggers the new equation $s \approx t$ is stored in a history entry (where m is assumed to be a fresh index that is larger than every earlier index). By $\mathsf{orient_l}$ we orient an equation from left to right and the corresponding history entry remains unchanged, whereas by $\mathsf{orient_r}$ we orient an equation from right to left and thus have to "mirror" the corresponding history entry. Here $>$ is a reduction order, which is part of the input. The rules $\mathsf{simplify_l}$ and $\mathsf{simplify_r}$ are used to $\mathcal{R}$-rewrite a left- or right-hand side of an equation. With delete we remove trivial equations from $\mathcal{E}$ and the corresponding history entry from $\mathcal{H}$. Finally, compose rewrites a right-hand side of a rule in $\mathcal{R}$ while collapse does the same for left-hand sides.

We write $(\mathcal{E}_i,\mathcal{R}_i,\mathcal{H}_i) \rightsquigarrow (\mathcal{E}_{i+1},\mathcal{R}_{i+1},\mathcal{H}_{i+1})$ for the application of an arbitrary inference rule to the triple $(\mathcal{E}_i,\mathcal{R}_i,\mathcal{H}_i)$ resulting in $(\mathcal{E}_{i+1},\mathcal{R}_{i+1},\mathcal{H}_{i+1})$.

Definition 1. A *run* of recording completion for $\mathcal{E}$ is a finite sequence $(\mathcal{E}_0,\mathcal{R}_0,\mathcal{H}_0) \rightsquigarrow^n (\mathcal{E}_n,\mathcal{R}_n,\mathcal{H}_n)$ of rule applications, where $\mathcal{E}_0 = \{i : s \approx t \mid s \approx t \in \mathcal{E}\}$ with fresh index i for each equation, $\mathcal{R}_0 = \varnothing$, and the initial history is $\mathcal{H}_0 = \{i : s \overset{i}{\rightarrow} t \approx t \mid i : s \approx t \in \mathcal{E}_0\}$. A

	$\mathcal{E}_0$	$\mathcal{R}_0$		$\mathcal{H}_0$
1:	$\mathsf{ff} \approx \mathsf{f}$	$\varnothing$	1:	$\mathsf{ff} \overset{1}{\to} \mathsf{f} \approx \mathsf{f}$
2:	$\mathsf{ggf} \approx \mathsf{g}$		2:	$\mathsf{ggf} \overset{2}{\to} \mathsf{g} \approx \mathsf{g}$

(a) Initial state.

$\mathcal{E}_n$		$\mathcal{R}_n$		$\mathcal{H}_n$
$\varnothing$	1:	$\mathsf{ff} \to \mathsf{f}$	1:	$\mathsf{ff} \overset{1}{\to} \mathsf{f} \overset{0}{\approx} \mathsf{f}$
	4:	$\mathsf{gf} \to \mathsf{g}$	2:	$\mathsf{ggf} \overset{2}{\to} \mathsf{g} \overset{0}{\approx} \mathsf{g}$
	5:	$\mathsf{gg} \to \mathsf{g}$	3:	$\mathsf{ggf} \overset{1}{\leftarrow} \mathsf{ggff} \overset{2}{\to} \mathsf{gf}$
			4:	$\mathsf{gf} \overset{3}{\leftarrow} \mathsf{ggf} \overset{2}{\to} \mathsf{g}$
			5:	$\mathsf{gg} \overset{4}{\leftarrow} \mathsf{ggf} \overset{2}{\to} \mathsf{g}$

(b) Final state.

Table 2: Example of recording completion.

run is *successful* if $\mathcal{E}_n = \varnothing$ and all critical pairs of $\mathcal{R}_n$ that are not contained in $\bigcup_{i \leq n} \mathcal{E}_i$ are joinable by $\mathcal{R}_n$. A run is *sound* if $\mathcal{R}_n$ is convergent and equivalent to $\mathcal{E}_0$.

The requirement on critical pairs for a successful run can be replaced by local confluence of $\mathcal{R}_n$.

Example 2. Recall $\mathcal{E}$ from Example 1. We start with the triple depicted in Table 2(a) and perform recording completion. Note that LPO with empty precedence orients all emerging rules in the desired direction. After orienting rules 1 and 2 from left to right we deduce a critical pair between rules 2 and 1, resulting in the equation $3 : \mathsf{ggf} \approx \mathsf{gf}$ and the history entry $3 : \mathsf{ggf} \overset{1}{\leftarrow} \mathsf{ggff} \overset{2}{\to} \mathsf{gf}$. Next we simplify the left-hand side of equation 3 by an application of rule 2 and obtain the equation $4 : \mathsf{g} \approx \mathsf{gf}$ with corresponding history entry $4 : \mathsf{g} \overset{2}{\leftarrow} \mathsf{ggf} \overset{3}{\to} \mathsf{gf}$. Orienting rule 4 from right to left causes the history entry to be mirrored, i.e., $4 : \mathsf{gf} \overset{3}{\leftarrow} \mathsf{ggf} \overset{2}{\to} \mathsf{g}$. Rules 2 and 4 allow to deduce equation $5 : \mathsf{gg} \approx \mathsf{g}$ with history $5 : \mathsf{gg} \overset{4}{\leftarrow} \mathsf{ggf} \overset{2}{\to} \mathsf{g}$, which we orient from left to right. Collapsing the left-hand side of rule 2 with rule 5 yields $6 : \mathsf{gf} \approx \mathsf{g}$ with $6 : \mathsf{gf} \overset{5}{\leftarrow} \mathsf{ggf} \overset{2}{\to} \mathsf{g}$. Now rule 4 simplifies equation 6 into $7 : \mathsf{g} \approx \mathsf{g}$ with $7 : \mathsf{g} \overset{4}{\leftarrow} \mathsf{gf} \overset{6}{\to} \mathsf{g}$, which is immediately deleted afterwards. Finally, $\mathcal{E}_n$ is empty and as all remaining critical pairs of $\mathcal{R}_n$ are joinable, the procedure can be stopped. Since there is no rule with index 6, the history entry 6 can be dropped. Hence, we obtain the result depicted in Table 2(b) where $\mathcal{R}_n$ is convergent and equivalent to $\mathcal{E}_0$.

We have formalized soundness of recording completion in IsaFoR [7] (see `Completion.thy`).

Theorem 1. *Every successful run of recording completion is sound.* □

Compare. Let $(\mathcal{E}, \varnothing, \mathcal{H}_0) \rightsquigarrow^n (\varnothing, \mathcal{R}, \mathcal{H}_n)$ be a successful run of recording completion and $s \approx t$ an equation. In the *compare* phase we test joinability of the terms s and t with respect to $\mathcal{R}$. If the two terms are joinable, then $s \approx t$ follows from $\mathcal{E}$ and the next phase constructs an $\mathcal{E}$-conversion $s \leftrightarrow^*_{\mathcal{E}} t$. Otherwise, $s \not\approx t$ w.r.t. $\mathcal{E}$. The *compare* phase is sound (cf. Theorem 1).

Recall. Let $(\mathcal{E}, \varnothing, \mathcal{H}_0) \rightsquigarrow^n (\varnothing, \mathcal{R}, \mathcal{H}_n)$ be a run of recording completion. Then the *recall* phase transforms a join $s \to^*_{\mathcal{R}} \cdot \ ^*_{\mathcal{R}}\!\leftarrow t$ into a conversion $s \leftrightarrow^*_{\mathcal{E}} t$ as follows. For each step $t_1 \overset{i}{\to} t_2$ where the index i is not in $\mathcal{E}$ the corresponding history entry is *inserted*. Let $i : \ell \to r$ be the rule with index i. Then there must be a history entry $i : \ell \overset{j}{\circ}_1 u \overset{k}{\circ}_2 r$, a position p, and a substitution σ such that $t_1|_p = \ell\sigma$ and $t_2|_p = r\sigma$. The step $t_1 \overset{i}{\to} t_2$ is replaced by the conversion $t_1 \overset{j}{\circ}_1 t_1[u\sigma]_p \overset{k}{\circ}_2 t_2$. This process terminates since $i > j, k$, i.e., any history entry (not in $\mathcal{H}_0$) refers to smaller indices and finally we arrive at a conversion using indices from $\mathcal{E}$.

The next lemma states the desired property of the *recall* phase. Note that we do not need a *successful* run of recording completion but any join $s \to_{\mathcal{R}}^* \cdot {}_{\mathcal{R}}^*\!\leftarrow t$ is transformed into $s \leftrightarrow_{\mathcal{E}}^* t$.

Lemma 1. *Let $(\mathcal{E}, \varnothing, \mathcal{H}_0) \rightsquigarrow^n (\mathcal{E}_n, \mathcal{R}, \mathcal{H}_n)$ be a run of recording completion. Then the recall phase transforms any join using rules from $\mathcal{R}$ into a conversion using rules from $\mathcal{E}$.* □

Alternatively, one can ensure $\leftrightarrow_{\mathcal{R}}^* \subseteq \leftrightarrow_{\mathcal{E}}^*$ to derive $s \leftrightarrow_{\mathcal{E}}^* t$ from $s \to_{\mathcal{R}}^* \cdot {}_{\mathcal{R}}^*\!\leftarrow t$. The former can be established by showing that all history entries $i\colon s' \circ_1^j u \circ_2^k t'$ are consequences of $\mathcal{E}$ (i.e., $s' \leftrightarrow_{\mathcal{E}}^* t'$) and can thus be used as auxiliary equations. To avoid cyclic references, history entries are processed in order of their indices. This approach requires the certifier to support such auxiliary equations. In return, proofs become much shorter as the history itself is the proof of $\leftrightarrow_{\mathcal{R}}^* \subseteq \leftrightarrow_{\mathcal{E}}^*$ which obviously has linear size. In contrast, the recall phase might produce certificates where the conversion $s \leftrightarrow_{\mathcal{E}}^* t$ is exponentially larger than the join $s \to_{\mathcal{R}}^* \cdot {}_{\mathcal{R}}^*\!\leftarrow t$.

3 Formalization and Certification

To maximize the reliability of the computed results, we have developed a verified certifier using the proof assistant Isabelle/HOL. Based on IsaFoR [7] the code generation facilities of Isabelle/HOL allow to generate the verified certifier CeTA, which is able to certify or falsify conversions, completion proofs, and equational proofs and disproofs which are performed via completion.[1] For the latter, although Theorem 1 has been formalized, it is not checked whether the completion rules are applied correctly. Instead it is just verified if the result of the completion procedure is a convergent TRS equivalent to the initial set of equations.

To decide whether $s \leftrightarrow_{\mathcal{E}}^* t$ holds it suffices to find a convergent TRS $\mathcal{R}$ that is equivalent to $\mathcal{E}$ and decide whether the $\mathcal{R}$-normal forms of s and t coincide.

For equivalence of $\mathcal{R}$ and $\mathcal{E}$ we have to consider two directions. To decide $\leftrightarrow_{\mathcal{E}}^* \subseteq \leftrightarrow_{\mathcal{R}}^*$, by convergence of $\mathcal{R}$ we just have to check that for all $s \approx t \in \mathcal{E}$, the $\mathcal{R}$-normal forms of s and t coincide. For the other direction, $\leftrightarrow_{\mathcal{R}}^* \subseteq \leftrightarrow_{\mathcal{E}}^*$, we have to guarantee $\ell \leftrightarrow_{\mathcal{E}}^* r$ for all $\ell \to r \in \mathcal{R}$. Here, we use the information from recording completion to get the required derivations.

Hence, to certify that such a proof is correct we have to guarantee that $\mathcal{R}$ is convergent by showing termination and local confluence. Concerning termination, already several techniques have been formalized in IsaFoR. Hence, the new part is on the certification of local confluence. Here, the key technique is the critical pair theorem of Huet [3]—making a result by Knuth and Bendix [5] explicit. It states that $\mathcal{R}$ is locally confluent iff all critical pairs of $\mathcal{R}$ are joinable.

During the formalization we detected that in general (no assumption on the set of variables $\mathcal{V}$) there is a problem of renaming variables in rules for building critical pairs. To solve this problem without demanding an infinite set of variables, we see two alternatives: Either keep the set of variables and when building critical pairs try to rename variables apart as good as possible; or use an enlarged set of variables in the definition of critical pairs (so that there are enough variables to perform renamings). It turns out that for both alternatives the critical pair theorem does not hold.

For the first alternative it is easy to see that joinability of critical pairs does not imply local confluence. To this end, consider $\mathcal{V} = \{x\}$ and $\mathcal{R} = \{\mathsf{f}(\mathsf{a}, x) \to \mathsf{a}, \mathsf{f}(x, \mathsf{b}) \to \mathsf{b}\}$. This TRS is not locally confluent due to the peak $\mathsf{a} \leftarrow \mathsf{f}(\mathsf{a}, \mathsf{b}) \to \mathsf{b}$. But without changing $\mathcal{V}$ it is not possible to rename the variables of the two rules in $\mathcal{R}$ apart, such that their left-hand sides are unifiable. Hence, for the first alternative all critical pairs are joinable.

[1] Both IsaFoR and CeTA are freely available from http://cl-informatik.uibk.ac.at/software/ceta/.

$$\mathcal{R}_1$$
$$\mathsf{f}(\mathsf{g}(x_1,x_2),\mathsf{g}(x_3,x_4)) \to \mathsf{h}(x_1,\mathsf{h}(x_2,\mathsf{g}(x_3,x_4)))$$
$$\mathsf{f}(\mathsf{g}(\mathsf{g}(x_1,x_2),\mathsf{g}(x_3,x_4)),x_5) \to \mathsf{h}(x_5,\mathsf{h}(\mathsf{g}(x_1,x_2),\mathsf{g}(x_3,x_4)))$$

$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$
$\mathsf{h}(\mathsf{g}(t,x_1),\mathsf{h}(x_2,x_3)) \to \mathsf{c}$	$\mathsf{h}(\mathsf{g}(y,x_1),\mathsf{h}(\mathsf{g}(x_2,x_3),\mathsf{g}(x_4,x_5))) \to \mathsf{c}$	$\mathsf{h}(x_1,\mathsf{c}) \to \mathsf{c}$
$\mathsf{h}(\mathsf{g}(x_1,t),\mathsf{h}(x_2,x_3)) \to \mathsf{c}$	$\mathsf{h}(\mathsf{g}(x_1,y),\mathsf{h}(\mathsf{g}(x_2,x_3),\mathsf{g}(x_4,x_5))) \to \mathsf{c}$	
$\mathsf{h}(x_1,\mathsf{h}(\mathsf{g}(t,x_2),x_3)) \to \mathsf{c}$	$\mathsf{h}(\mathsf{g}(x_1,x_2),\mathsf{h}(\mathsf{g}(y,x_3),\mathsf{g}(x_4,x_5))) \to \mathsf{c}$	
$\mathsf{h}(x_1,\mathsf{h}(\mathsf{g}(x_2,t),x_3)) \to \mathsf{c}$	$\mathsf{h}(\mathsf{g}(x_1,x_2),\mathsf{h}(\mathsf{g}(x_3,y),\mathsf{g}(x_4,x_5))) \to \mathsf{c}$	
$\mathsf{h}(x_1,\mathsf{h}(x_2,\mathsf{g}(t,x_3))) \to \mathsf{c}$	$\mathsf{h}(\mathsf{g}(x_1,x_2),\mathsf{h}(\mathsf{g}(x_3,x_4),\mathsf{g}(y,x_5))) \to \mathsf{c}$	
$\mathsf{h}(x_1,\mathsf{h}(x_2,\mathsf{g}(x_3,t))) \to \mathsf{c}$	$\mathsf{h}(\mathsf{g}(x_1,x_2),\mathsf{h}(\mathsf{g}(x_3,x_4),\mathsf{g}(x_5,y))) \to \mathsf{c}$	

Table 3: Rule schema for $\mathcal{R}_c$ with $y \in \mathcal{V}$ and $t \in \{\mathsf{c}, \mathsf{f}(x_4,x_5), \mathsf{g}(x_4,x_5), \mathsf{h}(x_4,x_5)\}$.

For the second alternative, $\mathcal{R}$ may be locally confluent although not every critical pair is joinable: the next example shows that if $\mathcal{V}$ is finite and $\mathcal{R}$ is locally confluent, then it need not be the case that all critical pairs of $\mathcal{R}$ are joinable.

Example 3. Let $\mathcal{R}_c = \bigcup_{i=1}^{4} \mathcal{R}_i$ be the TRS over $\mathcal{F} = \{\mathsf{f},\mathsf{g},\mathsf{h},\mathsf{c}\}$ and $\mathcal{V} = \{x_1,\ldots,x_5\}$ depicted in Table 3. It is constructed in such a way that each term $\mathsf{h}(\mathsf{g}(t_1,t_2),\mathsf{h}(\mathsf{g}(t_3,t_4),\mathsf{g}(t_5,t_6)))$ can be reduced to c (via $\mathcal{R}_2$ if some t_i is not a variable and via $\mathcal{R}_3$ if $t_i = t_j$ for $i < j$). Since there are only five different variables in $\mathcal{V}$, indeed every term $\mathsf{h}(\mathsf{g}(t_1,t_2),\mathsf{h}(\mathsf{g}(t_3,t_4),\mathsf{g}(t_5,t_6)))$ can be reduced to c. Moreover, all critical pairs, for which one of the rules is taken from $\mathcal{R}_c \setminus \mathcal{R}_1$, are joinable. Hence, the only critical pair that remains to be considered arises between the two rules of $\mathcal{R}_1$ where $u = \mathsf{f}(\mathsf{g}(\mathsf{g}(x_1,x_2),\mathsf{g}(x_3,x_4)),\mathsf{g}(x_5,x_6))$:

$$\mathsf{h}(\mathsf{g}(x_1,x_2),\mathsf{h}(\mathsf{g}(x_3,x_4),\mathsf{g}(x_5,x_6))) \leftarrow u \rightarrow \mathsf{h}(\mathsf{g}(x_5,x_6),\mathsf{h}(\mathsf{g}(x_1,x_2),\mathsf{g}(x_3,x_4)))$$

This critical pair is not joinable, as both terms are $\mathcal{R}_c$-normal forms. However, $\mathcal{R}_c$ is confluent since every *instance* of the critical pair (w.r.t. $\mathcal{T}(\mathcal{F},\mathcal{V})$) is joinable to c.

The example shows that confluence depends on the set of *variables* which most often is assumed to be infinite. Without this assumption, the requirement that all critical pairs have to be joinable can be too strict.[2] Another important consequence is that in the case of finite $\mathcal{V}$, Toyama's modularity result for confluence [8] does no longer hold.

Corollary 1. *Confluence is not a modular property of TRSs for an arbitrary set of variables.*

To summarize, it is not possible to formalize the critical pair theorem for arbitrary sets $\mathcal{V}$. Hence, we formalized it for strings, where it is conveniently possible to rename variables of rules apart without changing the type of variables (by using different prefixes). Of course, if $\mathcal{V}$ is infinite we can always obtain a renaming function (take *some* fresh variables) by the *Axiom of Choice*. However, then the definition of critical pairs is not executable.

Theorem 2. *A TRS over $\mathcal{T}(\mathcal{F}, String)$ is locally confluent iff all critical pairs are joinable.*

Note that the theorem does not require any variable-condition for $\mathcal{R}$. Hence, $\mathcal{R}$ may, e.g., contain left-hand sides which are variables or free variables in right-hand sides.

[2] We have only shown this result for $|\mathcal{V}| = 5$. However, $\mathcal{R}_c$ can be adapted to any finite $\mathcal{V}$ with $5 \leq |\mathcal{V}|$.

4 Implementation and Conclusion

We performed experiments[3] for completion proofs using KBCV [6] and MKBTT [9] (on 115 ESs).[4] Within a time limit of 300 seconds, KBCV could complete 86 ESs and MKBTT 80 ESs while both tools together succeeded on 94. The corresponding 94 completion proofs could be certified by CeTA (version 2.4). For an evaluation of other completion tools we refer to [4].

In our experiments we considered both possibilities (mentioned at the end of Section 2) to ensure $\leftrightarrow_{\mathcal{R}}^* \subseteq \leftrightarrow_{\mathcal{E}}^*$. While KBCV 1.6 performs the recall phase to explicitly construct $\ell \leftrightarrow_{\mathcal{E}}^* r$ for each $\ell \to r \in \mathcal{R}$, KBCV 1.7 just exports the relevant history entries, which are used as auxiliary equations. Hence it is not surprising that from the 86 ESs which KBCV 1.6 could complete only 80 have been certified. For two ESs (`TPTP_GRP487-1_theory` and `TPTP_GRP_490-1_theory`) the recall phase did not terminate within the time limit and for the remaining ESs (`LS94_P1`, `TPTP_GRP_481-1_theory`, `TPTP_GRP_486-1_theory`, `TPTP_GRP_490-1_theory`) the certificate was too large (365 MB, 230 MB, 406 MB, 581 MB) for CeTA. However, when using auxiliary equations all proofs could be computed and certified (typically within a second). Hence further optimization of the proof format seems dispensable.

While MKBTT follows recording completion, C*i*ME3 implements an annotated version of ordered completion [2]. Here—in contrast to our approach—the history is not saved as a standalone component but directly integrated into terms, equations, and rules. Hence a term t comes with an original version t^0, a current version t^*, and a reduction sequence from t^0 to t^*. Similarly an equation $s \approx t$ also contains all intermediate (rewrite) steps that show that both terms are equal. It requires further investigations to evaluate the pros and cons of the two approaches.

Acknowledgments: We thank Sarah Winkler for integrating certifiable output into MKBTT and helpful discussion.

References

[1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge Univ. Press, 1999.

[2] Évelyne Contejean and Pierre Corbineau. Reflecting proofs in first-order logic with equality. In *CADE 2005*, volume 3632 of *LNAI*, pages 7–22, 2005.

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

[4] Dominik Klein and Nao Hirokawa. Maximal completion. In *RTA 2011*, volume 10 of *LIPIcs*, pages 71–80, 2011.

[5] Donald E. Knuth and Peter Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, NY, 1970.

[6] Thomas Sternagel and Harald Zankl. KBCV – Knuth-Bendix completion visualizer. In *IJCAR 2012*, LNAI, 2012. To appear.

[7] René Thiemann and Christian Sternagel. Certification of termination proofs using CeTA. In *TPHOLs 2009*, volume 5674 of *LNCS*, pages 452–468, 2009.

[8] Yoshihito Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *Journal of the ACM*, 34(1):128–143, 1987.

[9] Sarah Winkler, Haruhiko Sato, Aart Middeldorp, and Masahito Kurihara. Multi-completion with termination tools. *Journal of Automated Reasoning*. doi:10.1007/s10817-012-9249-2. To appear.

[3] http://cl-informatik.uibk.ac.at/software/kbcv/experiments/12iwc
[4] http://cl-informatik.uibk.ac.at/software/mkbtt