Preparing FORT(ify) for the Confluence Competition*

Alexander Lochmann[1], Fabian Mitterwallner[1], Aart Middeldorp[1]

Department of Computer Science, University of Innsbruck, Austria
{alexander.lochmann,fabian.mitterwallner,aart.middeldorp}@uibk.ac.at

Abstract

Ground-confluence and confluence do not coincide. However, for the class of left-linear right-ground TRSs confluence can be reduced to ground-confluence by extending the signature with fresh constants. We present a formalization in Isabelle/HOL of a more general result, for linear variable-separated rewrite systems. From this formalization we obtain a sound procedure to decide confluence and commutation of such systems. We implemented this procedure in the decision tool FORT-h, which also can produce machine checkable proofs, and in the certifier FORTify to validate these.

1 Introduction

Dauchet and Tison [2] proved the decidability of the *first-order theory of rewriting* for the class of ground rewrite systems. The recent tool FORT-h [6] implements an extension of the decision procedure for the larger class of linear variable-separated rewrite systems. FORT-h is capable of producing certificates that witness the steps in the decision procedure. These certificates are validated by FORTify [6], a verified Haskell program obtained from the Isabelle/HOL formalization of the underlying theory reported in [5].

The decision procedure is based on tree automata techniques and hence restricted to properties on ground terms. In this paper we are concerned with extending FORT-h and FORTify to deal with confluence-related properties on arbitrary terms. This allows the combination of these tools to be the first participant that produces provably correct answers in categories like COM and UNC of the Confluence Competition (CoCo).

We assume familiarity with (first-order) term rewriting [1], but do not impose the usual variable restrictions on rewrite rules. In the next section we present the formalized signature extension results that allow to reduce confluence-related properties to properties on ground terms. Section 3 explains the changes made to FORT-h and FORTify. We conclude in Section 4 with suggestions for future research.

2 Theory

We start this section by recalling the results of [3, 7, 8] concerning the reduction of confluence-related properties to their ground versions. The first lemma is from [7, 8]. Here $\mathcal{P} = \{\mathsf{CR}, \mathsf{SCR}, \mathsf{WCR}, \mathsf{NFP}, \mathsf{UNR}, \mathsf{UNC}\}$ and, for a property $P \in \mathcal{P}$, $\mathsf{G}P$ denotes the property P restricted to ground terms. (SCR/WCR stands for strong/local confluence, NFP denotes the normal form property, and UNR/UNC stands for unique normal forms with respect to reduction/conversion.)

Lemma 1. *Let $\mathcal{R}$ be left-linear right-ground TRS over a signature $\mathcal{F}$ that contains at least one constant.*

*This work is supported by the Austrian Science Fund (FWF) project P30301.

1. $(\mathcal{F},\mathcal{R}) \vDash P \iff (\mathcal{F} \uplus \{c\},\mathcal{R}) \vDash \mathsf{G}P$ *for all* $P \in \mathcal{P} \setminus \{\mathsf{UNC}\}$
2. $(\mathcal{F},\mathcal{R}) \vDash \mathsf{UNC} \iff (\mathcal{F} \uplus \{c,d\},\mathcal{R}) \vDash \mathsf{GUNC}$
3. *If* $\mathcal{R}$ *is ground or* $\mathcal{F}$ *is monadic then* $(\mathcal{F},\mathcal{R}) \vDash P \iff (\mathcal{F},\mathcal{R}) \vDash \mathsf{G}P$ *for all* $P \in \mathcal{P}$. □

A signature is monadic if every function symbol has arity at most one. A formalization in Isabelle/HOL of the third item has been reported in [3]. For linear *variable-separated* TRSs, in which the left- and right-hand sides of rules do not share variables, the first item of Lemma 1 does not hold. In [3] a (non-formalized) proof is presented that two fresh constants are sufficient to reduce confluence to ground confluence.

Lemma 2 ([3, Theorem 6.4])**.** *If* $\mathcal{R}$ *is a linear variable-separated TRS over a signature* $\mathcal{F}$ *then* $(\mathcal{F},\mathcal{R}) \vDash \mathsf{CR} \iff (\mathcal{F} \uplus \{c,d\},\mathcal{R}) \vDash \mathsf{GCR}$. □

The necessity of adding two fresh constants follows from the following example from [3].

Example 3. Consider the linear variable-separated TRS $\mathcal{R}$ consisting of the single rule $\mathsf{a} \to x$ over the signature $\mathcal{F} = \{\mathsf{a}\}$. Since $x \mathrel{{}_{\mathcal{R}}{\leftarrow}} \mathsf{a} \to_{\mathcal{R}} y$ with distinct variables x and y, $\mathcal{R}$ is not confluent. Ground-confluence holds trivially as $\mathsf{a} \to_{\mathcal{R}} \mathsf{a}$ is the only rewrite step between ground terms. Adding a single fresh constant b does not destroy ground-confluence ($\mathsf{a} \to_{\mathcal{R}} \mathsf{a}$ and $\mathsf{a} \to_{\mathcal{R}} \mathsf{b}$ are the only steps). By adding a second fresh constant c, ground-confluence is lost: $\mathsf{b} \mathrel{{}_{\mathcal{R}}{\leftarrow}} \mathsf{a} \to_{\mathcal{R}} \mathsf{c}$.

We generalize Lemma 2 to commutation (COM). The proof below is formalized. The first preliminary lemma states that we can restrict attention to rewrite sequences that contain a root step. The proof, which performs a straightforward induction on the term structure and the multi-hole context closure of the rewrite relation, is omitted.

Lemma 4. *TRSs* $\mathcal{R}$ *and* $\mathcal{S}$ *over a common signature* $\mathcal{F}$ *commute if and only if the following two inclusions hold:*

$$\to^*_{\mathcal{R}^-} \cdot \to^{\epsilon}_{\mathcal{R}^-} \cdot \to^*_{\mathcal{R}^-} \cdot \to^*_{\mathcal{S}} \;\subseteq\; \to^*_{\mathcal{S}} \cdot \to^*_{\mathcal{R}^-} \qquad \to^*_{\mathcal{R}^-} \cdot \to^*_{\mathcal{S}} \cdot \to^{\epsilon}_{\mathcal{S}} \cdot \to^*_{\mathcal{S}} \;\subseteq\; \to^*_{\mathcal{S}} \cdot \to^*_{\mathcal{R}^-}$$

Root steps are important since they permit application of two arbitrary substitutions on the left and right of the relation. The next lemma is a key result. It allows the removal of introduced fresh constants while preserving the reachability relation. Note that variable-separation is not required.

Lemma 5. *Let* $\mathcal{R}$ *be a linear TRS over a signature* $\mathcal{F}$ *that contains a constant* c *which does not appear in* $\mathcal{R}$*. If* $s \to^*_{\mathcal{R}} t$ *with* $c \in \mathcal{F}\mathsf{un}(s) \setminus \mathcal{F}\mathsf{un}(t)$ *then* $s[u]_p \to^*_{\mathcal{R}} t$ *using the same rewrite rules at the same positions, for all terms* u *and positions* $p \in \mathcal{P}\mathsf{os}(s)$ *such that* $s|_p = c$.

The restriction to linear TRSs can also be lifted, at the expense of a more complicated replacement function and proof. Since the decision procedure implemented in FORT-h relies on linearity and variable-separation, we present a simple proof for linear TRSs. Due to calculations involving positions, the formalization in Isabelle/HOL was anything but simple.

Proof. We use induction on the length of $s \to^*_{\mathcal{R}} t$. If this length is zero then there is nothing to show as $\mathcal{F}\mathsf{un}(s) \setminus \mathcal{F}\mathsf{un}(t) = \varnothing$. Suppose $s \to_{\mathcal{R}} v \to^*_{\mathcal{R}} t$ and write $s = C[\ell\sigma] \to_{\mathcal{R}} C[r\sigma] = v$. Let p' be the position of the hole in C and let $p \in \mathcal{P}\mathsf{os}(s)$ with $s|_p = c$. We distinguish two cases.

If $p' \parallel p$ then $s[u]_p = (C[u]_p)[\ell\sigma]_{p'} \to_{\mathcal{R}} v'$ with $v' = (C[u]_p)[r\sigma]_{p'}$. Since $v|_p = C|_p = c$ we can apply the induction hypothesis to $v \to^*_{\mathcal{R}} t$. This yields $v' \to^*_{\mathcal{R}} t$ and hence $s[u]_p \to^*_{\mathcal{R}} t$ as desired.

In the remaining case, $p' \leqslant p$. From $s|_p = c$ and the fact that c does not appear in $\mathcal{R}$ we infer that there exists a variable $y \in \mathcal{V}\mathsf{ar}(\ell)$ such that $c \in \mathcal{F}\mathsf{un}(\sigma(y))$. Let q be the (unique) position of y in ℓ and consider the substitution

$$\tau(x) = \begin{cases} \sigma(y)[u]_{q'} & \text{if } x = y \\ \sigma(x) & \text{otherwise} \end{cases}$$

Here $q' = p \backslash (p'q)$ is the position of c in $\sigma(y)$. If $y \notin \mathcal{V}\mathsf{ar}(r)$ then $v = C[r\sigma] = C[r\tau]$ and thus $s[u]_p = C[\ell\tau] \to_{\mathcal{R}} C[r\tau] = v \to_{\mathcal{R}}^* t$. If $y \in \mathcal{V}\mathsf{ar}(r)$ then there exists a unique position $q'' \in \mathcal{P}\mathsf{os}(r)$ such that $r|_{q''} = y$. So $v|_{p'q''q'} = c$ and we obtain $s[u]_p = C[\ell\tau] \to_{\mathcal{R}} C[r\tau] = v[u]_{p'q''q'} \to_{\mathcal{R}}^* t$ from the induction hypothesis. □

Using the preceding two lemmata, the main result easily follows.

Lemma 6. *Linear variable-seperated TRSs $\mathcal{R}$ and $\mathcal{S}$ over a common signature $\mathcal{F}$ commute if and only if $\mathcal{R}$ and $\mathcal{S}$ ground-commute over $\mathcal{F} \uplus \{c, d\}$.*

Proof. First we prove the if direction. So suppose $\mathcal{R}$ and $\mathcal{S}$ commute on terms in $\mathcal{T}(\mathcal{F} \uplus \{c, d\})$. In order to conclude that $\mathcal{R}$ and $\mathcal{S}$ commute on terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$, according to Lemma 4 it suffices to show the inclusions

$$\to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{R}^-}^{\epsilon} \cdot \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{S}}^* \;\subseteq\; \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^* \qquad \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{S}}^* \cdot \to_{\mathcal{S}}^{\epsilon} \cdot \to_{\mathcal{S}}^* \;\subseteq\; \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^*$$

on terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. Suppose $s \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{R}^-}^{\epsilon} \cdot \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{S}}^* t$. Let the substitution σ_c map all variables to c and let σ_d map all variables to d. Since rewriting is closed under substitutions and the variable-separated rule used in the root step $\to_{\mathcal{R}^-}^{\epsilon}$ allows changing the substitution, we obtain $s\sigma_c \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{R}^-}^{\epsilon} \cdot \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{S}}^* t\sigma_d$. From ground commutation we obtain $s\sigma_c \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^* t\sigma_d$. Note that s and t are terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$ and hence do not contain the constants c and d. Therefore, $d \notin \mathcal{F}\mathsf{un}(s\sigma_c)$ and $c \notin \mathcal{F}\mathsf{un}(t\sigma_d)$. As a consequence, repeated applications of Lemma 5 transform $s\sigma_c \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^* t\sigma_d$ into a sequence $s \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^* t$ in which c and d do not appear, proving the first inclusion. The second inclusion $\to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{S}}^* \cdot \to_{\mathcal{S}}^{\epsilon} \cdot \to_{\mathcal{S}}^* \subseteq \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^*$ is obtained in the same way.

For the only-if direction we assume that $\mathcal{R}$ and $\mathcal{S}$ commute on terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$ and use Lemma 4 to establish the commutation of $\mathcal{R}$ and $\mathcal{S}$ on terms in $\mathcal{T}(\mathcal{F} \uplus \{c, d\})$. We prove the first inclusion. The second inclusion follows then by a symmetric argument. So let $s \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{R}^-}^{\epsilon} \cdot \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{S}}^* t$ and consider the following mapping $\phi\colon \mathcal{T}(\mathcal{F} \uplus \{c, d\}) \to \mathcal{T}(\mathcal{F}, \{x, y\})$:

$$\phi(t) = \begin{cases} x & \text{if } t = c \\ y & \text{if } t = d \\ f(\phi(t_1), \ldots, \phi(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Here x and y are distinct variables in $\mathcal{V}$. A straightforward induction proof shows $\phi(u) \to_{\mathcal{R}^-}^* \phi(v)$ whenever $u \to_{\mathcal{R}^-}^* v$, for all $u, v \in \mathcal{T}(\mathcal{F} \uplus \{c, d\})$. The same holds for $\mathcal{S}$. Hence, the given sequence from s to t is transformed into $\phi(s) \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{R}^-}^{\epsilon} \cdot \to_{\mathcal{R}^-}^* \cdot \to_{\mathcal{S}}^* \phi(t)$. Since c and d do not appear in the transformed sequence, we obtain $\phi(s) \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^* \phi(t)$ from the commutation of $\mathcal{R}$ and $\mathcal{S}$. Define the substitution $\tau = \{x \mapsto c, y \mapsto d\}$. Since rewriting is closed under substitution, $s = \phi(s)\tau \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}^-}^* \phi(t)\tau = t$. □

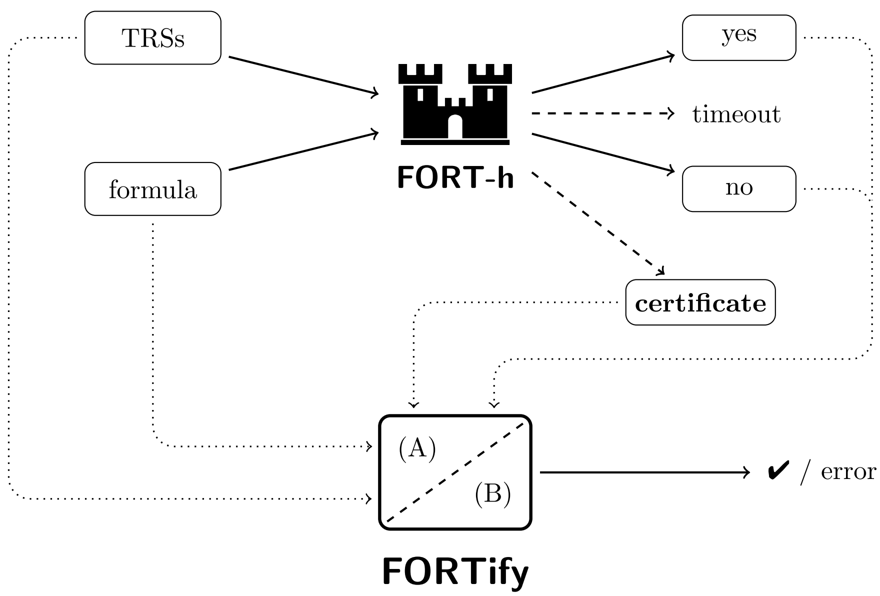

Figure 1: FORT-h and FORTify.

3 FORT-h and FORTify

The overall design of FORT-h and FORTify is shown in Figure 1. If FORT-h does not time out, it produces a certificate in the certificate language that is formally described in [6, Section 4]. Certificates can be viewed as a recipe for the certifier to perform certain operation on tree automata and formulas in order to confirm the yes/no claim of FORT-h. The certifier is the verified Haskell code base that is generated by Isabelle's code generation facility, corresponding to module (B) of FORTify. Module (A) contains a Haskell parser to translate strings representing formulas (TRSs, signatures, certificates) to semantically equivalent objects in the data types obtained from the generated code in module (B). The reader is referred to [6] for further details.

Here we briefly describe the required changes to this setup in order to accommodate the results mentioned in the preceding sections.

FORT-h already had support for some properties on open terms [6] based only on Lemma 2. If the input formula was one of the predefined macros for a property on open terms (e.g. CR), it would execute the decision procedure with the signature extended by two constants on the formula of the corresponding ground property (e.g. GCR). To improve the performance of the decision procedure we implemented the optimizations described in Lemma 1. This means the number of additional constants now depends on the properties of the input TRS, which in some cases leads to smaller signatures, therefore leading to faster decisions by the tool.

The more interesting changes relate to FORTify. Since the certificate serves as a proof that a formula holds for ground terms, we chose to keep the certificate format unchanged. The signature extension described in Lemmata 1, 2 and 6 were implemented as a preprocessing step of the formula which, just like FORT-h, checks if the input formula is a property on open terms. If that is the case, the signature is extended and the formula set to the corresponding ground property. Here care has to be taken that both FORT-h and FORTify use the same definitions for their ground property, since this formula has to match the one in the certificate. The choice to keep the certificate unchanged also means that the interface between FORT-h and FORTify remains unchanged and FORTify is fully backwards compatible. Note that this preprocessing step is implemented in module (A) of FORTify (see Figure 1) by hand, hence is not code generated from the formalization.

4 Conclusion

We showed that commutation of linear variable-separated TRSs reduces to ground-commutation after the signature is extended with two fresh constants. (This is not be confused with signature extension results for commutation, which is studied in [4, 9].) The proof is formalized in Isabelle/HOL and can be obtained from the website

https://fortissimo.uibk.ac.at/iwc2021

accompanying this paper. Precompiled binaries of the new versions of FORT-h and FORTify are available from the same site. Similar formalized proofs for UNR, UNC and NFP are expected soon.

The current implementation of FORTify supports certifying decisions of the properties CR and COM of FORT-h. At the moment these properties must appear at the root of the input formula. Possible future work is to permit these properties to appear within a formula. This would allow certifying results for a formula like $\mathsf{GCR} \land \neg\mathsf{CR}$. FORT-h already has support for this, but the results cannot be certified.

Another improvement would be moving the signature extension procedure from module (A) into the formality verified module (B). While this would necessarily change the interface between (A) and (B), the certificate format could still remain unchanged for backwards compatibility.

References

[1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. doi:10.1017/CBO9781139172752.

[2] Max Dauchet and Sophie Tison. The theory of ground rewrite systems is decidable. In *Proc. 5th LICS*, pages 242–248, 1990. doi:10.1109/LICS.1990.113750.

[3] Bertram Felgenhauer, Aart Middeldorp, T. V. H. Prathamesh, and Franziska Rapp. A verified ground confluence tool for linear variable-separated rewrite systems in Isabelle/HOL. In *Proc. 8th CPP*, pages 132–143, 2019. doi:10.1145/3293880.3294098.

[4] Nao Hirokawa. Commutation and signature extensions. In *Proc. 4th IWC*, pages 23–27, 2015.

[5] Alexander Lochmann, Aart Middeldorp, Fabian Mitterwallner, and Bertram Felgenhauer. A verified decision procedure for the first-order theory of rewriting for linear variable-separated rewrite systems. In *Proc. 10th CPP*, pages 250–263, 2021. doi:10.1145/3437992.3439918.

[6] Fabian Mitterwallner, Alexander Lochmann, Aart Middeldorp, and Bertram Felgenhauer. Certifying proofs in the first-order theory of rewriting. In *Proc. 27th TACAS*, volume 12652 of *LNCS*, pages 127–144, 2021. doi:10.1007/978-3-030-72013-1_7.

[7] Franziska Rapp and Aart Middeldorp. Confluence properties on open terms in the first-order theory of rewriting. In *Proc. 5th IWC*, pages 26–30, 2016.

[8] Franziska Rapp and Aart Middeldorp. FORT 2.0. In *Proc. 9th IJCAR*, volume 10900 of *LNAI*, pages 81–88, 2018. doi:10.1007/978-3-319-94205-6_6.

[9] Kiraku Shintani and Nao Hirokawa. CoLL: A confluence tool for left-linear term rewrite systems. In *Proc. 25th CADE*, volume 9195 of *LNCS*, pages 127–136, 2015. doi:10.1007/978-3-319-21401-6_8.