Layer Systems for Proving Confluence*

Bertram Felgenhauer, Harald Zankl, and Aart Middeldorp

Institute of Computer Science, University of Innsbruck, Austria

Abstract

We introduce layer systems for proving generalizations of the modularity of confluence for first-order rewrite systems. Layer systems specify how terms can be divided into layers. We establish structural conditions on those systems that imply confluence. Our abstract framework covers known results like many-sorted persistence, layer-preservation and currying. We present a counterexample to an extension of the former to order-sorted rewriting and derive new sufficient conditions for the extension to hold.

1998 ACM Subject Classification F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases Term rewriting, Confluence, Modularity, Persistence

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2011.288

1 Introduction

In this paper we revisit the celebrated modularity result of confluence, due to Toyama [13]. It states that the union of two confluent rewrite systems is confluent, provided the participating rewrite systems do not share function symbols. This result has been reproved several times, using category theory [10], ordered completion [6], and decreasing diagrams [14]. In practice, modularity is of limited use. More useful techniques, in the sense that rewrite systems can be decomposed into smaller systems that share function symbols and rules, are based on type introduction [2], layer-preservation [11], and commutativity [12].

Type introduction [16] restricts the set of terms that have to be considered to the well-typed terms according to any many-sorted type discipline which is compatible with the rewrite system under consideration. A property of rewrite systems for which type introduction is correct is called persistent and Aoto and Toyama [2] showed that confluence is persistent. In [1] they extended the latter result by considering an order-sorted type discipline. However, we show that the conditions imposed in [1] are not sufficient for confluence.

The proofs in [11] and [1, 2] are adaptations of the proof of Toyama's modularity result by Klop *et al.* [9]. A more complicated proof using concepts from [9] has been given by Kahrs, who showed in [7] that confluence is preserved under currying [8]. In this paper we introduce *layer systems* as a common framework to capture the results of [2, 7, 11, 13] and to identify appropriate conditions to restore the persistence of confluence for order-sorted rewriting [1]. Layer systems identify the parts that are available when decomposing terms. The key proof idea remains the same. We treat each such layer independently from the others where possible, and deal with interactions between layers separately. The main advantage of and motivation for our proof is that the result becomes reusable; instead of checking every detail of a complex proof, we have to check a couple of comparatively simple, structural conditions on layer systems instead.

The remainder of this paper is organized as follows. In the next section we recall preliminaries. Section 3 presents a counterexample to [1, Theorem 4.12]. Layer systems are

* This research was supported by the Austrian Science Fund (FWF) P22467-N23.

31st Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011).
Editors: Supratik Chakraborty, Amit Kumar; pp. 288–299

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

introduced in Section 4 and in Section 5 we develop conditions that guarantee the confluence of rewrite systems if a compatible layer system exists. In Section 6 we instantiate the abstract setting to cover the applications mentioned earlier before concluding in Section 7. Proofs can be found in an accompanying technical report [5] but key lemmata are presented to indicate the overall proof idea.

2 Preliminaries

We assume familiarity with rewriting [3]. A signature consists of function symbols $\mathcal{F}$ and variables $\mathcal{V}$. Each $f \in \mathcal{F}$ has a fixed arity $\mathrm{arity}(f)$. We assume that $\mathcal{V}$ is infinite. The set of terms over the signature $\langle \mathcal{F}, \mathcal{V} \rangle$ is denoted by $\mathcal{T}(\mathcal{F}, \mathcal{V})$. The sets of variables and function symbols occurring in a term t are denoted by $\mathcal{V}\mathsf{ar}(t)$ and $\mathcal{F}\mathsf{un}(t)$, respectively. The positions of a term t are strings of natural numbers, ϵ for the root, and ip if $t = f(t_1, \ldots, t_i, \ldots, t_n)$ and p is a position of t_i. $\mathcal{P}\mathsf{os}(t)$ is the set of all positions of t. For positions p, q, r we write $p < q$ if p is a proper prefix of q, $p \leqslant q$ if $p < q$ or $p = q$, $p \parallel q$ if neither $p < q$ nor $q \leqslant p$. If $p \leqslant q$ then $r = q \setminus p$ denotes the unique position such that $pr = q$. Given terms t and s, $t|_p$ is the subterm at position p of t, $t(p)$ is the root symbol of $t|_p$ and $t[s]_p$ denotes the result of replacing $t|_p$ by s in t. For $X \subseteq \mathcal{F} \cup \mathcal{V}$, we let $\mathcal{P}\mathsf{os}_X(t) = \{p \in \mathcal{P}\mathsf{os}(t) \mid t(p) \in X\}$. A substitution is a map $\sigma \colon \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ which extends homomorphically to terms.

A rewrite rule is a pair of terms $(\ell, r) \in \mathcal{T}(\mathcal{F}, \mathcal{V})^2$, written $\ell \to r$ such that $\ell \notin \mathcal{V}$ and $\mathcal{V}\mathsf{ar}(\ell) \supseteq \mathcal{V}\mathsf{ar}(r)$. A term rewrite system (TRS) is a set of rewrite rules. The rewrite relation induced by a TRS $\mathcal{R}$ is denoted $\to_{\mathcal{R}}$. We write $\leftarrow$, $\to^=$, $\to^+$ and $\to^*$ to denote the inverse, the reflexive closure, the transitive closure and the reflexive and transitive closure of a relation $\to$, respectively. A relation $\to$ is confluent if ${}^*\!\leftarrow \cdot \to^* \subseteq \to^* \cdot {}^*\!\leftarrow$ and terminating if $\to^+$ is well-founded. A TRS $\mathcal{R}$ inherits these properties from $\to_{\mathcal{R}}$.

Next we recall *many-sorted* terms. Let $\mathcal{S}$ be a set of sorts. A signature $\langle \mathcal{F}, \mathcal{V} \rangle$ is S-sorted, if every n-ary $f \in \mathcal{F}$ is equipped with a sort declaration $\alpha_1 \times \cdots \times \alpha_n \to \alpha$ where $\alpha_1, \ldots, \alpha_n, \alpha \in \mathcal{S}$ and every $x \in \mathcal{V}$ has exactly one sort $\alpha \in S$. We let $\mathcal{V}_\alpha = \{x \in \mathcal{V} \mid x \text{ has sort } \alpha\}$, and require that $\mathcal{V}_\alpha$ is infinite for all $\alpha \in S$. The set of many-sorted terms, $\mathcal{T}_{\mathcal{S}}(\mathcal{F}, \mathcal{V})$, is the union of the sets $\mathcal{T}_\alpha(\mathcal{F}, \mathcal{V})$ for $\alpha \in \mathcal{S}$ that are inductively defined as follows: $\mathcal{V}_\alpha \subseteq \mathcal{T}_\alpha(\mathcal{F}, \mathcal{V})$ and $f(t_1, \ldots, t_n) \in \mathcal{T}_\alpha(\mathcal{F}, \mathcal{V})$ whenever $f \in \mathcal{F}$ has sort declaration $\alpha_1 \times \cdots \times \alpha_n \to \alpha$ and $t_i \in \mathcal{T}_{\alpha_i}(\mathcal{F}, \mathcal{V})$ for all $1 \leqslant i \leqslant n$. If $t \in \mathcal{T}_\alpha(\mathcal{F}, \mathcal{V})$ for some $\alpha \in \mathcal{S}$ then we say that t has sort α and write $\mathrm{sort}(t) = \alpha$. A many-sorted (or $\mathcal{S}$-sorted) TRS consists of an $\mathcal{S}$-sorted signature and a set $\mathcal{R}$ of rewrite rules between many-sorted terms of the same sort. A property P of TRSs is called *persistent* if a many-sorted TRS has the property P if and only if its underlying (unsorted) TRS has the property P. To obtain $(\mathcal{S}, \succeq)$-*order-sorted* terms, we equip the sorts of an $\mathcal{S}$-sorted signature with a partial order $\succeq$. Each set $\mathcal{T}_\alpha(\mathcal{F}, \mathcal{V})$ of $\mathcal{S}$-sorted terms of sort α is extended with $f(t_1, \ldots, t_n)$ whenever $f \in \mathcal{F}$ has sort declaration $\alpha_1 \times \cdots \times \alpha_n \to \alpha$ and, for all $1 \leqslant i \leqslant n$, $t_i \in \mathcal{T}_{\beta_i}(\mathcal{F}, \mathcal{V})$ for some sort β_i with $\alpha_i \succeq \beta_i$. A variable occurrence $x = t|_p \in \mathcal{V}_\alpha$ is called *strictly bound* if $p = \epsilon$ or its sort exactly matches that of its context, i.e., $\alpha_i = \beta_i$ above.

3 A Counterexample

The result claimed in [1] states that the underlying unsorted TRS $\mathcal{R}$ of a confluent $(\mathcal{S}, \succeq)$-order-sorted TRS $\mathcal{R}$ is confluent (on arbitrary terms) provided the strict part of $\succeq$ is well-founded, $\mathrm{sort}(\ell) \succeq \mathrm{sort}(r)$ and variable occurrences are strictly bound in ℓ and r for every rewrite rule $\ell \to r \in \mathcal{R}$. Below we give a counterexample to this claim (we derive a corrected criterion in Section 6).

▶ **Example 3.1.** Consider the TRS $\mathcal{R}$ consisting of $\mathsf{c}(x) \to x$ and the following rewrite rules:

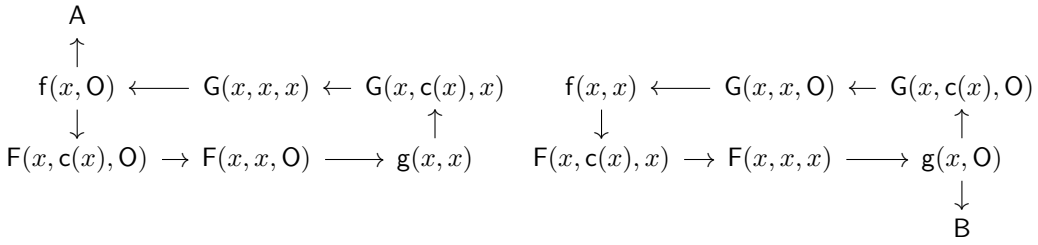
$$\mathsf{f}(x,y) \to \mathsf{F}(x,\mathsf{c}(x),y) \quad \mathsf{f}(x,\mathsf{O}) \to \mathsf{A} \quad \mathsf{F}(x,y,\mathsf{O}) \to \mathsf{g}(x,y) \quad \mathsf{F}(x,y,y) \to \mathsf{g}(x,\mathsf{O})$$
$$\mathsf{g}(x,y) \to \mathsf{G}(x,\mathsf{c}(x),y) \quad \mathsf{g}(x,\mathsf{O}) \to \mathsf{B} \quad \mathsf{G}(x,y,\mathsf{O}) \to \mathsf{f}(x,y) \quad \mathsf{G}(x,y,y) \to \mathsf{f}(x,\mathsf{O})$$

We take $\mathcal{S} = \{0,1,2\}$ as sorts with $1 \succeq 0$ and the signature given by

$$\mathsf{f},\mathsf{g} : 0 \times 1 \to 2 \quad \mathsf{A},\mathsf{B} : 2 \quad \mathsf{c} : 0 \to 1 \quad \mathsf{F},\mathsf{G} : 0 \times 1 \times 1 \to 2 \quad \mathsf{O}, y : 1 \quad x : 0$$

All rules except for $\mathsf{c}(x) \to x$ are many-sorted and sort-preserving, while $\mathrm{sort}(\mathsf{c}(x)) = 1 \succeq 0 = \mathrm{sort}(x)$ for $\mathsf{c}(x) \to x$. Hence $\mathcal{R}$ satisfies the aforementioned constraints from [1].

The rewrite relation defined by $\mathcal{R}$ is essentially finite: since c does not occur in any left-hand side of any rewrite rule but $\mathsf{c}(x) \to x$, we can apply this rule to any term and remove all occurrences of c, without interfering with any other future rewrite steps. The remaining order-sorted terms are flat terms (i.e., of depth 1 or 0), and there are only finitely many of those up to variable renaming. The interesting part of the rewrite relation looks as follows:



Note that x cannot be replaced by O anywhere, since this would make the terms ill-sorted. Hence $\mathcal{R}$ is confluent on $\mathcal{T}_{\mathcal{S}}(\mathcal{F},\mathcal{V})$. However, the corresponding unsorted TRS $\mathcal{R}$ is not confluent, since $\mathsf{A} \leftarrow \mathsf{f}(\mathsf{O},\mathsf{O}) \to^* \mathsf{g}(\mathsf{O},\mathsf{O}) \to \mathsf{B}$ for normal forms A and B.

4 Layer Systems

The existing proofs of modularity and its generalizations work by dividing terms into a top context (or native part) and principal subterms (aliens). This process can be characterized by the set of terms that are allowed as top contexts. We call this set a *layer system*. Layers are contexts, which can be represented by terms with holes. We use a fresh constant $\Box$ to denote such holes. Terms with holes can be merged.

▶ **Definition 4.1.** The partial function merge $\sqcup\colon \mathcal{T}(\mathcal{F} \cup \{\Box\},\mathcal{V})^2 \to \mathcal{T}(\mathcal{F} \cup \{\Box\},\mathcal{V})$ satisfies:

- $\Box \sqcup t = t \sqcup \Box = t$ for $t \in \mathcal{T}(\mathcal{F} \cup \{\Box\},\mathcal{V})$,
- $x \sqcup x = x$ for $x \in \mathcal{V}$,
- $f(s_1,\dots,s_n) \sqcup f(t_1,\dots,t_n) = f(u_1,\dots,u_n)$ if $f \in \mathcal{F}$ and $s_i \sqcup t_i = u_i$ for $1 \leqslant i \leqslant n$.

▶ **Example 4.2.** We have $\mathsf{f}(\mathsf{h}(\Box),\Box) \sqcup \mathsf{f}(\Box,\mathsf{g}(y)) = \mathsf{f}(\mathsf{h}(\Box),\mathsf{g}(y))$ but $\mathsf{f}(\mathsf{h}(\Box),\mathsf{g}(x)) \sqcup \mathsf{f}(\Box,\mathsf{g}(y))$ and $\mathsf{f}(\mathsf{h}(\Box),\Box) \sqcup \mathsf{f}(\mathsf{g}(\Box),\Box)$ are undefined (different non-hole symbols cannot be merged).

▶ **Definition 4.3.** A *layer system* is a set of terms $\mathbb{L} \subseteq \mathcal{T}(\mathcal{F} \cup \{\Box\},\mathcal{V})$ satisfying:

1. $\Box \in \mathbb{L}$, $\mathcal{V} \subseteq \mathbb{L}$, and $f(\Box,\dots,\Box) \in \mathbb{L}$ for $f \in \mathcal{F}$.
2. $\mathbb{L}$ allows *replacing holes by variables and vice versa*: If $t[\Box]_p \in \mathbb{L}$ then $\{x \in \mathcal{V} \mid t[x]_p \in \mathbb{L}\}$ is an infinite set. Furthermore, if $t[x]_p \in \mathbb{L}$ and $x \in \mathcal{V}$ then $t[\Box]_p \in \mathbb{L}$.
3. $\mathbb{L}$ is closed under *merging at function positions in* $\mathcal{F}$: If $s,t \in \mathbb{L}$, $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(t)$, and $u := t|_p \sqcup s$ is defined then $t[u]_p \in \mathbb{L}$.
4. $\mathbb{L}_{\mathcal{T}} := \mathbb{L} \cap \mathcal{T}(\mathcal{F},\mathcal{V})$ is closed under $\to_{\mathcal{R}}$: If $s \in \mathbb{L}_{\mathcal{T}}$ and $s \to_{\mathcal{R}} t$ then $t \in \mathbb{L}_{\mathcal{T}}$.

Requiring that $\mathbb{L}$ is closed under $\to_{\mathcal{R}}$ would imply Definition 4.3(4), but the weaker condition helps applications. For example, in the case of order-sorted persistence we do not have to worry about holes at positions with incompatible types, simplifying the proof that order-sorted terms form a layer system under certain conditions on $\mathcal{R}$ (cf. Theorem 6.2).

For pretty much the same reason, we do not require that holes can be replaced by any variable in Definition 4.3(2)—for example, in the order-sorted case, we would otherwise have to treat variables as essentially untyped, complicating that application.

Definition 4.3(3) allows one to merge two layers into a larger one. This is essential to obtain a *least layered representation* (cf. Theorem 5.2(4)) while Definition 4.3(1) ensures that any term can be layered (cf. Lemma 4.10(1)), by starting a new layer at every symbol.

We fix some layer system $\mathbb{L}$ that we want to use for layering terms. A key element of our proof is that this layering is made explicit using the notion of split terms.

▶ **Definition 4.4.** A *split term* $\hat{t}$ is a pair $\hat{t} = \langle t, P\rangle$ made of a term $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$ and a set of so-called *special* positions $P \subseteq \mathcal{P}\mathsf{os}(t)$ of $\hat{t}$. We write $P_{\hat{t}}$ for the special positions of $\hat{t}$. We call t the *underlying term* of $\hat{t}$ and often denote it by t. A special position is called *principal* if it is a minimal special position with respect to $<$. A split term $\langle t, P\rangle$ is *simple* if $P = \varnothing$, *special* if $\epsilon \in P$ and *normal* if $\epsilon \notin P$. The set of split terms is denoted by $\widehat{\mathcal{T}}(\mathcal{F},\mathcal{V})$.

We identify a term t with the unique corresponding simple split term $\langle t, \varnothing\rangle$.

▶ **Definition 4.5.** We define split counterparts to standard operations on terms.

subterm $\hat{t}|_p := \langle t|_p, \{q \setminus p \mid q \in P_{\hat{t}} \text{ with } q \geqslant p\}\rangle$. Note that if p is special then $\hat{t}|_p$ is special.

modifying special positions at the root $\hat{t}|_- := \langle t, P_{\hat{t}} \setminus \{\epsilon\}\rangle$ and $\epsilon(\hat{t}) := \langle t, P_{\hat{t}} \cup \{\epsilon\}\rangle$. We let $\hat{t}|_{p-} := (\hat{t}|_p)|_-$ and we will also use the notation $\hat{t}|_{p*}$ to denote one of $\hat{t}|_p$ or $\hat{t}|_{p-}$. If p is a principal position of $\hat{t}$, then $\hat{t}|_{p-}$ is a *principal subterm* of $\hat{t}$.

replacing subterms $\hat{t}[\hat{s}]_p := \langle t[s]_p, \{q \mid q \in P_{\hat{t}} \text{ with } q \not> p\} \cup \{pq \mid q \in P_{\hat{s}}\}\rangle$. For a set of pairwise parallel positions $Q \subseteq \mathcal{P}\mathsf{os}(t)$ we also write $\hat{t}[\hat{s}_q]_{q\in Q}$ for the split term obtained from $\hat{t}$ by replacing $t|_q$ by $\hat{s}_q$ for each $q \in Q$.

substitution If $\sigma\colon \mathcal{V} \to \widehat{\mathcal{T}}(\mathcal{F},\mathcal{V})$ then $\sigma(\hat{t}) = \hat{t}[\sigma(t|_p)]_{p\in\mathcal{P}\mathsf{os}_{\mathcal{V}}(t)}$.

Next we introduce two natural ways to represent split terms.

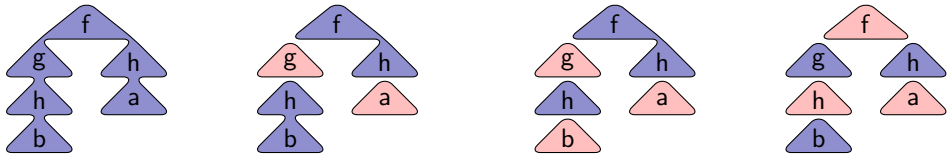
▶ **Definition 4.6.** A *split context* is a split term $\hat{C} = \langle C, P\rangle \in \widehat{\mathcal{T}}(\mathcal{F} \cup \{\Box\},\mathcal{V})$ such that $P = \{p \in \mathcal{P}\mathsf{os}(C) \mid C|_p = \Box\}$. We define $\hat{C}$ as the unique split context with underlying term C. For a split context $\hat{C}$ with n holes and normal terms $\hat{t}_i$ we denote by $\hat{C}[\![\hat{t}_1,\dots,\hat{t}_n]\!]$ the split term obtained by replacing the holes in $\hat{C}$ by $\hat{t}_1$ to $\hat{t}_n$ from left to right.

It is easy to see that any split term $\hat{t}$ can be uniquely represented as $\hat{t} = \hat{C}[\![\hat{t}_1,\dots,\hat{t}_n]\!]$. The corresponding split context is the *top layer* of $\hat{t}$.

▶ **Definition 4.7.** Let $\hat{t}$ be a split term. The *top layer* $\hat{L}(\hat{t})$ of $\hat{t}$ is obtained by replacing all its principal subterms by $\Box$. It is a split context as all remaining special positions are holes. We write $L(\hat{t})$ for the corresponding unsplit context. The *rank* of a split term $\hat{t} = \hat{C}[\![\hat{t}_1,\dots,\hat{t}_n]\!]$ is defined recursively by $\operatorname{rank}(\hat{t}) = 1 + \max\{0, \operatorname{rank}(\hat{t}_i) \mid 1 \leqslant i \leqslant n\}$.

The rank is useful for inductive proofs since principal subterms have smaller rank than the term itself. Next we use layer systems to restrict how terms can be split.

▶ **Definition 4.8.** A split term $\hat{t}$ is *layered according to* $\mathbb{L}$ or just *layered* if $L(\hat{t}) \in \mathbb{L}$ and $L(\hat{t}|_{p-}) \in \mathbb{L}$ for all $p \in P_{\hat{t}}$. The *layers* of $\hat{t}$ are the unsplit contexts $L(\hat{t})$ and $L(\hat{t}|_{p-})$ for $p \in P_{\hat{t}}$. For a layered term $\hat{s}$, a layer $s' \in \mathbb{L}_{\mathcal{T}}$ and a substitution σ we say that $\sigma(s')$ *designates* $\hat{s}$ if $\sigma(s') = \hat{s}$, $\sigma(x) \neq x$ implies $\sigma(x)$ is special, and $\sigma(x) = x$ for $x \in \mathcal{V}\mathsf{ar}(s)$.



(a) $P_{\hat{t}} = \varnothing$ **(b)** $P_{\hat{t}} = \{1, 11, 21\}$ **(c)** $P_{\hat{t}} = \{1, 11, 111, 21\}$ **(d)** $P_{\hat{t}} = \mathcal{P}\mathsf{os}(t)$

Figure 1 Some split terms for $t = \mathsf{f}(\mathsf{g}(\mathsf{h}(\mathsf{b})), \mathsf{h}(\mathsf{a}))$.

Designation is similar to the division of terms into a cap and an alien substitution in [6].

▶ **Example 4.9.** Figure 1 shows some split terms. Layers are colored alternately between (dark) blue and (light) red with the top layer (if non-empty) marked blue. Figure 1(a) depicts a simple term. The term in Figure 1(d) is special, the others are normal. In Figure 1(b) the principal positions are 1 and 21 with $\langle \mathsf{g}(\mathsf{h}(\mathsf{b})), \{1\}\rangle$ and $\langle \mathsf{a}, \varnothing\rangle$ as the corresponding principal subterms. The ranks of the terms in Figures 1(c) and 1(d) are 4 and 5 and the top layers are $\langle \mathsf{f}(\Box, \mathsf{h}(\Box)), \{1, 21\}\rangle$ and $\langle \Box, \{\epsilon\}\rangle$, respectively. For suitable $\mathbb{L}$, the term in Figure 1(b) is designated by $\sigma(\mathsf{f}(x, \mathsf{h}(y)))$ with $\sigma(x) = \langle \mathsf{g}(\mathsf{h}(\mathsf{b})), \{\epsilon, 1\}\rangle$ and $\sigma(y) = \langle \mathsf{a}, \{\epsilon\}\rangle$.

▶ **Lemma 4.10.** *We state some fundamental properties of layered terms.*

1. *For any term s the split term $\langle s, \mathcal{P}\mathsf{os}(s)\rangle$ is layered.*
2. *If $\hat{t}$ is layered and $p \in P_{\hat{t}}$ then $\hat{t}|_p$ and $\hat{t}|_{p-}$ are layered.*
3. *If $\hat{t}_1 = \langle t, P_1\rangle$ and $\hat{t}_2 = \langle t, P_2\rangle$ are layered then $\langle t, P_1 \cap P_2\rangle$ is layered.*

As a consequence of Lemma 4.10, every term t has a *least layered representation* $\hat{t}_{\mathbb{L}}$, which can be defined as $\hat{t}_{\mathbb{L}} := \langle t, \bigcap\{P \mid \langle t, P\rangle \text{ is layered according to } \mathbb{L}\}\rangle$.

▶ **Example 4.11.** Consider the $\{0, 1\}$-sorted signature with $\mathsf{f} : 0 \times 1 \to 0$, $\mathsf{g} : 0 \to 1$, $\mathsf{h} : 1 \to 1$, $\mathsf{a} : 0$, $\mathsf{b} : 1$, and let $\mathbb{L} := \mathcal{T}_{\mathcal{S}}(\mathcal{F} \cup \{\Box\}, \mathcal{V})$ where holes can appear anywhere in terms, both with sort 0 and 1. The term in Figure 1(a) is not layered according to $\mathbb{L}$, the others are. Figure 1(b) depicts $\hat{t}_{\mathbb{L}}$, i.e., a new layer starts if and only if a subterm does not match the sort of the context.

▶ **Lemma 4.12.** *Any layered term $\hat{s}$ is designated by some $\sigma(s')$ with $s' = L(\hat{s})[\vec{x}] \in \mathbb{L}$, $x_i \in \mathcal{V}$.*

Designations will be used to simulate so-called *outer* rewrite steps (cf. Definition 5.1) by rewrite steps inside $\mathbb{L}_{\mathcal{T}}$. Lemma 4.12 is a first step towards this goal.

5 Layer Systems for Confluence

In this section we first turn to rewriting split terms and then impose conditions on layer systems such that a TRS $\mathcal{R}$ is confluent if it is confluent on $\mathbb{L}_{\mathcal{T}}$. The overall idea is to start with rewrite sequences on terms, turn them to rewrite sequences on layered terms, which enjoy confluence, and map the resulting joining sequences back to (unsplit) terms.

▶ **Definition 5.1.** Let $\hat{s}$ and $\hat{t}$ be split terms. A *rule step* $\hat{s} \to_{p,\ell\to r} \hat{t}$ satisfies $\ell \to r \in \mathcal{R}$, $\hat{s}|_{p-} = \sigma(\ell)$ and $\hat{t} = \hat{s}[\sigma(r)]_p$ for some substitution σ. We also write $\hat{s} \to_{\mathcal{R}} \hat{t}$. A rule step $\hat{s} \to_{p,\ell\to r} \hat{t}$ is *inner*, $\hat{s} \to_i \hat{t}$, if $p \geqslant q$ for some $q \in P_{\hat{s}}$, and *outer*, $\hat{s} \to_o \hat{t}$, otherwise. A *fusion step* $\hat{s} \rightsquigarrow \hat{t}$ satisfies $s = t$ and $P_{\hat{s}} \supseteq P_{\hat{t}}$. It is *proper*, $\hat{s} \overset{\neq}{\rightsquigarrow} \hat{t}$, if $P_{\hat{s}} \supsetneq P_{\hat{t}}$. The union of $\to_{\mathcal{R}}$ and $\rightsquigarrow$ is denoted by $\rightsquigarrow_{\mathcal{R}}$.

Rule steps correspond to normal rewrite steps. Since ℓ and r are (simple) terms, matching is constrained to a single layer of a term, and the result of the rewrite step will only modify that layer and possibly permute, erase or duplicate special subterms below the layer.

Fusion steps, on the other hand, allow adjacent layers to be merged into a single one. Note that even when starting from a least layered representation, $\hat{s}_{\mathbb{L}} \to_{\mathcal{R}} \hat{t}$ may result in a term $\hat{t}$ that allows a proper fusion step. In the classical modularity setting this may happen after applying a collapsing rule. In many previous proofs from the literature, fusion was implicit. By making fusion explicit, we can capture the phenomena that destroy modularity more precisely, and, perhaps more importantly, we can delay fusion in joining rewrite sequences until we can comfortably deal with it.

▶ **Lemma 5.2.** *Rewriting on split terms has the following properties.*

1. *If $\hat{s} \rightsquigarrow_{\mathcal{R}} \hat{t}$ then $\sigma(\hat{s}) \rightsquigarrow_{\mathcal{R}} \sigma(\hat{t})$ for any (split) substitution σ.*
2. *If $\hat{s} \to_o \hat{t}$ then every principal subterm of $\hat{t}$ is a principal subterm of $\hat{s}$.*
3. *If $\hat{s} \rightsquigarrow_{\mathcal{R}} \hat{t}$ then $\mathrm{rank}(\hat{s}) \geqslant \mathrm{rank}(\hat{t})$.*
4. *If $\hat{t}$ is layered then $\hat{t} \rightsquigarrow \hat{t}_{\mathbb{L}}$.*

We extend fusion steps to substitutions: $\sigma \rightsquigarrow \tau$ if $\sigma(x) \rightsquigarrow \tau(x)$ for all $x \in \mathcal{V}$.

▶ **Lemma 5.3.** *Let $\hat{s}$ and $\hat{t}$ be split terms. If $\hat{s} \rightsquigarrow \hat{s}'$ then $\hat{t}[\hat{s}]_p \rightsquigarrow \hat{t}[\hat{s}']_p$ for any $p \in \mathcal{P}\mathsf{os}(t)$. For split substitutions σ and σ', if $\sigma \rightsquigarrow \sigma'$ then $\sigma(\hat{t}) \rightsquigarrow \sigma'(\hat{t})$.*

We impose suitable additional constraints on the layer system $\mathbb{L}$ to prove that if $\to_{\mathcal{R}}$ is confluent on $\mathbb{L}_{\mathcal{T}}$ then $\to_{\mathcal{R}}$ is confluent. The converse is trivial, cf. Definition 4.3(4).

▶ **Definition 5.4.** A layer system $\mathbb{L}$ is *weakly consistent* with a TRS $\mathcal{R}$ if for all $\ell \to r \in \mathcal{R}$:

1. $\ell \in \mathbb{L}$,
2. $t[t|_{pq'}]_{pq} \in \mathbb{L}$ whenever $t \in \mathbb{L}$, $\ell|_q = \ell|_{q'} \in \mathcal{V}$ and $t|_p$ can be obtained from ℓ by replacing variables by terms—where different instances of the same variable may be replaced by different terms. (For left-linear TRSs this means that ℓ matches $t|_p$.)

Definition 5.4(2) always holds for left-linear systems, but both conditions are essential for general systems. The next example shows how Definition 5.4(1,2) may fail for non-confluent systems.

▶ **Example 5.5.** Consider the non-confluent $\mathcal{R} = \{\mathsf{f}(\mathsf{f}(x)) \to \mathsf{a}, \mathsf{f}(\mathsf{f}(x)) \to \mathsf{b}\}$. Then $\mathbb{L} = \mathcal{V} \cup \{\Box, \mathsf{a}, \mathsf{b}, \mathsf{f}(\Box)\} \cup \{\mathsf{f}(x) \mid x \in \mathcal{V}\}$ is a layer system such that $\to_{\mathcal{R}}$ is confluent on $\mathbb{L}_{\mathcal{T}}$. We have $\mathsf{f}(\mathsf{f}(x)) \notin \mathbb{L}$, so it violates Definition 5.4(1).

Next consider $\mathcal{R} = \{\mathsf{f}(x, x) \to \mathsf{a}, \mathsf{f}(\mathsf{a}, x) \to \mathsf{b}\}$ with $\mathbb{L} = \{\mathsf{f}(x, y), \mathsf{f}(\mathsf{a}, x), x, \mathsf{a}, \mathsf{b} \mid x, y \in \mathcal{V} \cup \{\Box\}\}$. This is a layer system satisfying Definition 5.4(1) and $\to_{\mathcal{R}}$ is confluent on $\mathbb{L}_{\mathcal{T}}$. However, since $t = \ell' = \mathsf{f}(\mathsf{a}, x)$ can be obtained from $\ell = \mathsf{f}(x, x)$ by replacing the first x by a, by Definition 5.4(2), we should have $t[\ell'|_1]_2 = \mathsf{f}(\mathsf{a}, \mathsf{a}) \in \mathbb{L}$.

As a final example, consider $\mathcal{R} = \{\mathsf{f}(x, x, y) \to \mathsf{g}(x, y), \mathsf{f}(x, y, z) \to \mathsf{a}\}$, and $\mathbb{L} = \{\mathsf{f}(x, y, z) \mid x \in \mathcal{V}_1 \cup \{\Box\}, y \in \mathcal{V}_2 \cup \{\Box\}, z \in \mathcal{V} \cup \{\mathsf{a}, \Box\}\} \cup \{\mathsf{a}, x, \mathsf{g}(x, y) \mid x, y \in \mathcal{V} \cup \{\Box\}\}$, where $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathcal{V}$ are disjoint and infinite. Again, $\to_{\mathcal{R}}$ is confluent on $\mathbb{L}_{\mathcal{T}}$ but not on $\mathcal{T}(\mathcal{F}, \mathcal{V})$. In this case, $s = \mathsf{f}(\mathsf{a}, \mathsf{a}, \mathsf{a})$ has $\hat{s}_{\mathbb{L}} = \langle s, \{1, 2\}\rangle$ and $\hat{s}_{\mathbb{L}} \to_{\mathcal{R}} \hat{t} = \langle \mathsf{g}(\mathsf{a}, \mathsf{a}), \{1\}\rangle$, but $\hat{t}$ is not layered. In this case, $\mathbb{L}$ violates Definition 5.4(2); taking $t = \mathsf{f}(x_1, x_2, \mathsf{a}) \in \mathbb{L}$ and $\ell = \mathsf{f}(x, x, y)$ implies $t[t|_1]_2 = \mathsf{f}(x_1, x_1, \mathsf{a}) \in \mathbb{L}$, i.e., $x_1 \in \mathcal{V}_1 \cap \mathcal{V}_2 = \varnothing$.

In the remainder of this section we assume that $\mathbb{L}$ is a layer system.

▶ **Lemma 5.6.** *Let $\mathbb{L}$ be weakly consistent with $\mathcal{R}$. If $\hat{s} \to_{p,\ell\to r} \hat{t}$ is an outer rule step then we can designate $\hat{s}$ by $\sigma(s')$ so as to find $t' \in \mathbb{L}_{\mathcal{T}}$ such that $s' \to_{p,\ell\to r} t'$ and $\hat{t} = \sigma(t')$.*

Lemma 5.6 extends Lemma 4.12 to cover outer rewrite steps. It is a key ingredient for Lemma 5.7, which shows that we can map arbitrary rewrite steps on terms to layered terms.

▶ **Lemma 5.7.** *Let $\mathbb{L}$ be weakly consistent with $\mathcal{R}$.*
1. *If $\hat{s}$ is layered and $\hat{s} \to_{\mathcal{R}} \hat{t}$ then $\hat{t}$ is layered.*
2. *If $s \to_{p,\ell\to r} t$ then $\hat{s}_{\mathbb{L}} \to_{p,\ell\to r} \cdot \rightsquigarrow \hat{t}_{\mathbb{L}}$.*

In order to prove the key Lemma 5.10, we introduce the relation $\overset{\propto}{\to}{}_i^*$. Note that because $\propto$ and $\to_i^*$ are both reflexive and transitive, so is $\overset{\propto}{\to}{}_i^*$.

▶ **Definition 5.8.** Let $\vec{t} = (t_i)_{i\in I}$ and $\vec{u} = (u_i)_{i\in I}$ be vectors. We write $\vec{t} \propto \vec{u}$ if $t_i = t_j$ implies $u_i = u_j$ for all $i, j \in I$. If $\hat{s} = \hat{C}[\![\vec{s}]\!]$, $\hat{s} \to_i^* \hat{t}$ (hence $\hat{t} = \hat{C}[\![\vec{t}]\!]$ for a suitable $\vec{t}$) and $\vec{s} \propto \vec{t}$, we write $\hat{s} \overset{\propto}{\to}{}_i^* \hat{t}$.

▶ **Lemma 5.9.** *If $\to_{\mathcal{R}}$ is confluent on layered terms of rank less than n then for layered terms of rank at most n (1) ${}_i^*\!\leftarrow \cdot \to_i^* \subseteq \overset{\propto}{\to}{}_i^* \cdot {}_i^*\!\overset{\propto}{\leftarrow}$, and (2) ${}_o^*\!\leftarrow \cdot \overset{\propto}{\to}{}_i^* \subseteq \overset{\propto}{\to}{}_i^* \cdot {}_o^*\!\leftarrow$.*

▶ **Lemma 5.10.** *Let $\mathbb{L}$ be weakly consistent with $\mathcal{R}$. If $\to_{\mathcal{R}}$ is confluent on $\mathbb{L}_{\mathcal{T}}$ and $\to_o$ is confluent on layered terms then $\to_{\mathcal{R}}$ is confluent on layered terms.*

Lemma 5.10 is proved by induction on the rank using Lemma 5.9. It is used to conclude confluence of $\to_{\mathcal{R}}$ for both our main results, Theorems 5.13 and 5.27, which we develop in Sections 5.1 and 5.2 below.

5.1 Left-Linear Systems

In this section we deal with a left-linear TRS $\mathcal{R}$ and a layer system $\mathbb{L}$ weakly consistent with $\mathcal{R}$. We want to show that $\mathcal{R}$ is confluent if $\mathcal{R}$ is confluent on $\mathbb{L}_{\mathcal{T}}$.

▶ **Lemma 5.11.** *If $\to_{\mathcal{R}}$ is confluent on $\mathbb{L}_{\mathcal{T}}$ then $\to_o$ is confluent on layered terms.*

Consequently, $\to_{\mathcal{R}}$ is confluent on layered terms by Lemma 5.10. The next lemma deals with the interaction of rewriting and fusion steps.

▶ **Lemma 5.12.** *On layered terms $\leftsquigarrow \cdot \rightsquigarrow \subseteq \rightsquigarrow \cdot \leftsquigarrow$ and $\leftsquigarrow \cdot \to_{\mathcal{R}}^* \subseteq \to_{\mathcal{R}}^* \cdot \leftsquigarrow$.*

Combining Lemmata 5.7 (to map rewrite sequences to layered terms), 5.10, 5.11 and 5.12 (for confluence of rewriting layered terms), we obtain our main result for left-linear TRSs.

▶ **Theorem 5.13.** *Let $\mathcal{R}$ be a left-linear TRS and $\mathbb{L}$ a layer system that is weakly consistent with $\mathcal{R}$. If $\mathcal{R}$ is confluent on $\mathbb{L}_{\mathcal{T}}$ then $\mathcal{R}$ is confluent.*

The above result also holds for non-duplicating TRSs, but the proof is considerably more involved. It can be found in our technical report [5].

5.2 General Systems

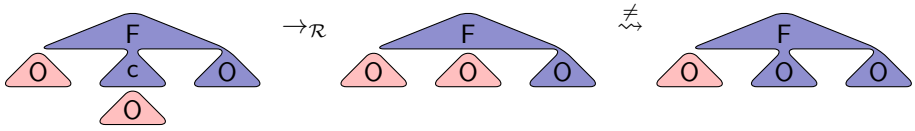
In this section we consider the case of general TRSs that may be non-left-linear. When this happens, we can conclude confluence of $\mathcal{R}$ if $\mathbb{L}$ is *consistent* with $\mathcal{R}$.

▶ **Definition 5.14.** A layer system $\mathbb{L}$ is called *consistent* with a TRS $\mathcal{R}$ if it is weakly consistent with $\mathcal{R}$ and conditions 3 and 4 hold.

3. Let $s, t \in \mathbb{L} \setminus (\mathcal{V} \cup \{\Box\})$ with $s \to_{p,\ell\to r} t$ and $q \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(\ell)$, $q' \in \mathcal{P}\mathsf{os}_{\mathcal{V}}(r)$ such that $\ell|_q = r|_{q'}$. Furthermore let $t' \in \mathbb{L}$ be obtained from t by replacing some holes by terms from $\mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{V})$. Then $s[t'|_{pq'}]_{pq} \in \mathbb{L}$.

4. Let $s, t \in \mathbb{L}$ with $s \to_{p,\ell\to r} t$ and $q \parallel p$. If $t|_q \in \mathcal{V} \cup \{\Box\}$ and $t[t']_q \in \mathbb{L}$ then $s[t']_q \in \mathbb{L}$.

Let us examine the effect of conditions 3 and 4 of Definition 5.14. In order to deal with non-left-linear systems we have to restrict layer systems further. One key step in our proof (and the original proof in [9]) is the construction of witnesses. It is interesting to see how this fails in the counterexample from Section 3, if we define layers to be order-sorted terms, which results in a weakly consistent layer system:



Here we have a fusion step that is enabled by a rewrite step above the layer being fused. This phenomenon, *fusion from above*, defeats any attempt to build a witness in a bottom up fashion as the proof does.

The additional constraints for consistency prevent fusion from above. They demand that any subterm that can be fused with a layer *after* a rewrite step inside that layer could already have been fused *before* the rewrite step.

▶ **Example 5.15.** In the counterexample from Section 3, if we let $\mathbb{L}$ be the set of order-sorted terms closed under replacing variables by holes, we have $\mathbb{L} \ni s = \mathsf{F}(\Box, \mathsf{c}(\Box), \mathsf{O}) \to_{2,\mathsf{c}(x)\to x} \mathsf{F}(\Box, \Box, \mathsf{O}) = t \in \mathbb{L}$ and $t' = \mathsf{F}(\Box, \mathsf{O}, \mathsf{O}) \in \mathbb{L}$. With $q = 1$, $q' = \epsilon$ we conclude from Definition 5.14(3) that $s[t'|_2]_{21} = \mathsf{F}(\Box, \mathsf{c}(\mathsf{O}), \mathsf{O}) \in \mathbb{L}$. Since this term is not order-sorted, we see that $\mathbb{L}$ is not consistent with the given TRS.

As the example shows, condition 3 of Definition 5.14 needs careful consideration when the TRS $\mathcal{R}$ has collapsing rules: If $s \to_{p,\ell\to r} t$, then the subterm $s|_{pq}$ replaces $s|_p$ completely in the reduct. That is, any subterm u that can occur in place of the left-hand side of the rule application, $\sigma(\ell) = s|_p$ (meaning that $s[u]_p \in \mathbb{L}$) must also be allowed at $s|_{pq}$, i.e., $s[u]_{pq} \in \mathbb{L}$.

From now on we assume that $\mathbb{L}$ is consistent with $\mathcal{R}$.

▶ **Lemma 5.16.** *For layered terms $\hat{s}$ and $\hat{t}$, if $\hat{s} \to_o \hat{t}$ then $\hat{s}_{\mathbb{L}} \to_o \hat{t}_{\mathbb{L}}$ or both $\hat{s}_{\mathbb{L}} \to_o \epsilon(\hat{t}_{\mathbb{L}})$ and* $\mathrm{rank}(\hat{s}_{\mathbb{L}}) > \mathrm{rank}(\hat{t}_{\mathbb{L}})$.

▶ **Lemma 5.17.** *If $\to_{\mathcal{R}}$ is confluent on $\mathbb{L}_{\mathcal{T}}$ then $\to_o$ is confluent on layered terms.*

Lemma 5.17 enables us to use Lemma 5.10 again, but we still have to deal with fusion steps. The remainder of this section is based on the *simplified proof* by Klop *et al.* [9].

▶ **Definition 5.18.** A term $\hat{t}$ is called *inner preserved* if $\hat{u}|_{\_} = \hat{u}'|_{\_}$ whenever $\hat{t} \to_{\mathcal{R}}^* \hat{u} \rightsquigarrow \hat{u}'$. The term $\hat{t}$ is *preserved* if $\hat{u} = \hat{u}'$ under the same condition.

Note that the principal subterms of an inner preserved term are inner preserved as well. On inner preserved terms, fusion steps can only affect the root special position, so do not interact with rewrite steps in any essential way. Hence from Lemmata 5.17 and 5.10 we have:

▶ **Lemma 5.19.** *The relation $\rightsquigarrow_{\mathcal{R}}$ is confluent on inner preserved terms.*

▶ **Definition 5.20.** A proper fusion step $\hat{a} \overset{\neq}{\rightsquigarrow} \hat{b}$ is *inner*, denoted by $\hat{a} \to_{\ni} \hat{b}$, if $\min(P_{\hat{a}}) = \min(P_{\hat{b}})$, i.e., only non-principal positions are removed. Let $\to_{i,\ni} = \to_i \cup \to_{\ni}$. We define $\hat{s} \overset{\propto}{\to}{}^*_{i,\ni} \hat{t}$ if $\hat{C}[\![\vec{s}]\!] = \hat{s} \to_{i,\ni}^* \hat{t} = \hat{C}[\![\vec{t}]\!]$ (the top context is not affected by $\to_{i,\ni}$) and $\vec{s} \propto \vec{t}$.

The relation $\overset{\propto}{\to}{}^*_{i,\ni}$ is different from $\overset{\propto}{\to}{}^*_i$ introduced earlier in that it includes inner fusion steps. The overall intention is the same: $\overset{\propto}{\to}{}^*_{i,\ni}$ only affects principal subterms of a term and rewrites equal principal subterms in the same way.

▶ **Lemma 5.21.** *Let the relation $\rightsquigarrow_{\mathcal{R}}$ be confluent on terms of rank less than n. If $\hat{s}$ has rank n and $\hat{t}\ {}^{*}_{\Subset,i}\!\leftarrow \hat{s} \rightarrow^{*}_{i,\Supset} \hat{u}$ then $\hat{t} \overset{\propto}{\rightarrow}{}^{*}_{i,\Supset} \cdot {}^{*}_{\Subset,i}\!\overset{\propto}{\leftarrow} \hat{u}$.*

▶ **Definition 5.22.** A *witness* of a term $\hat{s}$ is an inner preserved term $\hat{t}$ such that $\hat{s} \rightsquigarrow_i^* \hat{t}$, where $\rightsquigarrow_i = \rightsquigarrow \cup \rightarrow_i$.

We can apply Lemma 5.19 to witnesses. Lemma 5.24 states that witnesses always do exist. An important step in its proof is Lemma 5.23, which allows us to replace principal subterms of a term by preserved reducts.

▶ **Lemma 5.23.** *For every inner preserved term $\hat{t}$ there exists a preserved term $\hat{u}$ with $\hat{t} \rightsquigarrow^*_{\mathcal{R}} \hat{u}$.*

▶ **Lemma 5.24.** *If $\rightsquigarrow_{\mathcal{R}}$ is confluent on terms of rank less than n then every term $\hat{t}$ with $\mathrm{rank}(\hat{t}) \leqslant n$ has a witness $\dot{t}$.*

▶ **Lemma 5.25.** *Let $\rightsquigarrow_{\mathcal{R}}$ be confluent on terms $\hat{t}$ with $\mathrm{rank}(\hat{t}) < n$. If $\hat{s} \rightsquigarrow_{\mathcal{R}} \hat{t}$ and $\mathrm{rank}(\hat{s}) = n$ then $\dot{s} \rightsquigarrow^*_{\mathcal{R}} \cdot {}^*_{\mathcal{R}}\!\rightsquigarrow \dot{t}$.*

Therefore we can construct witnesses for all terms in a conversion between two layered terms by Lemma 5.24, then join consecutive witnesses using Lemma 5.25, and finally use Lemma 5.19 to join the witnesses of the outermost terms, proving Lemma 5.26.

▶ **Lemma 5.26.** *The relation $\rightsquigarrow_{\mathcal{R}}$ is confluent if $\mathcal{R}$ is confluent on $\mathbb{L}_{\mathcal{T}}$.*

Since by Lemma 5.7 we can map rewrite sequences to layered terms, Lemma 5.26 implies our main result for arbitrary TRSs.

▶ **Theorem 5.27.** *Let $\mathcal{R}$ be a TRS and $\mathbb{L}$ a layer system that is consistent with $\mathcal{R}$. If $\mathcal{R}$ is confluent on $\mathbb{L}_{\mathcal{T}}$ then $\mathcal{R}$ is confluent.*

6 Applications

In this section we present three applications of layer systems.

Layer-Preservation: Let $\mathcal{T}_X(\mathcal{F},\mathcal{V})$ denote the set of terms with root symbol from X. Let $\mathcal{C} := \mathcal{F}_1 \cap \mathcal{F}_2$, $\mathcal{D}_1 := \mathcal{F}_1 \setminus \mathcal{F}_2$ and $\mathcal{D}_2 := \mathcal{F}_2 \setminus \mathcal{F}_1$.

▶ **Theorem 6.1** (Ohlebusch [11])**.** *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be TRSs such that $\mathcal{R}_1 \subseteq \mathcal{T}(\mathcal{C},\mathcal{V})^2 \cup \mathcal{T}_{\mathcal{D}_1}(\mathcal{F}_1,\mathcal{V})^2$, $\mathcal{R}_2 \subseteq \mathcal{T}(\mathcal{C},\mathcal{V})^2 \cup \mathcal{T}_{\mathcal{D}_2}(\mathcal{F}_2,\mathcal{V})^2$ and $\mathcal{R}_1 \cap \mathcal{T}(\mathcal{C},\mathcal{V})^2 = \mathcal{R}_2 \cap \mathcal{T}(\mathcal{C},\mathcal{V})^2$. The union $\mathcal{R}_1 \cup \mathcal{R}_2$ is confluent if and only if $\mathcal{R}_1$ and $\mathcal{R}_2$ are confluent.*

Proof Sketch. Let $\mathbb{L} := \mathcal{T}(\mathcal{C} \cup \{\Box\},\mathcal{V}) \cup \mathcal{T}_{\mathcal{D}_1}(\mathcal{F}_1 \cup \{\Box\},\mathcal{V}) \cup \mathcal{T}_{\mathcal{D}_2}(\mathcal{F}_2 \cup \{\Box\},\mathcal{V})$. It is straightforward (but somewhat tedious) to verify that with the given constraints, this is a layer system that is consistent with $\mathcal{R}_1 \cup \mathcal{R}_2$. Confluence follows by Theorem 5.27. ◀

Order-Sorted Persistence: Note that many-sorted persistence [1] arises as a special case of Theorem 6.2 by making sorts mutually incomparable, which in turn entails Toyama's classical modularity result [13]. The conditions are easy to implement, as outlined in [15].

▶ **Theorem 6.2.** *Let $\mathcal{R}$ be an $(\mathcal{S},\succeq)$-order-sorted TRS. Assume the following conditions:*

1. *$\mathcal{R}$ is compatible with $\mathcal{S}$, i.e., for $\ell \to r \in \mathcal{R}$ the terms ℓ and r are order-sorted with variable occurrences strictly bound in ℓ and $\mathrm{sort}(\ell) \succeq \mathrm{sort}(r)$.*

2. *If $\mathcal{R}$ is non-left-linear then for $\ell \to r \in \mathcal{R}$, variable occurrences in r are strictly bound as well. Furthermore, for collapsing rules ($r \in \mathcal{V}$) the sort of r must be maximal.*

If $\mathcal{R}$ is confluent on order-sorted terms then $\mathcal{R}$ is confluent on all terms.

Proof Sketch. The proof idea is to use $\mathbb{L} = \mathcal{T}_{\mathcal{S}}(\mathcal{F}, \mathcal{V})$ as a layer system, after closing it under replacing variables by holes. Conditions (1) and (2) of Theorem 6.2 ensure weak consistency and consistency of $\mathbb{L}$ with $\mathcal{R}$, respectively. Condition (1) also makes $\mathbb{L}$ closed under (unsorted) rewriting, by only allowing variables x to be assigned terms t having $\mathrm{sort}(t) \preceq \mathrm{sort}(x)$ when matching rules. Theorems 5.13 and 5.27 conclude the proof. ◀

Actually, condition (2) in the above theorem can be weakened to non-left-linear and duplicating TRSs, see [5].

There are a couple of notable differences between Theorem 6.2 and [1, Theorem 4.12]. First, the maximality constraint on the sorts of collapsing rules in the general case is new, and indeed rules out the counterexample from Section 3. Second, we have weaker constraints on left-linear systems. Finally, we do not require the order on sorts to be well-founded. (This is needed in [1] because every step that weakens the sort of a special subterm is treated as destructive, while in our approach only actual fusion steps must be considered.)

Theorem 6.2 is strictly stronger than many-sorted persistence, as the next example shows.

▶ **Example 6.3** (adapted from [1]). Consider the TRS $\mathcal{R}$ consisting of the rewrite rules

$$1\colon \mathsf{f}(x,\mathsf{A}) \to \mathsf{G}(x) \qquad 2\colon \mathsf{f}(x,\mathsf{f}(x,\mathsf{B})) \to \mathsf{B} \qquad 3\colon \mathsf{G}(\mathsf{C}) \to \mathsf{C} \qquad 4\colon \mathsf{F}(x) \to \mathsf{F}(\mathsf{G}(x))$$

and the sorts $\mathcal{S} = \{0,1,2\}$ with $1 \succeq 0$ and the signature given by $\mathsf{A},\mathsf{B} : 1$, $\mathsf{C} : 0$, $\mathsf{F} : 0 \to 2$, $\mathsf{G} : 0 \to 0$, $\mathsf{f} : 0 \times 1 \to 1$ and $x : 0$. It is straightforward to check that the requirements of Theorem 6.2 are fulfilled. In the order-sorted TRS, only rules (1), (2) and (3) can be applied to terms of sort 1 and their derivatives, rules (3) and (4) can be applied to terms derived from terms of sort 2 and only rule (3) can be applied to terms of sort 0. Hence, since $\mathcal{R}_1 = \{(1),(2),(3)\}$ (which is terminating and has no critical pairs), $\mathcal{R}_2 = \{(3),(4)\}$ (which is orthogonal), and $\mathcal{R}_3 = \{(3)\}$ (orthogonal) are confluent, $\mathcal{R}$ is confluent. No such decomposition can be obtained with many-sorted persistence. Consider a *most general* signature making all rules many-sorted: $\mathsf{A},\mathsf{B},\mathsf{C},x : 0$, $\mathsf{F} : 0 \to 1$, $\mathsf{G} : 0 \to 0$, and $\mathsf{f} : 0 \times 0 \to 0$. Since terms of sort 1 can have subterms of sort 0, no decomposition is possible.

The weaker conditions in Theorem 6.2 for left-linear TRSs are beneficial.

▶ **Example 6.4.** Consider the TRS $\mathcal{R}$ consisting of the rewrite rules

$$\mathsf{c}(x) \to x \qquad \mathsf{f}(\mathsf{A}) \to \mathsf{f}(\mathsf{f}(\mathsf{c}(0))) \qquad \mathsf{g}(\mathsf{B}) \to \mathsf{g}(\mathsf{g}(\mathsf{c}(0)))$$

the sorts $\mathcal{S} = \{0,1,2\}$ with $1,2 \succeq 0$ and the signature given by $0, x : 0$, $\mathsf{c} : 0 \to 0$, $\mathsf{A} : 1$, $\mathsf{f} : 1 \to 1$, $\mathsf{B} : 2$ and $\mathsf{g} : 2 \to 2$. This satisfies the conditions of Theorem 6.2 for left-linear systems, and we can decompose the system into the component induced by sort 1: $\{\mathsf{c}(x) \to x, \mathsf{f}(\mathsf{A}) \to \mathsf{f}(\mathsf{f}(\mathsf{c}(0)))\}$ and sort 2: $\{\mathsf{c}(x) \to x, \mathsf{g}(\mathsf{B}) \to \mathsf{g}(\mathsf{g}(\mathsf{c}(0)))\}$. If we add the restrictions for non-left-linear systems, the collapsing rule $\mathsf{c}(x) \to x$ enforces $\mathsf{c} : \alpha \to \alpha$ for a maximal sort α. Hence also the argument of f and g has sort α, and $\alpha \succeq \mathrm{sort}(\mathsf{A}), \mathrm{sort}(\mathsf{B}), \mathrm{sort}(\mathsf{f}(x)), \mathrm{sort}(\mathsf{g}(x)), \mathrm{sort}(0)$. So the component induced by α contains all rules.

Currying: Currying is a transformation of TRSs that introduces partial applications. It is useful in the construction of polynomial-time procedures for deciding properties of TRSs as, for example, in [4]. Kahrs [7] proved that confluence is preserved by this transformation.

▶ **Definition 6.5.** Let $\mathcal{R}$ be a TRS over a signature $\langle \mathcal{F}, \mathcal{V} \rangle$. Let $\mathcal{F}' = \mathcal{F} \cup \{\mathsf{Ap}\}$ where Ap is a binary function symbol and all function symbols in $\mathcal{F}$ become constants. The *curried version* of $\mathcal{R}$ operates on $\mathcal{T}(\mathcal{F}', \mathcal{V})$ and is defined as $\mathsf{Cu}(\mathcal{R}) = \{\mathsf{Cu}(\ell) \to \mathsf{Cu}(r) \mid \ell \to r \in \mathcal{R}\}$. Here $\mathsf{Cu}(t) = t$ if t is a variable or a constant and $\mathsf{Cu}(f(t_1, \ldots, t_n)) = \mathsf{Ap}(\cdots \mathsf{Ap}(f, \mathsf{Cu}(t_1)) \cdots, \mathsf{Cu}(t_n))$ (with n occurrences of Ap). Let $\mathcal{F}'' = \mathcal{F}' \cup \{f_i \mid f \in \mathcal{F}, 0 \leqslant i < \mathrm{arity}(f)\}$, where f_i has arity i. The *partial parametrization* $\mathsf{PP}(\mathcal{R})$ of $\mathcal{R}$ is defined as the union of $\mathcal{R}$ and $\mathcal{U}$, where $\mathcal{U}$ consists of all *uncurrying* rules:

$$\mathsf{Ap}(f_i(x_1, \ldots, x_i), x_{i+1}) \to f_{i+1}(x_1, \ldots, x_{i+1})$$

for all $f \in \mathcal{F}$ and $0 \leqslant i < \mathrm{arity}(f)$, with the convention that $f_n = f$ if $n = \mathrm{arity}(f)$.

Note that $\to_{\mathcal{U}}$ is terminating and confluent (because $\mathcal{U}$ is orthogonal). Partial parametrization is closely related to currying: by [7, Proposition 3.1] and [8, Theorem 2.2], $\mathsf{PP}(\mathcal{R})$ is confluent if and only if $\mathsf{Cu}(\mathcal{R})$ is confluent.

▶ **Theorem 6.6.** *If $\mathcal{R}$ is confluent then $\mathsf{PP}(\mathcal{R})$ is confluent.*

Proof Sketch. Consider the term $t = \mathsf{Ap}(\cdots \mathsf{Ap}(f_i(t_1, \ldots, t_i), t_{i+1}) \cdots, t_m)$ where $f \in \mathcal{F}$, which is a curried function application of f with m arguments. Let t'_i be the $\to_{\mathcal{U}}$ normal form of t_i and $n = \mathrm{arity}(f)$. If $m < n$, the $\to_{\mathcal{U}}$ normal form of t will be $f_m(t'_1, \ldots, t'_m)$, and we call the function application *partial*. Otherwise, $m \geq n$, and the $\to_{\mathcal{U}}$ normal form of t is $\mathsf{Ap}(\cdots \mathsf{Ap}(f(t'_1, \ldots, t'_n), t'_{n+1}) \cdots, t'_m)$. We call $\mathsf{Ap}(\cdots \mathsf{Ap}(f_i(t_1, \ldots, t_i), t_{i+1}) \cdots, t_n)$ a *saturated* function application, and t_{n+1} to t_m *extra* arguments of f. We define $\mathbb{L} = \mathbb{L}_1 \cup \mathbb{L}_2 \cup \mathbb{L}_3$ where

- $\mathbb{L}_1 = \{t \mid t \to^*_{\mathcal{U}} u \text{ with } u \in \mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{V})\}$,
- $\mathbb{L}_2 = \{t \mid t \to^*_{\mathcal{U}} f_n(x_1, \ldots, x_n),\ f \in \mathcal{F},\ x_i \in \mathcal{V} \cup \{\Box\},\ n < \mathrm{arity}(f)\}$,
- $\mathbb{L}_3 = \{\mathsf{Ap}(x_1, x_2) \mid x_i \in \mathcal{V} \cup \{\Box\}\}$.

The idea is to start a new layer for any partial function application ($\mathbb{L}_2$), any Ap that binds extra arguments and thus remains in the $\to_{\mathcal{U}}$ normal form of t ($\mathbb{L}_3$), and for each extra argument, but not for any saturated function applications that appear as arguments of other saturated function applications ($\mathbb{L}_1$). It can be shown that $\mathbb{L}$ is indeed a layer system that is consistent with $\mathcal{R}$. ◀

7 Conclusion

In this paper we have presented an abstract layer framework that covers several known results about the modularity and persistence of confluence. The framework enabled us to correct the result claimed in [1] on order-sorted persistence, and, by weaker conditions for left-linear systems, to increase its applicability. We also showed how Kahrs' confluence result for curried systems is obtained as an instance of our layer framework. Furthermore, we have incorporated a decomposition technique due to Theorem 6.2 into CSI [15], our confluence prover.

As future work, we plan to investigate how to apply layer systems to other properties of TRSs, like termination or having unique normal forms. Since the applications in this paper use regular languages for the layer system $\mathbb{L}$, we also plan to investigate how the consistency conditions translate into restrictions on tree automata. Another interesting

question is whether van Oostrom's constructive modularity proof [14] can be adapted for layer systems. Finally, we worked out the technical details of our main results to prepare for future certification efforts in a theorem prover like Isabelle.

Acknowledgments

We thank the anonymous reviewers for their helpful and detailed comments.

References

1 T. Aoto and Y. Toyama. Extending persistency of confluence with ordered sorts. Technical Report IS-RR-96-0025F, School of Information Science, JAIST, 1996.

2 T. Aoto and Y. Toyama. Persistency of confluence. *Journal of Universal Computer Science*, 3(11):1134–1147, 1997.

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

4 H. Comon, G. Godoy, and R. Nieuwenhuis. The confluence of ground term rewrite systems is decidable in polynomial time. In *Proc. 42nd Annual Symposium on Foundations of Computer Science*, pages 298–307, 2001.

5 B. Felgenhauer, H. Zankl, and A. Middeldorp. Layer systems for proving confluence. Technical report, University of Innsbruck, 2011. Available at `http://cl-informatik.uibk.ac.at/software/csi/layerframework_report.pdf`.

6 J.-P. Jouannaud and Y. Toyama. Modular Church-Rosser modulo: The complete picture. *International Journal of Software and Informatics*, 2(1):61–75, 2008.

7 S. Kahrs. Confluence of curried term-rewriting systems. *Journal of Symbolic Computation*, 19(6):601–623, 1995.

8 R. Kennaway, J.W. Klop, M. Ronan Sleep, and F.-J. de Vries. Comparing curried and uncurried rewriting. *Journal of Symbolic Computation*, 21(1):15–39, 1996.

9 J.W. Klop, A. Middeldorp, Y. Toyama, and R. de Vrijer. Modularity of confluence: A simplified proof. *Information Processing Letters*, 49:101–109, 1994.

10 C. Lüth. Compositional term rewriting: An algebraic proof of Toyama's theorem. In *Proc. 7th International Conference on Rewriting Techniques and Applications*, volume 1103 of *Lecture Notes in Computer Science*, pages 261–275, 1996.

11 E. Ohlebusch. *Modular Properties of Composable Term Rewriting Systems*. PhD thesis, Universität Bielefeld, 1994.

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

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

14 V. van Oostrom. Modularity of confluence constructed. In *Proc. 4th International Joint Conference on Automated Reasoning*, volume 5195 of *Lecture Notes in Computer Science*, pages 348–363, 2008.

15 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.

16 H. Zantema. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50, 1994.