Theoretical Computer Science 464 (2012) 3–19

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Modularity in term rewriting revisited

Bernhard Gramlich

TU Wien, Austria

ARTICLE INFO

Keywords:
Term rewriting
Termination
Confluence
Modularity

ABSTRACT

We revisit modularity in term rewriting which for the last 25 years has been a very active and fruitful research field. Starting with the pioneering works of Yoshihito Toyama on the modularity of confluence and the non-modularity of termination he thus initiated an extremely productive line of research, with many non-trivial and deep results, striking counterexamples and a substantial amount of systematic theoretical foundations, methodological principles and novel proof techniques. In this focused summary we will revisit the modularity analysis for ordinary term rewriting systems, considering various confluence and termination properties and restricting ourselves mainly to the case of disjoint unions. We will summarize known results on the (non-) modularity of various confluence and termination properties, and exhibit crucial ideas, constructions and phenomena. Later on we will also briefly consider various extensions, applications, related questions and open problems, as well as recent developments.

1. Introduction and overview

We revisit modularity in term rewriting, a field that has developed a lot since the seminal works of Yoshihito Toyama more than two decades ago. The focus will be on modularity of confluence and termination properties of (first-order) term rewriting systems (TRSs). Later we will also consider directions for extending the analysis, by weakening the disjointness requirement, looking at further properties and aspects to be investigated w.r.t. modularity, and taking into account other types of rewrite systems. We will summarize what is known so far, highlight some historic milestones, discuss basic ideas, crucial notions and concepts, main proof techniques, fundamental constructions and the main sources of difficulties in modularity analysis. Furthermore we will briefly discuss the impact of research and results in modularity on other fields and vice versa, and mention some open problems and research challenges. Since the field of modularity in (term) rewriting has become enormously diverse and rich, any such summary has to concentrate on certain aspects and questions and is thus highly incomplete and also subjective to a certain degree. In order to complete the picture a bit, we try to mention at least those aspects and subfields where modularity is also important and studied, but where due to lack of space we do not go into details. Instead, we try to give lots of pointers to the abundant literature on the respective subjects.

Let us be more concrete now and begin with something positive. In [82] Toyama proved that confluence is a modular property of disjoint TRSs whereas termination is not. The former celebrated result is fairly non-trivial, since termination of the disjoint union of two terminating systems cannot be assumed, due to the non-modularity of termination. The famous counterexample Toyama discovered is the following. [1]

E-mail address: gramlich@logic.at.

[1] In an inspiring invited talk at RTA 2005 in Nara, Japan [84] Toyama told a bit about the history of these early days of modularity. He displayed a copy of the coverpage of a manuscript where he had ‘proved’ the modularity of termination. When trying to complete this proof he realized that there was a non-repairable gap — and shortly afterwards he got the inspiration for the counterexample when crossing a red traffic light ....

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.09.008

Example 1. Consider the two disjoint TRSs given by

$$\mathcal{R}_1 = \{f(a,b,x) \to f(x,x,x)\} \qquad \mathcal{R}_2 = \left\{ \begin{matrix} G(x,y) \to x \\ G(x,y) \to y \end{matrix} \right\}.$$

Both systems are terminating, but their disjoint union $\mathcal{R}$ is not as witnessed by the cyclic reduction

$$f(a,b,G(a,b)) \to_{\mathcal{R}_1} f(G(a,b),G(a,b),G(a,b)) \to^{+}_{\mathcal{R}_2} f(a,b,G(a,b)) \to_{\mathcal{R}_1} \cdots .$$

This remarkably simple, beautiful and ingenious (counter)example is probably the most cited and used (combined) TRS in the literature on term rewriting since. Variations thereof have been designed and exploited in many different contexts and for many different purposes.

In our survey we will focus on modularity of properties of TRSs w.r.t. disjoint unions. Note that in general the question of whether a property of TRSs or ARSs also holds for their union and vice versa is hopeless for most interesting cases. The ARSs (or TRSs) given by $a \to b$ and $b \to a$, respectively, are both terminating, but their union is not. Similarly, $a \to b$ and $a \to c$ together yield a non-confluent system, whereas the individual one-rule systems are obviously confluent. Thus, without further restrictions the hope for obtaining non-trivial modularity results appears not justified. A drastic way of restricting the interaction between two TRSs in their union is to separate them syntactically as much as possible, by requiring that the respective signatures are disjoint. Once this case of modularity w.r.t. disjoint unions has been well understood, the disjointness requirement can be weakened, considering more general combinations of systems that are (at most) *constructor sharing*, *composable* or *hierarchically* structured. We will focus on the disjoint union case, because for most of the results and approaches for more general types of combinations developed later on the understanding, the basic ideas and proof techniques of the disjoint union case are fundamental. In some detail cases, but selectively, we will consider combinations of TRSs with shared constructors in order to illustrate which new sources of problems occur in this setting and how results from the disjoint union case can be extended by adequately taking into account these new problems.

The properties of TRSs we will investigate for their (non-)modularity are mainly confluence and termination properties, in various versions. Rather than repeating proofs from the literature we will concentrate on main ideas, crucial observations, proof techniques, fundamental constructions and analogies. Many positive modularity results have a symmetric and an asymmetric version which we will sometimes mention. This is related to a phenomenon called *vaccination* [52] where the idea is that some property becomes modular provided one of the involved TRSs in a (disjoint) union or both of them have been *vaccinated* against the danger of non-modularity, i.e. enjoy some additional property implying modularity. For the disjoint union case we will summarize the state-of-the-art concerning modularity analysis.

An interesting property similar to modularity is *persistency* (of properties) which is defined via *many-sorted* TRSs and states that a property holds for a many-sorted TRS iff it holds for the unsorted version of the system. We will also briefly summarize the most important results and open questions in this area.

Extensions of the basic modularity theory and results for more general types of combinations will only be touched, mainly for combining constructor sharing TRSs. This also applies to combinations/decompositions of systems which are not primarily specified syntactically, but using other properties resulting in certain non-disjoint combinations of systems. Nevertheless we will give numerous pointers to the relevant literature on these extensions and other versions.

Similarly, as another dimension of the modularity analysis, instead of basic TRSs also many other types of rewrite systems have been investigated for their modularity behaviour: higher-order rewrite systems (HORs), simply typed term rewriting systems (STTRSs), term graph rewriting systems, infinitary rewriting systems, conditional TRSs (CTRSs), equational term rewriting systems (i.e. where rewriting happens modulo some underlying equational theory), rewrite systems with restricted reduction relation (via strategies, contextual restrictions, forbidden patterns). We will also only briefly touch these fields, but again try to give pointers to the relevant literature.

In the paper we concentrate on confluence and termination properties, but clearly there also exist modularity results about other properties and aspects like notions related to sequentiality and neededness, complexity and constructivity. Again we only refer to the literature here.

The remainder of the paper is structured as follows. After some preliminaries in Section 2 (which we recommend the informed reader to consult only by need), in the main part (Sections 4 and 5 we present the analysis and summary of the (non-)modularity of confluence and termination properties of TRS. In the discussion in Section 7 we then reason about methodological benefits (from modularity research), applications and implications of modularity, difficulties and (some) open problems as well as recent progress and emerging trends.

Much of this summary is highly incomplete, due to the diversity, breadth and depth of the field. This applies even to the main Sections 4 and 5 where we tried to cite all relevant papers in the area. But to some degree this list of references is still certainly subjective and partially incomplete, due to the existence of papers that the author is not aware of.

2. Preliminaries

We assume familiarity with the basic theory, notions and notations of abstract and term rewriting (cf. e.g. [6,9]), and for modularity (cf. e.g. [67]). Nevertheless, for the sake of readability we repeat here some of these notions and notations, introduce acronyms for various properties and provide some basic well-known facts.

2.1. Abstract reduction systems

An *abstract reduction system (ARS)* is a pair $\mathcal{A} = \langle A, \to \rangle$ consisting of a set A and a reduction (or rewrite) relation, i.e., a binary relation $\to \subseteq A \times A$ for which we use infix notation. [2] A *reduction sequence* or *derivation* (in $\mathcal{A}$) is a (finite or infinite) sequence $a_1 \to a_2 \to a_3 \to \cdots$. For $b \to a$ we also write $a \leftarrow b$. The symmetric, transitive, transitive–reflexive and symmetric–transitive–reflexive closures of $\to$ are denoted by $\leftrightarrow$, $\to^+$, $\to^*$ and $\leftrightarrow^*$, respectively. If $a \to^* b$ we say that a *reduces* or *rewrites* to b and we call b a *reduct* of a. By $a \to^m b$ we mean that a reduces to b in m steps. Accordingly $a \to^{\leq n} b$ means $a \to^m b$ for some $m \leq n$. Two elements $a, b \in A$ are *joinable*, denoted by $a \downarrow b$, if there exists an element $c \in A$ with $a \to^* c \leftarrow^* b$. An element $a \in A$ is *irreducible* or *a normal form* if there is no $b \in A$ with $a \to b$. An element $a \in A$ *has a normal form* if there exists a normal form $b \in A$ with $a \to^* b$. In that case b is called a *normal form of* a. The set of all normal forms of $\mathcal{A}$ is denoted by $\mathrm{NF}(\mathcal{A})$ or simply $\mathrm{NF}(\to)$ or NF when $\mathcal{A}$ is clear.

$\mathcal{A} = \langle A, \to \rangle$ is said to be *weakly normalizing* (or *weakly terminating*) (WN) if every element of A has a normal form. $\mathcal{A}$ is *strongly normalizing* or *terminating* (SN) if there is no infinite reduction sequence $a_1 \to a_2 \to a_3 \to \cdots$, i.e., of every reduction sequence eventually ends in some normal form. $\mathcal{A}$ is *confluent* or *Church-Rosser* (CR) [3] if for all $a, b, c \in A$ with $b \leftarrow^* a \to^* c$ we have $b \downarrow c$. $\mathcal{A}$ is *locally confluent* or *weakly Church-Rosser (WCR)* if for all $a, b, c \in A$ with $b \leftarrow a \to c$ we have $b \downarrow c$. A confluent and strongly normalizing system is said to be *convergent* or *complete*.

$\mathcal{A}$ *has the normal form property* (NF) if for all $a, b \in A$ with $a \leftrightarrow^* b$ and $b \in \mathrm{NF}(\mathcal{A})$ we have $a \to^* b$. $\mathcal{A}$ *has unique normal forms* (UN) if for all $a, b \in A$ with $a \leftrightarrow^* b$ and $a, b \in \mathrm{NF}(\mathcal{A})$ we have $a = b$. $\mathcal{A}$ *has unique normal forms w.r.t. reduction* ($\mathrm{UN}^{\to}$) if for all $a, b, c \in A$ with $a \leftarrow^* b \to^* c$ and $a, c \in \mathrm{NF}(\mathcal{A})$ we have $a = c$.

If an ARS $\mathcal{A} = \langle A, \to \rangle$ has a certain property P (denoted by $P(\mathcal{A})$), we also say that $\to$ has the property P (and also write $P(\to)$). This notation is extended in a straightforward manner to local versions, i.e. corresponding properties of single elements of $\mathcal{A}$, provided there is a sensible local interpretation of P. In order to make clear the underlying reduction relation $\to$ we shall also write $P(a, \to)$ instead of $P(a)$ (for $a \in A$).

2.2. Term rewriting systems

Terms are built over a signature $\mathcal{F}$ of function symbols (with fixed arities; the arity of $f \in \mathcal{F}$ is denoted by $ar(f)$) and a countably infinite set $\mathcal{V}$ of variables. The set of all terms over $\mathcal{F}$ is denoted by $\mathcal{T}(\mathcal{F}, \mathcal{V})$, the set of all ground terms by $\mathcal{T}(\mathcal{F})$. In the following we shall always assume that $\mathcal{T}(\mathcal{F})$ is non-empty, i.e., there is at least one constant in $\mathcal{F}$. The set of variables occurring in a term t is denoted by $Var(t)$.

A *context* $C[, \ldots,]$ is a term with 'holes', i.e. a term in $\mathcal{T}(\mathcal{F} \uplus \{\Box\}, \mathcal{V})$ (the symbol '$\uplus$' denotes disjoint set union) where $\Box$ is a new special constant symbol. If $C[, \ldots,]$ is a context with n occurrences of $\Box$ and $t_1, \ldots, t_n$ are terms then $C[t_1, \ldots, t_n]$ is the term obtained from $C[, \ldots,]$ by replacing from left to right the occurrences of $\Box$ by $t_1, \ldots, t_n$. A context containing precisely one occurrence of $\Box$ is denoted by $C[]$. A non-empty context is a context different from $\Box$.

A *term rewriting system (TRS)* is a pair $\mathcal{R} = (\mathcal{F}, R)$ consisting of a signature $\mathcal{F}$ and a set $R \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V}) \times \mathcal{T}(\mathcal{F}, \mathcal{V})$ of (rewrite) rules (l, r), denoted by $l \to r$, with $l \notin \mathcal{V}$ and $V(r) \subseteq V(l)$. This restriction of excluding variable left-hand sides and right-hand side extra-variables is not a severe one. In particular, concerning termination of rewriting it only excludes trivial cases. Instead of $\mathcal{R} = (\mathcal{F}, R)$ we also write $\mathcal{R}^{\mathcal{F}}$ or simply $\mathcal{R}$ or R when $\mathcal{F}$ is clear from the context or irrelevant. For the left- and right-hand side of a rewrite rule we also write lhs and rhs, respectively. The rewrite or reduction relation induced by a TRS $\mathcal{R} = (\mathcal{F}, R)$ is denoted by $\to_{\mathcal{R}}$ or just $\to$ if $\mathcal{R}$ is clear from the context. We also flexibly decorate the arrow by further information when needed. For instance, $s \to_p t$ indicates the position p of the redex $s|_p$ in s contracted in the step from s to t. Furthermore, we write $s \to_i t$ ($s \to_o t$) if $s \to t$ by contracting an innermost (outermost) redex in s. A TRS $\mathcal{R} = (\mathcal{F}, R)$ can be viewed as an ARS $\langle \mathcal{T}(\mathcal{F}, \mathcal{V}), \to_{\mathcal{R}} \rangle$ with more syntactic structure where the reduction relation is induced by the rewrite rules. If $\mathcal{R}^{\mathcal{F}}$ is a TRS with $\mathcal{F} = \mathcal{C} \uplus \mathcal{D}$ such that $\mathcal{D} = \{root(l) \mid l \to r \in \mathcal{R}^{\mathcal{F}}\}$ and $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$, we say that the symbols in $\mathcal{D}$ are the *defined (function) symbols* (of $\mathcal{R}^{\mathcal{F}}$) and those in $\mathcal{C}$ the *constructors*. Terms in $\mathcal{T}(\mathcal{C}, \mathcal{V})$ are called *constructor terms*. $\mathcal{R}^{\mathcal{F}}$ is said to be a *constructor system* (CS) if for every rule all arguments of all lhs's are constructor terms, i.e., if no left-hand side argument contains a defined symbol. [4]

We say that a TRS $\mathcal{R}$ is *(inter)reduced* (RED) if (a) r is $\mathcal{R}$-irreducible for every rule $l \to r \in \mathcal{R}$, and (b) no lhs l, for some $l \to r \in \mathcal{R}$, is $\mathcal{R} \setminus \{l \to r\}$-reducible. [5]

For common well-known syntactic properties of (rewrite rules and) TRSs we use the following abbreviations: left-linear (LL), right-linear (RL), *non-collapsing* (NCOL) – i.e., no rhs of a rule is a variable, *non-duplicating* (NDUP) – i.e., no variable occurs (strictly) more often in a rhs than in the lhs of a rule, *non-erasing* (NE) [6] — i.e., every variable of the lhs also appears in the rhs, *non-overlapping* (NO) — i.e., there are no *critical pairs*, *orthogonal* (ORTH) — i.e., non-overlapping plus left-linear,

[2] For our purposes this simple version of ARS is sufficient, whereas for other purposes versions with more structure like indexed families of reductions are appropriate, cf. [9].

[3] or *has the Church-Rosser property*

[4] This definition of constructor system corresponds to what is usually meant when one speaks of a constructor discipline (for specifying functions).

[5] Of course, equality of rules is meant modulo renaming of variables.

[6] In the literature, instead of *non-erasing* sometimes also *variable-preserving* is used.

weakly non-overlapping (WNO) — i.e., all critical pairs $\langle s, t\rangle$ are trivial (that is $s = t$), *weakly orthogonal* (WORTH) — weakly non-overlapping and left-linear, *overlaying* or being an *overlay system* (OS) — i.e., all critical pairs are critical *overlays* (in other words, there is no critical overlap below the root of rules), being a *constructor system* (CS) — i.e., all lhs's l are constructor-based, that is $l = f(s_1, \dots, s_n)$ with $f \in \mathcal{D}$ and $s_i \in \mathcal{T}(C, V)$ for $1 \le i \le n$, [7] *strongly non-overlapping* (SNO) — i.e., non-overlapping even after linearizing all lhs's of the rules.

3. Modularity: some notions, notations and ideas

The following notations and definitions for dealing with disjoint unions of TRSs mainly follow [82] and [55].

Let $\mathcal{R}_1^{\mathcal{F}_1}$, $\mathcal{R}_2^{\mathcal{F}_2}$ be TRSs with disjoint signatures $\mathcal{F}_1$, $\mathcal{F}_2$. Their *disjoint union* $\mathcal{R}^{\mathcal{F}}$ is the TRS $(\mathcal{F}, R)$ with $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2$, $R = R_1 \uplus R_2$. A property $\mathcal{P}$ of TRSs is said to be *modular* if for all disjoint TRSs $\mathcal{R}_1^{\mathcal{F}_1}$, $\mathcal{R}_2^{\mathcal{F}_2}$ the following holds: $\mathcal{R}^{\mathcal{F}}$ has property $\mathcal{P}$ iff both $\mathcal{R}_1^{\mathcal{F}_1}$ and $\mathcal{R}_2^{\mathcal{F}_2}$ have property $\mathcal{P}$. Note that for proving modularity in most interesting cases the 'only-if' part is trivial, whereas the 'if-part' is the real problem.

Let $t = C[t_1, \dots, t_n]$, $n \ge 1$, with $C[, \dots,] \ne \Box$. We write $t = C[\![t_1, \dots, t_n]\!]$ if $C[, \dots,]$ is a context over the signature $\mathcal{F}_a$ and $root(t_1), \dots, root(t_n) \in \mathcal{F}_b$ for some $a, b \in \{1, 2\}$ with $a \ne b$. In this case the t_i's are the *principal subterms* or *principal aliens* of t. Note that every $t \in \mathcal{T}(\mathcal{F}_1 \uplus \mathcal{F}_2, \mathcal{V}) \setminus (\mathcal{T}(\mathcal{F}_1, \mathcal{V}) \cup \mathcal{T}(\mathcal{F}_2, \mathcal{V}))$ has a unique representation of the form $t = C[\![t_1, \dots, t_n]\!]$. The set of all *aliens* (or *special subterms*) of t can be recursively defined in an obvious manner.

The *rank* of a term $t \in \mathcal{T}(\mathcal{F}_1 \uplus \mathcal{F}_2, \mathcal{V})$ is defined by

$$rank(t) = \begin{cases} 1 & \text{if } t \in \mathcal{T}(\mathcal{F}_1, \mathcal{V}) \cup \mathcal{T}(\mathcal{F}_2, \mathcal{V}) \\ 1 + max\{rank(t_i) | 1 \le i \le n\} & \text{if } t = C[\![t_1, \dots, t_n]\!]. \end{cases}$$

An important basic fact about the rank of terms occurring in a $(\mathcal{R}^{\mathcal{F}})$-derivation is the following [82]: If $s \to^* t$ then $rank(s) \ge rank(t)$. Moreover, if $s \in \mathcal{T}(\mathcal{F}_1 \uplus \mathcal{F}_2, \mathcal{V})$ with $rank(s) = n$ then there exists a ground instance $s\sigma$ of s with $rank(s\sigma) = n$, too. This is easily verified by substituting appropriately $\mathcal{F}_1$- or $\mathcal{F}_2$-ground terms for all variables in s. Appropriately means that for all variables in deepest aliens an $\mathcal{F}_i$-ground term should be chosen whenever the alien is an $\mathcal{F}_i$-alien. For all other variables (appearing in non-deepest aliens) any $\mathcal{F}_1$- or $\mathcal{F}_2$-ground term can be substituted. A (finite or infinite) derivation $D: s_1 \to s_2 \to s_3 \cdots$ is said to have rank n ($rank(D) = n$) if n is the minimal rank of all the s_i's, i.e. $n = \min\{rank(s_i) | 1 \le i\}$.

The topmost homogeneous part of a term $t \in \mathcal{T}(\mathcal{F}_1 \uplus \mathcal{F}_2, \mathcal{V})$, denoted by $top(t)$, is obtained from t by replacing all principal subterms by $\Box$, i.e.

$$top(t) = \begin{cases} t & \text{if } rank(t) \le 1 \\ C[, \dots,] & \text{if } t = C[\![t_1, \dots, t_n]\!]. \end{cases}$$

For the sake of better readability the function symbols from $\mathcal{F}_1$ are considered to be black and those of $\mathcal{F}_2$ to be white. Variables have no colour. A *top black (white)* term has a black (white) root symbol.

The one-step reduction $s \to t$ in a disjoint union is said to be *inner*, denoted by $s \xrightarrow{i} t$, if the reduction takes place in one of the principal subterms of s. Otherwise, we speak of an *outer* reduction step and write $s \xrightarrow{o} t$. A rewrite step $s \to t$ is *destructive at level 1* if the root symbols of s and t have different colours. The step $s \to t$ is *destructive at level* $n + 1$ (for $n \ge 1$) if $s = C[\![s_1, \dots, s_j, \dots, s_n]\!] \xrightarrow{i} C[s_1, \dots, t_j, \dots, s_n] = t$ with $s_j \to t_j$ destructive at level n. Clearly, if a rewrite step is destructive (at some level) then the applied rewrite rule is collapsing, i.e., has a variable as right-hand side. This is a basic fact which should be kept in mind subsequently.

For encoding/abstracting principal subterms, e.g., by new variables or constants, and for dealing with outer rewrite steps involving non-linear rules the following definitions are useful. For terms $s_1, \dots, s_n, t_1, \dots, t_n$ we write $\langle s_1, \dots, s_n\rangle \propto \langle t_1, \dots, t_n\rangle$ if $t_i = t_j$ whenever $s_i = s_j$, for all $1 \le i < j \le n$. Symmetrically, we write $\langle s_1, \dots, s_n\rangle \infty \langle t_1, \dots, t_n\rangle$ if both $\langle s_1, \dots, s_n\rangle \propto \langle t_1, \dots, t_n\rangle$ and $\langle t_1, \dots, t_n\rangle \propto \langle s_1, \dots, s_n\rangle$. In many situations, especially in proofs (of modularity results), one of the following two *transformation by abstraction constructions* turns out to be useful and applicable. Suppose $\mathcal{R}^{\mathcal{F}}$ is the disjoint union of the TRSs $\mathcal{R}_1^{\mathcal{F}_1}$, $\mathcal{R}_2^{\mathcal{F}_2}$, and consider a mixed term $s = C[\![s_1, \dots, s_n]\!]$ and a (finite or infinite) derivation $D: s = C[\![s_1, \dots, s_n]\!] \to_{\mathcal{R}} \cdots$.

Identifying abstraction. We abstract away the principal aliens s_i in $s = C[\![s_1, \dots, s_n]\!]$ (ignoring their concrete form) by replacing them all by the same fresh variable x yielding $\hat{s} = C[\![\widehat{s_1, \dots, s_n}]\!] = C[x, \dots, x]$, and analogously for D yielding $\widehat{D}$. [8]

[7] Note that every *constructor system* is by definition also an *overlay system*.

[8] This *identifying abstraction* is like taking the topmost homogeneous part $top(t)$ of a mixed term t, except for the fact that in the latter we use a fresh constant ($\Box$) instead of a fresh variable. Technically, identifying abstraction is preferable, because there no signature extension takes place.

Injective abstraction. We abstract the principal aliens s_i in $s = C[\![s_1, \ldots, s_n]\!]$ or in D: $s = C[\![s_1, \ldots, s_n]\!] \to_{\mathcal{R}} \ldots$, respectively, by replacing them by fresh variables $x_1, \ldots, x_n$ in such a way into that $\langle s_1, \ldots, s_n\rangle \infty \langle x_1, \ldots, x_n\rangle$ holds, i.e., only identical principal aliens are abstracted by the same fresh variable, whereas for the other ones we need mutually distinct fresh variables, yielding $\tilde{s} = C[\![\widetilde{s_1, \ldots, s_n}]\!] = C[x_1, \ldots, x_n]$, and analogously for D yielding $\widetilde{D}$.

Unfortunately, modularity of properties of TRSs is in general neither closed by strengthening nor by weakening, i.e., for properties $\mathcal{P}$, $\mathcal{Q}$ of TRSs,

- if $\mathcal{P}(\mathcal{R}^{\mathcal{F}}) \Longrightarrow \mathcal{Q}(\mathcal{R}^{\mathcal{F}})$ and $\mathcal{Q}$ is modular, then $\mathcal{P}$ need not be modular, and
- if $\mathcal{P}(\mathcal{R}^{\mathcal{F}}) \Longrightarrow \mathcal{Q}(\mathcal{R}^{\mathcal{F}})$ and $\mathcal{P}$ is modular, then $\mathcal{Q}$ need not be modular.

Counterexamples for the two above non-closure properties above is obtained e.g. by choosing P = SN, Q = WN and P = CR, Q = GCR: Clearly termination (SN) implies weak termination (WN), and WN is modular, whereas SN is not (see Example 1). Confluence (CR) is modular (see Section 4 and implies *ground confluence* [9] (GCR), but clearly ground confluence is not modular (since it is an inductive property that depends on the signature). For instance, $\mathcal{R}^{\mathcal{F}}$ consisting of the rules $f(x) \to x, f(x) \to a$ over the signature $\mathcal{F} = \{a, f\}$ is easily seen to be GCR (and not CR), since every ground term reduces to a, whereas it becomes non-GCR when we add a new constant symbol b: $a \leftarrow f(b) \to b$ with a and b normal forms that are not joinable any more. Thus, GCR is not even preserved under signature extensions, a very special case of modularity.

However, when trying to find subclasses of TRSs where some in general non-modular property $\mathcal{P}$ of TRSs is modular, one may proceed in a modular way as follows.

- Find a related property $\mathcal{Q}$ that is modular, with $\mathcal{Q} \Longrightarrow \mathcal{P}$.
- Find equivalence conditions $\mathcal{M}$ for $\mathcal{P}$ and $\mathcal{Q}$, i.e., $\mathcal{M}$ with $(*)$ $[\ \mathcal{M} \Longrightarrow [\ \mathcal{Q} \iff \mathcal{P}\ ]\ ]$ such that $\mathcal{M}$ is modular, too.
- If this search is successful, then $\mathcal{P}$ is modular for TRSs satisfying $\mathcal{M}$ (more precisely, $\mathcal{P}$ & $\mathcal{M}$ is modular, due to modularity of $\mathcal{M}$ and of $\mathcal{Q}$, and because of $(*)$.

We will study now concretely the modularity behaviour of the most fundamental confluence and termination properties of TRSs. In the following, modularity always refers to disjoint unions if not otherwise stated. $\mathcal{R}_1$, $\mathcal{R}_2$ or ${\mathcal{R}_1}^{\mathcal{F}_1}$, ${\mathcal{R}_2}^{\mathcal{F}_2}$ are always disjoint TRSs with $\mathcal{R}$ or $\mathcal{R}^{\mathcal{F}}$ being their disjoint union. First let us collect obvious modularity properties concerning the syntactic shape of TRSs which are often implicitly used.

Fact 2. *The properties LL, RL, NDUP, NCOL, NE, NO, ORTH, WNO, WORTH, SNO, OS and CS are all modular.*

4. Modularity analysis of confluence and related properties

First we will consider ordinary confluence. Later we shall also consider weaker and related versions of confluence.

4.1. Confluence

Local confluence is modular, since it is well-known to be equivalent to joinability of all critical pairs (CP($\mathcal{R}$)) of the considered system $\mathcal{R}$. But the critical pair construction is modular (due to the disjoint signatures): CP($\mathcal{R}$) = CP($\mathcal{R}_1$) $\cup$ CP($\mathcal{R}_2$), and also the property *joinability of all critical pairs*, since the rules from $\mathcal{R}$ that can possibly be used to join the critical pairs in CP($\mathcal{R}_i$), $i \in \{1, 2\}$, are exactly the ones from $\mathcal{R}_i$. Joinability of all critical pairs of $\mathcal{R}$ is well-known to be equivalent to local confluence of $\mathcal{R}$, hence it follows that WCR is modular. By *Newman's Lemma* we know that confluence is equivalent to local confluence for terminating systems (SN $\Longrightarrow$ [WCR $\iff$ CR]). Thus, suppose we have two terminating and confluent disjoint TRSs $\mathcal{R}_1^{\mathcal{F}_1}$, $\mathcal{R}_2^{\mathcal{F}_2}$, then their disjoint union $\mathcal{R}^{\mathcal{F}}$ will be locally confluent and confluent provided it is terminating. Unfortunately the assumption of termination is in general not justified here, due to Toyama's basic Counterexample 1. Moreover, we cannot appeal to similar knowledge for the case that termination of the $\mathcal{R}_i$ is not given or not known.

Theorem 3 (*[82]*). *Confluence is a modular property of TRSs.*

The original proof of Theorem 3 from 1987 is fairly involved, and quite non-trivial, due to the problems mentioned above. Since there is no termination assumption, any inductive proof cannot rely on a reduction ordering for the rewrite relation(s) involved. Technically, what causes most problems is that the layer structure of terms is not stable during reduction, i.e., it may happen that one layer collapses (disappears) thus modifying the layer above substantially, in such a way that new rewrite steps get enabled which were not possible before. This phenomenon can already be seen in Toyama's Counterexample 1 where, after the first step, no rule in the outermost layer is applicable, but is made possible by two collapsing steps in the principal aliens.

Seven years later, a substantially simplified proof of Theorem 3 was published in [34]. It has a clear structure, works by induction on rank, first shows that inner preserved terms are confluent, and is based on a few principles and ideas which are partially already implicitly used in [82], but hidden.

[9] A TRS $\mathcal{R}^{\mathcal{F}}$ is *ground confluent* if $\mathcal{R}^{\mathcal{F}}$ is confluent on every ground term $s \in \mathcal{T}(\mathcal{F})$.

(1) In the disjoint union every term s has a *preserved reduct*. Here a term is *preserved* if there are no reduction sequences starting from it that contain a destructive step. A term is *inner preserved* if all its principal subterms are preserved.
(2) Preserved reducts of a term are exactly the normal forms of the so-called *collapsing reduction* relation $\to_c$, defined by $s \to_c t$ if for some context $C[]$, $s = C[s_1]$, $t = C[t_1]$ where s_1 is a special subterm of s_1, $s_1 \to^* t_1$ and t_1 is a variable or the root symbols of s_1 and t_1 have different colours.
(3) Outer reduction in $\mathcal{R}_i$ is confluent, for $i \in \{1, 2\}$.
(4) Every term $s = C[s_1, \ldots, s_n]$ has a witness, i.e., an inner preserved term $t = C[t_1, \ldots, t_n]$ with $s_i \to^* t_i$ for $1 \le i \le n$, and $\langle s_1, \ldots, s_n\rangle \propto \langle t_1, \ldots, t_n\rangle$.
(5) Every finite set of confluent terms can be *represented*. Here, for a set S of confluent terms, a set $\hat{S}$ *represents* S, if
   (a) every term $s \in S$ has a unique reduct $\hat{s}$ in $\hat{S}$, called the *representative* of s, and
   (b) joinable terms in S have the same representative in $\hat{S}$.

(1) The actual existence of preserved reducts is the only property which is really specific for the disjoint union case. Here, due to (2), it is obvious that a preserved reduct always exists. Actually $\to_c$ is even SN. (3) is easy and intuitive, via injective abstraction of the principal aliens. (5) is generally true and provides a way how to find common successors of joinable confluent terms. (4) is a straightforward synchronization property stating that when constructing preserved reducts of (parallel) principal subterms, this can be done such that if two initial principal subterms are the same, then also their constructed preserved reducts are the same.

4.1.1. Extensions

Building on the above extended and abstracted analysis for the modularity of confluence, several authors have investigated further aspects and extensions of this analysis, in order to better understand the essence of the modularity of confluence.

In [49] a category theoretical approach to modularity of confluence is developed. For this purpose a compositional semantics for TRSs is defined, based on the categorical construct of a *monad*. The main result implies modularity of confluence, with two generalizations: First, the results also holds for TRSs with extra variables in rhs's of rules, and second, it also holds for systems consisting of *generalized* rewrite rules, which can be viewed as a very restricted form of conditional TRSs. Furthermore, it is shown that dropping the other variable restriction for TRSs (no lhs should be a variable) cannot simply be dropped. A counterexample is given by the disjoint confluent one-rule systems consisting of $x \to A(x)$ and of $B(C(y)) \to C(B(y))$. In the union we have the non-joinable local divergence $B(A(C(z))) \leftarrow B(C(z)) \to C(B(z))$.

Another, more recent and very interesting extension, also based on the simplified proof of Toyama's result for confluence, is presented in [33]. It is shown that two different properties are crucial for a successful proof, a so-called *cleaning lemma*, whose goal is to anticipate the application of collapsing reductions, and a modularity property of *ordered completion* that enables to pairwise match the *caps* and *alien substitutions* of two equivalent terms obtained from the cleaning lemma. Here, the basic idea of *ordered rewriting* is to rewrite with the (in general infinitely many) ordered instances of a given set of equations E, w.r.t. a reduction ordering $\succ$ that is total on ground terms. *Ordered completion* is based on computing *ordered critical pairs*, and is able to complete $(E, \succ)$ into $(E^\infty, \succ)$ such that the associated rewrite system R^∞ is confluent. Given two sets of equations E and S over disjoint signatures, a key observation is that $(E \cup S)^\infty = E^\infty \cup S^\infty$ for any reduction ordering that is total on ground terms. Therefore, ordered completion is modular for disjoint unions. Furthermore, the result of completion is not changed when adding an arbitrary set of free variables (considered as constants) provided the ordering is appropriately extended which is e.g. possible with the *lexicographic path ordering*. Thus, $(E \cup S)$-equivalent terms become joinable by ordered rewriting with $E^\infty \cup S^\infty$. The approach works under the assumption that variables are in normal form w.r.t. the union of the rewrite systems, i.e., if no rule has a variable as lhs (as in [49]), but also allows extra variables in rhs's. Consider the (extended) one-rule systems $\mathcal{R} = \{x \to f(x)\}$ and $\mathcal{S} = \{g(a) \to b\}$. Then we have the non-joinable divergence $b \leftarrow_{\mathcal{S}} g(a) \to_{\mathcal{R}} g(f(a))$. However, one can recover confluence here by considering (the also confluent) $\mathcal{R}' = \{x \to f(x), f(x) \to x\}$ instead of $\mathcal{R}$. It is easy to see that then $\mathcal{R}' \cup \mathcal{S}$ is confluent. However, recovering confluence systematically simply by adding inverse rules of *expansion rules* as above does not work in general.

Example 4 (*[33]*). $\mathcal{R} = \{x \to f(x), f(x) \to h(x)\}$ and $\mathcal{S} = \{g(a) \to a\}$ are both confluent. $\mathcal{R}' = \{x \to f(x), f(x) \to x, f(x) \to h(x)\}$ is also confluent, since $f(x) \to x$ is an equational consequence of $\mathcal{R}$. But the union $\mathcal{R}' \cup \mathcal{S}$ is non-confluent, as witnessed by the non-joinable divergence $a \leftarrow_{\mathcal{S}} g(a) \to_{\mathcal{R}} g(f(a)) \to_{\mathcal{R}} g(h(a))$. Here, no reduct of a contains an occurrence of g, whereas all reducts of $g(h(a))$ contain the symbol g.

The overall approach developed in [33] extends relatively easily to (modularity of confluence) w.r.t. rewriting modulo, under reasonable assumptions and for the common forms of rewriting modulo. For details we refer to [33].

Van Oostrom [90] has investigated (and proved) modularity of *constructive confluence*, based on the powerful technique and framework of *decreasing diagrams* (for characterizing, proving and reasoning about confluence), cf. e.g. [87,89,88]. The constructive character of checking confluence modularly is based on (recursively) applying decreasing diagrams in a clever way, assuming that confluence of the components can be constructively checked. For more details about this approach we refer to the extensive literature [90].

4.2. Weakened confluence properties related to normal forms

Since in many situations and settings the full confluence property of a given ARS or TRS does not or cannot hold, weaker versions of such properties related to normal forms may still hold. This is why, besides CR, the properties NF, UN and $\mathrm{UN}^{\rightarrow}$ are also interesting. Their relationship is well-known to be as follows.

Fact 5. *For ARSs we have:*

(a) $CR \implies NF \implies UN \implies UN^{\rightarrow}$*, and in general all implications are proper.*
(b) $WN \implies [\ UN^{\rightarrow} \iff UN \iff NF \iff CR\ ]$.

The modularity of these confluence properties related to normal forms has been thoroughly investigated by Middeldorp [53,55]. First of all, he could show that UN is modular, where the key idea for the proof is the interesting observation (and fairly complex construction) that every TRS with the unique normal form property (UN) can be extended to a confluent TRS with the same set of normal forms.

Concerning the normal form property NF, Middeldorp observed its non-modularity, cf. Example 6.

Example 6. Consider the two disjoint TRSs given by

$$\mathcal{R}_1 = \begin{Bmatrix} a \to b & b \to b \\ a \to c & c \to c \end{Bmatrix} \qquad \mathcal{R}_2 = \{F(x,x) \to C\}.$$

Both systems are obviously NF, but in their disjoint union $\mathcal{R}$ we have

$$F(b,c) \leftarrow F(a,c) \leftarrow F(a,a) \to C$$

with C being a normal form. Yet, $F(b,c) \to^* C$ does not hold, hence $\mathcal{R}$ is not NF.

Note that in Example 6, $\mathcal{R}_2$ is not left-linear. Middeldorp could show that non-left-linearity is essential for such counterexamples, i.e., that for left-linear TRSs the property NF is indeed modular.

The last of the confluence properties related to normal forms is $\mathrm{UN}^{\rightarrow}$. Similarly to NF, $\mathrm{UN}^{\rightarrow}$ is not modular.

Example 7 (*[53,55]*)**.** Consider the two disjoint TRSs given by

$$\mathcal{R}_1 = \begin{Bmatrix} a \to b & a \to c & c \to c \\ d \to c & d \to e & \end{Bmatrix} \qquad \mathcal{R}_2 = \{F(x,x) \to C\}.$$

Both TRSs are easily seen to be $\mathrm{UN}^{\rightarrow}$ but in their disjoint union the mixed term $F(a,d)$ has two distinct normal forms:

$$F(b,e) \leftarrow F(b,d) \leftarrow F(a,d) \to F(c,d) \to F(c,c) \to C.$$

Hence $\mathcal{R}$ is not $\mathrm{UN}^{\rightarrow}$.

Middeldorp conjectured that in Example 7 again non-left-linearity is crucial to make the counterexample work. He succeeded to prove the modularity of $\mathrm{UN}^{\rightarrow}$ for left-linear TRSs, but only under the additional assumption that both systems are non-collapsing. Only a couple of years later this conjecture was proven by Marchiori [51], based on a construction which was used in similar contexts, too.

Theorem 8. *The following (non-)modularity properties hold:*

- *UN is modular [53,55].*
- *NF and $UN^{\rightarrow}$ are not modular [53,55].*
- *NF is modular for left-linear TRSs [53,55].*
- *$UN^{\rightarrow}$ is modular for left-linear TRSs [51].*

4.3. Weakened confluence properties related to variables as normal forms

Let us now consider a few more properties having to do with confluence and a special version of normal forms, namely variables. A TRS $\mathcal{R}$ is said to be

- *consistent* (CONS) if $x \leftrightarrow^*_{\mathcal{R}} y$ for distinct variables x, y does not hold,
- *consistent w.r.t. reduction* ($\mathrm{CONS}^{\rightarrow}$) if there is no term s with $x \leftarrow^*_{\mathcal{R}} s \to^* y$ for two distinct variables x, y,
- *non-deterministically collapsing* (NDC) if $x \leftarrow^* s \to^* y$ for some term $s = C[x, y]$ and distinct variables x, y (thus NDC is just the negation of $\mathrm{CONS}^{\rightarrow}$ and vice versa),
- *collapsing confluent* (CCR) if $x \leftarrow^*_{\mathcal{R}} s \to^*_{\mathcal{R}} t$ implies $t \to^*_{\mathcal{R}} x$ for every $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V}), x \in \mathcal{V}$, and

- *uniquely collapsing* (UC) if, for every $s \in \mathcal{T}(\mathcal{F}, \{x\})$, $s = C[x, \dots, x]_{p_1,\dots,p_n} \to^* x$ (where all occurrences of x in s are displayed) implies that x has a unique ancestor [10] in s at p_i (for some unique i, $1 \le i \le n$).

Fact 9. *The properties CR, CCR, CONS, CONS$^{\to}$ and UC are related as follows:*

$$CR \Longrightarrow CCR \Longrightarrow CONS \Longrightarrow CONS^{\to} \Longleftarrow UC$$

where in general all implications are proper.

Concerning modularity of these properties, Schmidt-Schauß had shown in [75] that the equational version of consistency (CONS) is modular. However, this does not hold for the operational version CONS$^{\to}$ [51], and also not for the properties CCR and UC.

Example 10 (*[25]*)**.** Consider the two disjoint TRSs given by

$$\mathcal{R}_1 = \begin{Bmatrix} f(x, x, y, z) \to y \\ f(x, g(x), y, z) \to z \end{Bmatrix} \qquad \mathcal{R}_2 = \begin{Bmatrix} G(x) \to x \\ G(x) \to A \end{Bmatrix}.$$

Both TRSs are CONS$^{\to}$, but their disjoint union is not:

$$z \leftarrow f(A, g(A), y, z) \leftarrow^+ f(G(g(A)), G(g(A)), y, z) \to y.$$

Example 11 (*[25]*)**.** Consider the two disjoint TRSs given by

$$\mathcal{R}_1 = \{f(x, x, y) \to y\} \qquad \mathcal{R}_2 = \begin{Bmatrix} A \to B \\ A \to C \end{Bmatrix}.$$

They are obviously CCR, but their disjoint union $\mathcal{R}$ is not, due to

$$f(B, C, y) \leftarrow^+ f(A, A, y) \to y,$$

where both $f(B, C, y)$ and y are in normal form. Hence we do not have $f(B, C, y) \to^* y$, thus $\mathcal{R}$ is not CCR.

Analogously to NF and UN$^{\to}$, for CCR and UC it turns out that together with left-linearity (LL) they become modular.

Theorem 12. *The following modularity results hold:*

- *CONS$^{\to}$ is modular for left-linear TRSs [51].*
- *CCR is modular for left-linear TRSs [25].*
- *UC is modular for left-linear TRSs [25].*

Actually, the last result in Theorem 12 is simply due to the equivalence of the first and third property under left-linearity:

$$\text{LL} \Longrightarrow [\text{UC} \Longleftrightarrow \text{CONS}^{\to}].$$

The proof of the first result, the modularity of CONS$^{\to}$ for left-linear TRSs is similar to the proof of modularity of UN$^{\to}$ for left-linear TRSs, cf. the last item in Theorem 8 and is highly non-trivial. It heavily relies on the uniquely collapsing property (UC) which can be exploited, due to left-linearity, to transform finite or infinite derivations into smaller equivalent finite or smaller infinite derivations, respectively.[11]

Summarizing, the following modularity results for properties of TRSs related to confluence and normal forms are known.[12]

property	modular?	reference	property	modular?	reference
CR	+	[82]			
NF	−	[53,55]	NF & LL	+	[53,55]
UN	+	[53,55]			
UN$^{\to}$	−	[53,55]	UN$^{\to}$ & LL	+	[51]
CONS	+	[75]			
CONS$^{\to}$	−	[51]	CONS$^{\to}$ & LL	+	[51]
CCR	−	[25]	CCR & LL	+	[25]
UC	−	[25]	UC & LL	+	[25]

10 i.e., along the derivation $D: s = C[x, \dots, x]_{p_1,\dots,p_n} \to^* x, \epsilon$, the position of x has a unique position p_j in s (for some unique j) where it stems from. For a precise definition of *ancestor* we refer e.g. to [9]. Note that for a non-left-linear rule $l = l[x, \dots, x]_{p_1,\dots,p_n} \to x$ the x on the rhs (i.e. its position ϵ) does not have a unique ancestor in l (but n ones, namely $p_1, \dots, p_n$).

11 Smaller is meant here w.r.t. some well-founded ordering.

12 We do not claim any completeness of this list, and concentrate e.g. on the known symmetric modularity results.

5. Modularity analysis of termination properties

5.1. Modularity criteria for termination

Toyama's basic Counterexample 1 to the modularity of termination has a couple of properties which appear to be crucial for the counterexample to work: The first system is duplicating, and the second system is collapsing and non-confluent (even non-CONS$^{\rightarrow}$). For a detailed presentation and discussion of tempting conjectures and counterexamples, inspired by [82], cf. [81].

Confluence as additional property is not sufficient.

Example 13 (*[12]*).

$$\mathcal{R}_1 = \begin{Bmatrix} f(a,b,x) \to f(x,x,x) \\ a \to c \\ b \to c \\ f(x,y,z) \to c \end{Bmatrix} \qquad \mathcal{R}_2 = \begin{Bmatrix} K(x,y,y) \to x \\ K(y,x,y) \to x \end{Bmatrix}.$$

Both systems are terminating and confluent, but in their disjoint union we have the cycle

$$\begin{aligned} f(a,b,K(a,b,c)) &\to_{\mathcal{R}_1} f(K(a,b,c),K(a,b,c),K(a,b,c)) \\ &\to^+_{\mathcal{R}_1} f(K(a,c,c),K(c,b,c),K(a,b,c)) \\ &\to^+_{\mathcal{R}_2} f(a,b,K(a,b,c)) \to \cdots. \end{aligned}$$

As observed in [81], in Example 13 the first system is not (inter)reduced. Yet, further more sophisticated counterexamples from [81] showed that being, besides terminating, (inter)reduced (RED) and confluent (CR), is still not sufficient for the modularity of termination. Still another conjecture, motivated by the existing counterexamples, was based on the observation that one of the systems in each counterexample was erasing. However, Ohlebusch [66] could show that there exists a (striking) counterexample to the modularity of termination where both systems are non-erasing, confluent and even (inter)reduced.

Looking at Counterexamples 1 and 13, we observe that they enjoy a couple of (mainly) syntactical properties which appear to be essential for the respective counterexample property: In Example 13, both systems are confluent (and terminating), but one of them (the second one) is collapsing and non-left-linear, and the other one is duplicating, not an overlay system and not simply terminating [13] (moreover, both systems are erasing). On the positive side there exist various sufficient criteria for modularity of termination. But first, let us introduce a bit more terminology and notation for termination related properties.

A *rewrite ordering* (on $\mathcal{T}(\mathcal{F},\mathcal{V})$) is a strict partial ordering on terms closed under contexts and substitutions. A *reduction ordering* is a well-founded rewrite ordering. A rewrite ordering $>$ is a *simplification ordering* if it possesses the subterm property $C[s] > s$ for any s and any non-empty context $C[]$. A TRS $\mathcal{R}^{\mathcal{F}}$ is *simplifying* if there exists a simplification ordering $>$ with $\to_{\mathcal{R}} \subseteq >$. It is *simply terminating* if there exists a well-founded simplification ordering $>$ which contains $\to_{\mathcal{R}}$. The *embedding* TRS $\mathcal{R}^{\mathcal{F}}_{emb} = (\mathcal{F}, R^{\mathcal{F}}_{emb}) = \{f(x_1,\ldots,x_n) \to x_i \mid 1 \le i \le n = ar(f), f \in \mathcal{F}\}$ consists of all projection rules for all $f \in \mathcal{F}$. The following facts about simplifyingness and simple termination are well-known.

- A TRS $\mathcal{R}^{\mathcal{F}}$ is simplifying iff $\to^+_{R\cup\mathcal{R}^{\mathcal{F}}_{emb}}$ is irreflexive.
- A TRS $\mathcal{R}^{\mathcal{F}}$ is simply terminating iff $(\mathcal{F}, R \cup \mathcal{R}^{\mathcal{F}}_{emb})$ is terminating.
- Simplifyingness and simple termination of $\mathcal{R}^{\mathcal{F}}$ coincide if $\mathcal{F}$ is finite, or – slightly weaker – if $\mathcal{R}^{\mathcal{F}}$ *introduces only finitely many function symbols* [61]. [14]

Less well-known is a rational reconstruction of the theory of simplification orderings and of the notion of simple termination that corresponds more closely to the applicability of Kruskal's Tree Theorem [40] than the usual version of these notions and easily covers the case of arbitrary infinite signatures as well [58].

The *homeomorphic embedding relation* $\trianglelefteq$ on terms is recursively defined by $s = f(s_1,\ldots,s_m) \trianglelefteq g(t_1,\ldots,t_n) = t$ if either $s \trianglelefteq t_i$ for some $i \in \{1,\ldots,n\}$ or $f = g$ and $s_j \trianglelefteq t_{i_j}$ for all $j \in \{1,\ldots,m\}$ and some $i_1,\ldots,i_m$ with $1 \le i_1 < i_2 < \cdots < i_m \le n$. A TRS $\mathcal{R}$ is said to be *self-embedding* if there exists a self-embedding $\mathcal{R}$-derivation, i.e. a reduction sequence $s_1 \to s_2 \to s_3 \to \cdots$ with $s_i \trianglelefteq s_j$ for some i, j with $i < j$.

We have the following sufficient criteria for modularity of termination.

Theorem 14. *Let $\mathcal{R}_1^{\mathcal{F}_1}$, $\mathcal{R}_2^{\mathcal{F}_2}$ be two disjoint terminating TRSs. Then their disjoint union $\mathcal{R}^{\mathcal{F}}$ is terminating if one of the following conditions is satisfied:*

[13] A simply terminating TRS cannot be self-embedding, which is the case here: $s = f(a,b,K(a,b,c)) \to^+_{\mathcal{R}} f(K(a,b,c),K(a,b,c),K(a,b,c)) = t$ with $s \trianglelefteq t$ (see below for the definitions of *simply terminating*, *self-embedding* and of $s \trianglelefteq t$).

[14] The *introduced* function symbols (of $\mathcal{R}^{\mathcal{F}}$) are those appearing in some rule $l \to r$ of $\mathcal{R}$ in r, but not in l.

- *Both systems are non-collapsing [72].*
- *Both systems are non-duplicating [72].*
- *One of the systems is non-collapsing and non-duplicating [54].*
- *Both systems are simply terminating [43].* [15]
- *Both systems are non-deterministically collapsing [19].*

Actually, all the modularity results of Theorem 14 turned out to be consequences of the following abstract characterization result for minimal counterexamples to modularity.

Theorem 15 *([63,61]). Let $\mathcal{R}_1^{\mathcal{F}_1}, \mathcal{R}_2^{\mathcal{F}_2}$ be two disjoint terminating TRSs such that their disjoint union $\mathcal{R}^{\mathcal{F}}$ is non-terminating. Then one of the systems, let us say $\mathcal{R}_1^{\mathcal{F}_1}$, is collapsing and the other one, $\mathcal{R}_2^{\mathcal{F}_2}$, is such that $\mathcal{R}_2^{\mathcal{F}_2} \uplus (\{A, G\}, \{G(x, y) \to x, G(x, y) \to y\})$ is non-terminating, or vice versa.*

TRSs $\mathcal{R}^{\mathcal{F}}$ with the property that $\mathcal{R}^{\mathcal{F}} \uplus (\{A, G\}, \{G(x, y) \to x, G(x, y) \to y\})$ is terminating are called $\mathcal{C}_{\mathcal{E}}$-terminating ($\mathcal{C}_{\mathcal{E}}$-SN) [63,61].

Note the result of Theorem 15, under the assumption that the involved TRSs are finitely branching, was first proved in [19]. The latter proof works by minimal counterexample and an abstracting transformation that transforms the principal aliens, let us say of top-white colour, such that all "black information" that eventually may pop up there is computed in advance, in such a way that it then only has to be revealed (selected). This happens via a fresh white layer separating variadic function symbol G, whose arguments represent the top-black subterms that may eventually come up. Since the principal aliens may have an unknown rank, the whole construction has to proceed recursively, and needs (1) a fixed ordering for putting (recursively abstracted) top-black successors of the principal alien as arguments of the fresh G-symbol in the right order, and (2) the fact that only finitely many such top-black successors of the initial principal subterm exist (in order for the whole construction to yield a finite term). The latter finiteness assumption is guaranteed by the finitely branching requirement (*) for both systems. The former requirement (1) is easy to satisfy. In order not to leave the setting of fixed-arity function symbols, instead of one variadic function symbol G we can also use a binary G together with a fresh constant A for encoding $G(t_1, \ldots, t_n)$ as $G(t_1, G(t_2, \ldots, G(t_n, A) \ldots))$.

The proof without the finitely branching assumption needs a substantially more complex abstracting transformation which does not explicitly compute in advance all top-black successors but proceeds more lazily. Yet, also only the black rule system is needed plus the selection system using the non-deterministic G-projection rules, in order to simulate the original infinite derivation of minimal rank.

Using this construction (one of the two above) every outer step using a black rule translates in a (the same) black step in the outermost layer, and every inner step translates into a sequence of steps from the black system and from $\{G(x, y) \to x, G(x, y) \to y\}$. Since there must be infinitely many outer steps due to minimality, the claim holds, i.e., non-termination of $\mathcal{R}_2^{\mathcal{F}_2} \uplus (\{A, G\}, \{G(x, y) \to x, G(x, y) \to y\})$ follows and the other system $\mathcal{R}_1^{\mathcal{F}_1}$ must be collapsing, because otherwise, by identifying abstraction of the principal aliens, $\mathcal{R}_2^{\mathcal{F}_2}$ would have already been non-terminating.

Interestingly, all individual modularity criteria of Theorem 14 (and others) can be derived relatively easily from Theorem 15.

Let us mention a few more results of a different type.

Theorem 16. *Termination is modular for:*

(a) *locally confluent overlay systems [20];*
(b) *left-linear confluent TRSs [85]; and for*
(c) *left-linear TRSs that are consistent w.r.t. reduction [76].*

Note that in Example 13 both systems are confluent (hence also consistent w.r.t. reduction), the first one is not an overlay system and the second one is not left-linear. In Toyama's Example 1 both systems are left-linear, but the second one is not consistent w.r.t. reduction. (c) also relies on the modularity of $\mathrm{CONS}^{\to}$ for left-linear systems.

The proof of (a) above is very elegant, by the 'modular' approach (see below in Section 5.2 [20]). The proofs of (b) [85] and of (c) [76] are highly complex, and heavily rely on left-linearity. In [85] the authors develop, for left-linear confluent systems, a kind of comprehensive structure theory of collapsing reduction which goes far beyond what is needed for the actual proof of (b). In view of (c), however, it becomes clear, that full confluence of the individual systems is not needed (besides left-linearity) to preserve termination. Instead, consistency w.r.t. reduction ($\mathrm{CONS}^{\to}$) is sufficient. The crucial property in disjoint left-linear systems with $\mathrm{CONS}^{\to}$ is the following. If $s = C[\![s_1, \ldots, s_k]\!] \to_{\mathcal{R}}^* t$ where the s_i are preserved and where the reduction $s \to_{\mathcal{R}}^* t$ is colour changing with only one step that is destructive at level 1, t must be a descendant of some unique s_i (due to left-linearity and UC). This is reflected by the fact that $\bar{s} = C[x, \ldots, x]_{p_1,\ldots,p_n} \to_{\mathcal{R}}^* x$, where all occurrences of the variable x are displayed, implies that $x\,(\epsilon)$ has a unique ancestor p_i in $\bar{s}$.

The following table summarizes (non-)modularity results for termination of TRSs. [16]

[15] Interestingly, the related property of being *(homeomorphically) non-self-embedding* is not modular [19].

[16] Again we do not claim any completeness of this list, and concentrate e.g. on the known symmetric modularity results.

property	modular?	reference
SN	−	[81,82], Example 1
SN & NCOL	+	[72]
SN & NDUP	+	[54]
SN & CR	−	[12]
SN & CR & RED	−	[81]
SN & CR & RED & NE	−	[66]
simple SN	+	[43]
$\mathcal{C}_{\mathcal{E}}$-SN	+	[19,63]
simple SN[17]	+	[58]
SN & WCR & OS	+	[20]
SN & CR & LL	+	[85]
SN & CONS$^{\rightarrow}$ & LL	+	[50,76]

One further preservation result is the following: SN & NDC is preserved under disjoint unions [19], which is an easy consequence of the modularity of $\mathcal{C}_{\mathcal{E}}$-SN. It does not quite fit into the table, since strictly speaking it is not a modularity result, but only a preservation result in one direction. If the disjoint union of two systems is SN & NDC, then, clearly, both systems are SN, but not both of them need to be NDC. A simple example is given by $\mathcal{R}_1 = \{f(x) \to x\}$, $\mathcal{R}_2 = \{G(x, y) \to x, G(x, y) \to y\}$. The alternative formulation 'SN is modular for TRSs satisfying NDC' is also not good, because it remains unclear what should be modular precisely, in the sense of its definition. More generally speaking, there are many preservation results of the above type, where the other direction does not hold.

Let us next consider the question how big the rank of minimal counterexamples to the modularity of termination can/must be. It cannot be 1; it can also not be 2, because then the first collapsing step would reduce it to 1; and it can be 3, as Example 1 shows. The natural question is: Is it necessarily always 3, as almost all counterexamples in the literature suggest. A positive response would lead to a substantial simplification of the proofs of many modularity results, because the layer structure of assumed counterexamples would be much simpler. Unfortunately, the answer to the question is No.

Fact 17 *([19]). The rank of minimal counterexamples to modularity of termination may be arbitrarily high.*

Example 18. Consider the disjoint TRSs given by
$\mathcal{R}_1 \colon f(x, g(x), \dots, g^n(x), y) \to f(y, \dots, y)$, $\quad \mathcal{R}_2 \colon G(x) \to x, G(x) \to A$.
Here, f has arity $n + 2$ and $g^n(x)$ stands for the n-fold application of g to x. Both $\mathcal{R}_1$ and $\mathcal{R}_2$ are clearly terminating, but $\mathcal{R}$ is non-terminating. For instance, we have the following infinite ($\mathcal{R}$-)derivation

$$\begin{array}{rll} D\colon f((Gg)^nA, (Gg)^nA, \dots, (Gg)^nA) & \to_{\mathcal{R}_2} & f(A, (Gg)^nA, \dots, (Gg)^nA) \\ & \to_{\mathcal{R}_2} & f(A, g(Gg)^{n-1}A, \dots, (Gg)^nA) \\ & \to_{\mathcal{R}_2} & f(A, gA, \dots, (Gg)^nA) \\ & \vdots & \\ & \to_{\mathcal{R}_2} & f(A, gA, g^2A, \dots, g^nA, (Gg)^nA) \\ & \to_{\mathcal{R}_1} & f((Gg)^nA, (Gg)^nA, \dots, (Gg)^nA) \\ & \to_{\mathcal{R}_2} & \cdots \end{array}$$

of rank $2n + 2$. A careful analysis of possible reductions shows that for this example $2n + 2$ is the minimal rank of any conceivable non-terminating ($\mathcal{R}$)-derivation. Moreover, it is straightforward to modify the above example in such a way that only (finite) signatures with function symbols of (uniformly) bounded arities are involved. For instance, one may use a binary f' and the encoding $f'(x_1, f'(x_2, \dots, f'(x_{n-1}, x_n) \dots))$ for $f(x_1, \dots, x_n)$.

System $\mathcal{R}_1$ in Example 18 is obviously non-left-linear (and $\mathcal{R}_2$ is non-confluent), and this seems to be essential for the family of counterexamples to work. Yet, very recently it has been shown that there also exist families of left-linear counterexamples (as well as families of confluent counterexamples) to modularity of termination where the minimal rank of non-terminating derivations is arbitrarily high [26]. The basic example from [26] is as follows.

Example 19. Consider the family of disjoint TRSs $(\mathcal{R}_1^n, \mathcal{R}_2^n)$ given by

$$\mathcal{R}_1^n = \left\{ \begin{array}{rcl} f_1(g(x), a, y) & \to & f_2(x, x, y) \\ & \vdots & \\ f_{n-1}(g(x), a, y) & \to & f_n(x, x, y) \\ f_n(g(x), a, y) & \to & f_1(y, y, y) \end{array} \right\} \qquad \mathcal{R}_2^n = \left\{ \begin{array}{l} H(x, y) \to x \\ H(x, y) \to y \end{array} \right\}.$$

[17] Here we mean the notion of simple termination as defined by Middeldorp and Zantema [58], which is closer in spirit to the underlying *Tree Theorem of Kruskal* [40].

Note that $\mathcal{R}_2^n$ is fixed and $\mathcal{R}_1^n$ is parameterized by $n \geq 1$. We observe that this example is based on a variant of Toyama's Counterexample 1. Both systems $\mathcal{R}_1^n$ and $\mathcal{R}_2^n$ are terminating for all $n \geq 1$. Note that for $n = 1$ we obtain the basic $\mathcal{R}_1^1 = \{f_1(g(x), a, y) \to f_1(y, y, y)\}$. In the disjoint union we have the non-termination witness $s = f_1(\phi^n(a), \phi^n(a), \phi^n(a))$, where $\phi(x) = H(g(x), a)$, of rank $2n + 1$:

$$\begin{array}{rcl}
s & = & f_1(\phi^n(a), \phi^n(a), \phi^n(a)) \\
& = & f_1(H(g(\phi^{n-1}(a)), a), H(g(\phi^{n-1}(a)), a), \phi^n(a)) \\
& \to^+ & f_1(g(\phi^{n-1}(a)), a, \phi^n(a)) \\
& \to & f_2(\phi^{n-1}(a), \phi^{n-1}(a), \phi^n(a))) \\
& \to^+ & f_2(g(\phi^{n-2}(a)), a, \phi^n(a)) \\
& \vdots & \vdots \\
& \to^+ & f_n(g(\phi^0(a)), a, \phi^n(a)) \\
& = & f_n(g(a), a, \phi^n(a)) \\
& \to & f_1(\phi^n(a), \phi^n(a), \phi^n(a)) = s.
\end{array}$$

In [26] it is shown that $2n + 1$ is indeed the minimal rank of non-terminating derivations in the disjoint union of $\mathcal{R}_1^n$ and $\mathcal{R}_2^n$.

In the above example the system $\mathcal{R}_2^n$ is always non-confluent. In [26] it is shown that also for confluent systems there are families of examples where the minimal rank of counterexamples (to modularity of termination) is arbitrarily high (there, at least one of the systems in each case is necessarily non-left-linear, due to the result from [85]). Similar complexity results for witnessing the non-modularity of the properties $\mathrm{UN}^{\to}$ and $\mathrm{CONS}^{\to}$ (in non-left-linear systems) are also presented in [26].

5.2. *Modularity of weak termination properties*

From the very beginning of modularity research, besides ordinary termination (SN), weaker variants of termination have also been investigated. One very early notion explored in this context is *modular reduction* due to Kurihara and Kaji [41,42] which we will not deepen here. Other obvious candidates are normalization (WN) of the ordinary reduction relation and normalization and termination of rewriting under various strategies, as well as normalization and termination of otherwise restricted versions of rewriting (like *lazy rewriting* [15,48,74], *context-sensitive* rewriting [47,29,30], *rewriting with forbidden patterns* [31]) and many more.

In contrast to SN, WN is indeed a modular property which is not difficult to prove and has been remarked by several researchers at around the same time (cf. e.g. [85, conf. version], [8]). The proof is bottom-up by an easy induction on the rank.

Innermost rewriting has nice modularity properties. Innermost normalization ($\mathrm{WIN} = \mathrm{WN}(\to_i)$) and innermost termination ($\mathrm{SIN} = \mathrm{SN}(\to_i)$) are also both modular, since in essence one can reason inside-out.

Especially the modularity of SIN is very useful, in combination with further properties P that turn the implication ($\mathrm{SIN} \Longleftarrow \mathrm{SN}$) into an equivalence: $P \Longrightarrow [\,\mathrm{SIN} \Longleftrightarrow \mathrm{SN}\,]$. Examples for P are (OS & WCR), NO, WNO (cf. [20,23,21]. If P is modular, too, as is the case with the just mentioned examples of P, then one can use the *modular approach* for proving modularity of SN (more precisely, of (SN & P)) as it has been mentioned in Section 3. Known conditions for the sufficiency of $\mathrm{SIN} \Longrightarrow \mathrm{SN}$) are among others the following[18]:

- OS & WCR,
- NO,
- WNO, and
- OS & RL.

The last result appears in [73], as noticed by [93]. It does not provide a really new modularity result for termination, since right-linear systems are in particular non-duplicating, and for such systems termination is well-known to be modular. However, it is still interesting as a sufficient criterion for $\mathrm{SIN} \Longrightarrow \mathrm{SN}$, in order to be able to switch from an SN-proof of some given system to a (hopefully easier) SIN-proof of that system. Second, it is also interesting, because among the criteria above it is the only one that, together with SIN, does not imply confluence. Note that RL cannot be weakened to NDUP, as witnessed by a counterexample in [73].

Recently, also more fine-grained variants of innermost rewriting have been investigated for normalization and termination, namely leftmost innermost and (maximal) parallel innermost rewriting. All these variants of SIN turn out to be modular as well. The same analysis can of course be done with outermost rewriting, too [28,27]. Somehow surprisingly, outermost rewriting in general has extremely bad modularity properties w.r.t. normalization and termination. For instance,

18 A few further and partially slightly more general conditions, which are based on critical pair criteria (specifying how critical pairs should be joinable) are given in [23].

even under signature extensions in general none of these properties is preserved, see e.g. Example 20 below. But fortunately also various sufficient criteria are meanwhile known, especially left-linearity, cf. [28,27].

We show that outermost termination (SON), outermost normalization (WON), leftmost outermost termination (SLON) and leftmost outermost normalization are not preserved under signature extensions in general.

Example 20 (*[28]*). Consider the TRS $\mathcal{R}$ over the signature $\mathcal{F} = \{f_1, f_2, g, c, d_1, d_2\}$:

$$\begin{array}{rclcrcl} g(f_1(x,y,y)) & \to & g(f_2(x,x,y)) & \qquad & g(f_2(*,x,y)) & \to & c \\ g(f_1(y,x,y)) & \to & g(f_2(x,x,y)) & & g(f_2(x,*,y)) & \to & c \\ f_2(x,x,y) & \to & f_1(x,x,y) & & g(f_2(x,y,*)) & \to & c \\ d_1 & \to & d_2 & & g(f_2(x,x,x)) & \to & c. \end{array}$$

Here, $* \in \{c, d_1, d_2, g(z_1), f_1(z_1,z_2,z_3), f_2(z_1,z_2,z_3)\}$. With some effort one can show that $\mathcal{R}$ is SON, hence also WON, SLON and WLON. However, if we add a fresh unary function symbol H, all these properties get lost. To wit, consider $s = g(f_1(H(d_1), H(d_1), H(d_2)))$ which initiates the (only) outermost derivations (the contracted outermost redexes are underlined and outermost reduction is denoted by $\overset{o}{\to}$):

$$\begin{array}{rcl} s & = & g(f_1(H(\underline{d_1}), H(d_1), H(d_2))) \overset{o}{\to} \underline{g(f_1(H(d_2), H(d_1), H(d_2)))} \\ & \overset{o}{\to} & g(\underline{f_2(H(d_1), H(d_1), H(d_2))}) \overset{o}{\to} s \to \cdots \quad \text{and} \\ s & = & g(f_1(H(d_1), H(\underline{d_1}), H(d_2))) \overset{o}{\to} \underline{g(f_1(H(d_1), H(d_2), H(d_2)))} \\ & \overset{o}{\to} & g(\underline{f_2(H(d_1), H(d_1), H(d_2))}) \overset{o}{\to} s \to \cdots. \end{array}$$

6. Extensions

We will now briefly look at various extensions of the basic modularity theory in several directions.

First let us mention a few further interesting works on other modularity topics. Durand and Middeldorp studied modularity of deciding call-by-need [13]. A modular complexity analysis (of rewriting) via relative completion techniques was presented in [91].

In the following we will look at modularity when weakening the disjointness requirement, extending the type of rewrite systems investigated and when taking another kind of modularity principle as basis of the analysis.

6.1. Beyond disjointness

Most of the theory and modularity criteria for disjoint systems can be extended, by properly taking into account the additional sources of problems. What is different in constructor sharing systems is that interaction/interference between the two systems can not only happen via destructive steps using collapsing rules, but also via rules that have a shared constructor at the root of the rhs.

For instance, $\mathcal{R}_1 = \{f(x,x) \to a, f(x,c(x)) \to b\}$ and $\mathcal{R}_2 = \{d \to c(d)\}$ with shared constructor c are both confluent, but their union $\mathcal{R}$ is not:

$$b \leftarrow f(d, c(d)) \leftarrow f(d,d) \to a.$$

The problem here is the *constructor* (or *shared symbol*) *lifting* rule $d \to c(d)$. The disturbing effect of collapsing rule applications can be achieved here also via applying constructor-lifting rules. Consequently, the non-collapsing requirement in disjoint unions becomes here *layer preservation* [67,63]. The structure of the proof of modularity of confluence from the disjoint union case [34] carries over to the case of (at most) constructor sharing systems. Only *collapsing reduction* $\to_c$ has to be redefined, since now collapsing as well as shared symbol lifting rules may have a disturbing effect. For example, in the counterexample above, the 'new' *collapsing reduction* is not weakly normalizing any more: $d \to_c c(d) \to_c c(c(d)) \to_c \cdots$. If one can ensure WN($\to_c$) somehow, then the modularity proof for confluence goes through. One obviously sufficient criterion for WN($\to_c$) clearly is WN itself. So, if the union enjoys WN, everything is fine. And the union enjoys indeed WN, since the proof from the disjoint union case goes through without major modifications [62].

Interestingly, concerning modularity of termination for systems with (at most) shared constructors the adapted abstract characterization result of Theorem 15 does not hold anymore, only the weaker version with the finitely branching requirement. With this requirement the proof of [19] can easily be extended from the disjoint union case to the case of constructor sharing systems, cf. [19]. The finitely branching requirement here is crucial as was observed by Ohlebusch [63]. A relatively simple counterexample without this condition is the following.

Example 21. Consider the TRSs $\mathcal{R}_1 = \{A \to c_i \mid i \geq 1\}$ and $\mathcal{R}_2 = \{f_i(c_i, x) \to f_{i+1}(x,x) \mid i \geq 1\}$ with shared constructors. Both systems share the constructors $\{c_i \mid i \geq 1\}$. $\mathcal{R}_1$ is infinitely branching, $\mathcal{R}_2$ finitely branching, but infinite (has infinitely many rules). In the union we have

$$f_1(c_1, A) \to f_2(A, A) \to f_2(c_2, A) \to f_3(c_3, A) \to \cdots.$$

Slightly more general than systems with *(at most) shared constructors* are *composable systems* where both systems share all defining rules for the shared defined function symbols. For comprehensive information on *composable* systems cf. e.g. [63,61,62,64,44]. Another generalization are the more interesting *hierarchical* combinations where one system may use defined symbols of the other one, but not vice versa. Such hierarchical combinations are in fact quite frequent, and in general not easy to handle. For techniques and results about these areas cf. e.g. [68,35–38,11].

6.2. Beyond TRSs

Modularity aspects of rewriting properties have also been investigated to some extent in other rewriting settings and formats, e.g. in *non-copying* term rewriting [45] and term graph rewriting [70,69,39] (here, non-duplication can be guaranteed by sharing), infinitary rewriting [77,78] and higher-order rewriting [5] (where many results from the finitary case do not hold anymore), conditional rewriting [55–57,18,19,21,22,24,60,65] (where one major difficulty is that rewrite steps are non-local, i.e., one step may involve the whole system for verifying the conditions), and context-sensitive rewriting [46,47,29,30] (where the missing monotonicity property substantially complicates the analysis).

6.3. Persistency

One very interesting notion which is closely related to modularity, is *persistency*. A property of TRSs is called *persistent* if for every many-sorted TRS $\mathcal{R}$ the property holds for $\mathcal{R}$ if and only if it also holds for $\theta(\mathcal{R})$. Here, the type elimination map θ is defined in the obvious way (it just removes any type information). In many-sorted rewriting we have a set S of *sorts*, S-sorted variables $(\mathcal{V}_s)_{s\in S}$ with $\mathcal{V}_s$ pairwise disjoint, and function symbols $\mathcal{F}$ equipped by sorts. The *arity* and the *sort* of a function symbol f are described by functions $ar\colon \mathcal{F} \to S^*$, $st\colon \mathcal{F} \to \mathcal{F}$. The sets of well-sorted terms $\mathcal{T}(\mathcal{F}, \mathcal{V})_s$ are defined mutually inductively as usual. Rewrite rules $l \to r$ are also S-sorted with l and r of the same sort, and enjoy the same variable conditions as before. A many-sorted TRS consists of the S-sorted signature (including variables) and S-sorted rules. The corresponding rewrite relation is defined as expected.

Some results about persistency that are known are as follows.

- Every component closed [19] persistent property is modular [92].
- Confluence is persistent [3].
- Termination is persistent for the class of many-sorted TRSs not containing both collapsing and duplicating rules [92].
- Termination is persistent for the class of TRSs in which all variables have the same sort [1].

Example 22. The one-rule TRS $\{f(a, b, x) \to f(x, x, x)\}$ can be easily shown to be terminating by applying the third result above. Let us type the system as follows: a, b, f have sort s_1, f has sort s_2 and arity (s_1, s_1, s_1). Then this many-sorted TRS is trivially terminating, since at most one reduction step is possible. By persistency the (unsorted) TRS above is also terminating.

Very interesting in this context is the work of van de Pol [86] who has shown the following results relating

1. persistency
2. modularity for TRSs
3. modularity for many-sorted TRSs[20]

(of component closed properties).

Zantema [92] showed (1) $\Rightarrow$ (2), van de Pol [86] proved (1) $\Rightarrow$ (3). The implication (3) $\Rightarrow$ (2) is trivial, and the main result of [86] is that the implication (3) $\Rightarrow$ (1) holds.

What remains open is whether (2) $\Longrightarrow$ (1) holds, i.e., whether modularity for TRSs implies persistency. This problem, with the above main result, reduces to the question whether (2) $\Longrightarrow$ (3) holds which appears to be difficult.

Yet, in practical applications reasoning via persistency is very powerful, flexible and fine-grained, much more than modularity. For that reason any progress in persistency research would be highly welcome and probably insightful.

7. Discussion

Applications and implications of modularity

Generally it can be said that more than 25 years of modularity research have been extremely fruitful and inspiring for the area itself, but also for other fields. The rigorous analysis of (often) syntax-based interactions, structuring mechanisms and commutation properties of (abstract rewrite systems and) TRSs (cf. also [71,83,16,7,11,33,4]) has provided deeper insights into relationships between and crucial aspects of TRS properties. For instance, the modular approach to proving modularity

[19] A property of binary relations is called *component closed* if it holds for $(\mathcal{T}, \to)$ if and only if it holds for $(C, (C \times C) \cap \to)$ for all components C of $(\mathcal{T}, \to)$. Here, a *component* is an equivalence class of the equivalence relation generated by $\to$. Note that component closedness is a quite reasonable restriction which is satisfied by most interesting TRS properties like SN, CR etc.

[20] Modularity for many-sorted TRSs (w.r.t. disjoint unions) is defined as for (unsorted or one-sorted) TRSs, but without requiring the sets of sorts to be disjoint.

of SN for certain classes of systems via modularity of SIN and sufficient criteria for the implication SIN $\Longrightarrow$ SN to hold has triggered research about further such criteria. Any such criterion allows to switch from an SN proof to an equivalent (easier) proof of SIN. This has been applied at various places in the literature (e.g. in [59]). It was and is massively exploited in the *dependency pair framework* [17] for proving termination which itself is very modularly defined and thus highly flexible. The correctness proof of one important refinement of this framework, namely the restriction to so-called *usable rules* heavily relies on a proof construction which is very similar to the abstraction construction of the characterization results of non-modularity of termination mentioned as Theorem 15 (the one for the finitely branching systems). This applies to restricting attention to *usable rules* in the frameworks for different versions of rewrite systems, i.e., for normal TRSs, for context-sensitive TRSs etc.

Since the proofs of results e.g. about termination, modular termination and termination via the dependency framework have become increasingly and inherently complex as well as hard to verify by hand, to increase trust there is also an increasing need to provide automatic support for certifying such proofs. This is an active and still growing area of research which itself sometimes yields new insights into modularity aspects, cf. e.g. [79].

Positive results on persistency of TRS properties allow for fine-grained search procedures based on (parallel) proof attempts for termination or confluence. In modern tools for automatic termination or confluence proofs, this principle, together with many other features, is increasingly incorporated, cf. the existing termination and confluence proving competitions [80,10]. Note that successfully incorporating persistency based search decompositions has become more and more feasible due to the massive benefits from using powerful available SAT and SMT solving tools for various subtasks in the proof attempts.

7.1. Open problems and questions

One major concrete and important open problem in modularity is the following.

Open Problem 23. *Does modularity imply persistency (cf. the discussion in 6.3 above)?*

In the author's view, further interesting questions, tasks and research directions would be the following.

Questions/Tasks 24.
- *Further develop the theory of order-sorted persistency (cf. [2,14]).*
- *Investigate persistency and persistency vs. modularity for non-disjoint unions, e.g. for (at most) constructor sharing TRSs.*
- *Try to obtain further interesting characterization results for minimal counterexamples to modularity/persistency of (non-modular/non-persistent) properties of TRS, in the spirit of Theorem 15.* [21]
- *Provide/find more semantic modularity/persistency criteria. Make modularity/persistency criteria more robust (against slight syntactical changes).* [22]
- *Further develop the theory about modularity in extended formats of rewriting, e.g., in rewriting modulo, conditional/constrained rewriting, higher-order rewriting, infinitary rewriting.* [23]

7.2. Recent work, emerging trends and perspectives

Recent interesting works on modularity aspects in term rewriting include [79,91,14]. Particularly interesting is [14] where the authors introduce *layer systems* that axiomatically specify how terms can be divided into layers. Structural conditions on those systems are presented that imply confluence. The approach covers known results like many-sorted persistency, layer-preservation and currying. For order-sorted persistency, appropriate conditions are developed.

Nowadays, for the automated search for termination or for confluence proofs there exist more and more already quite powerful tools that can be used [80,10]. In particular, using SAT/SMT search based procedures for many tasks has become a standard practical way of development, progress and increased performance.

Relative termination tools together with the decreasing diagrams technique appear to be very powerful for practically verifying confluence of TRSs. This does not have a direct connection to modularity, but the combination of the techniques (in the sense of divide-and-conquer) may be very promising. Another trend in research on automated termination proving is to tackle termination of programs in different programming languages, via reduction to termination problems in rewriting. These transformational techniques are already quite sophisticated and successful in various cases, but may also be unsatisfactory due to the encoding character of the transformational approach. Thus, a general research question is to investigate how direct approaches to develop a theory on modularity/persistency in extended formats of rewriting

[21] One easy such result which is an implicit consequence of known results is the following: If, for two terminating left-linear disjoint TRSs the union is non-terminating, then one of the systems must be CONS$^{\rightarrow}$, but not the other one.

[22] Note that most existing modularity/persistency criteria heavily rely on syntactical properties of the involved TRSs. Only slight (even equivalence preserving) changes in the representation of the systems may invalidate the preservation properties. This task appears to be particularly challenging, and perhaps even unrealistic.

[23] Since the basic theory in these extended rewriting formats is by far not as developed as in ordinary first-order term rewriting and since some of the basic theory of term rewriting does not extend easily, this also appears to be challenging.

(e.g. context-sensitive, conditional, higher-order) compare to transformational approaches. This question is of course not restricted to modularity issues, but applies to other areas (like the dependency pair framework) as well.

Papers like [14] and [32,33] seem to be instances of the attempt to converge towards a maximally abstract framework for modularity that really captures the essence of what is needed conceptually and technically. In this direction the author expects more to come.

Acknowledgements

The author is grateful to the anonymous reviewers for their detailed criticisms and comments. Thanks also to Hans Zantema for the pointer to [73].

References

[1] T. Aoto, Solution to the problem of Zantema on a persistent property of term rewriting systems, Journal of Functional and Logic Programming 2001 (11) (2001).
[2] T. Aoto, Y. Toyama, Extending persistency of confluence with ordered sorts, Technical Report IS-RR-96-0025F, School of Information Science, JAIST, 1996.
[3] T. Aoto, Y. Toyama, Persistency of confluence, Journal of Universal Computer Science 3 (11) (1997) 1134–1147.
[4] T. Aoto, Y. Toyama, A reduction-preserving completion for proving confluence of non-terminating term rewriting systems, Logical Methods in Computer Science 8 (1) (2012).
[5] C. Appel, V. van Oostrom, J. G. Simonsen, Higher-order (non-)modularity, in: C. Lynch (Ed.), Proc. 21st RTA, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 6, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010, pp. 17–32.
[6] F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.
[7] L. Bachmair, N. Dershowitz, Commutation, transformation, and termination, in: J. Siekmann (Ed.), Proc. 8th CADE, in: LNCS, vol. 230, Springer-Verlag, 1986, pp. 5–20.
[8] J. Bergstra, J. Klop, A. Middeldorp, Termherschrijfsystemen. Deventer, 1989 (In Dutch).
[9] M. Bezem, J. Klop, R. Vrijer (Eds.), Term Rewriting Systems, in: Cambridge Tracts in Theoretical Computer Science, vol. 55, Cambridge University Press, 2003.
[10] Confluence competition 2012. http://coco.nue.riec.tohoku.ac.jp/2012/.
[11] N. Dershowitz, Innocuous constructor-sharing combinations, in: H. Comon (Ed.), Proc. 8th RTA, in: LNCS, vol. 1232, Springer-Verlag, 1997, pp. 202–216.
[12] K. Drosten, Termersetzungssysteme, in: Informatik-Fachberichte, 210, Springer-Verlag, 1989 (In German).
[13] I. Durand, A. Middeldorp, On the modularity of deciding call-by-need, in: Proc. 4th FoSSaCS, in: LNCS, vol. 2030, Springer-Verlag, 2001, pp. 199–213.
[14] B. Felgenhauer, H. Zankl, A. Middeldorp, Layer systems for proving confluence, in: S. Chakraborty and A. Kumar (Eds.), Proc. FSTTCS, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 13, Dagstuhl, Germany, 2011, pp. 288–299.
[15] W. Fokkink, J. Kamperman, P. Walters, Lazy rewriting on eager machinery, ACM Transactions on Programming Languages and Systems (TOPLAS) 22 (1) (2000) 45–86.
[16] A. Geser, Relative termination, Ph.D. Thesis, Universität Ulm, 1990. reprinted in: Ulmer Informatik-Berichte Nr. 91-03, 1991.
[17] J. Giesl, P. Schneider-Kamp, R. Thiemann, Automatic termination proofs in the dependency pair framework, in: U. Furbach, N. Shankar (Eds.), 3rd IJCAR, in: LNCS, vol. 4130, Springer-Verlag, 2006, pp. 281–286.
[18] B. Gramlich, Sufficient conditions for modular termination of conditional term rewriting systems, in: M. Rusinowitch, J.-L. Rémy (Eds.), Proc. 3rd CTRS, 1992, in: LNCS, vol. 656, Springer-Verlag, 1993, pp. 128–142.
[19] B. Gramlich, Generalized sufficient conditions for modular termination of rewriting, Applicable Algebra in Engineering, Communication and Computing 5 (1994) 131–158.
[20] B. Gramlich, Abstract relations between restricted termination and confluence properties of rewrite systems, Fundamenta Informaticae 24 (1995) 3–23.
[21] B. Gramlich, Termination and confluence properties of structured rewrite systems, PhD Thesis, Universität Kaiserslautern, 1996.
[22] B. Gramlich, Conditional rewrite systems under signature extensions: Some counterexamples, in: K. Schulz and S. Kepser, (Eds.), Proc. 10th Int. Workshop on Unification (Extended Abstracts), CIS-Bericht-96-91, Universität München, 1996, pp. 45–50.
[23] B. Gramlich, On proving termination by innermost termination, in: H. Ganzinger (Ed.), Proc. 7th RTA, in: LNCS, vol. 1103, Springer-Verlag, 1996, pp. 93–107.
[24] B. Gramlich, On termination and confluence properties of disjoint and constructor-sharing conditional rewrite systems, Theoretical Computer Science 165 (1) (1996) 97–131.
[25] B. Gramlich, Modular aspects of rewrite-based specifications, in: F. Parisi-Presicce (Ed.), Recent Trends in Algebraic Development Techniques, 12th Int. Workshop, WADT 1997, Selected Papers, in: LNCS, vol. 1376, Springer-Verlag, 1998, pp. 253–268.
[26] B. Gramlich, K. Györgyfalvay, A note on minimal counterexamples to modularity of termination. Technical Report E1852-2012-02, Faculty of Informatics, TU Wien, Austria, 2012.
[27] B. Gramlich, K. Györgyfalvay, On modularity of termination properties of rewriting under position-based strategies. Technical Report, Faculty of Informatics, TU Wien, Austria, 2012 (Forthcoming).
[28] B. Gramlich, K. Györgyfalvay, On modularity of termination properties of rewriting under strategies, in: G. Moser, (ed.), Proc. 12th WST (Extended Abstracts), Feb. 2012, pp. 59–63.
[29] B. Gramlich, S. Lucas, Modular termination of context-sensitive rewriting, in: C. Kirchner (Ed.), Proc. 4th PPDP, ACM Press, 2002, pp. 50–61.
[30] B. Gramlich, S. Lucas, Simple termination of context-sensitive rewriting, in: B. Fischer, E. Visser (Eds.), Proc. 3rd RULE, ACM Press, 2002, pp. 29–41.
[31] B. Gramlich, F. Schernhammer, Extending context-sensitivity in term rewriting, in: Proc. 9th WRS, in: EPTCS, vol. 15, 2009, pp. 56–68.
[32] J.-P. Jouannaud, Modular Church-Rosser modulo, in: F. Pfenning (Ed.), Proc. 17th RTA, in: LNCS, vol. 4098, Springer-Verlag, 2006, pp. 96–107.
[33] J.-P. Jouannaud, Y. Toyama, Modular Church-Rosser modulo: The complete picture, International Journal of Software and Informatics 2 (1) (2008) 61–75.
[34] J. W. Klop, A. Middeldorp, Y. Toyama, R. Vrijer, Modularity of confluence: A simplified proof, Information Processing Letters 49 (1994) 101–109.
[35] M. Krishna Rao, Completeness of hierarchical combinations of term rewriting systems, in: R. Shyamasundar (Ed.), Proc. 13th FSTTCS, in: LNCS, vol. 761, Springer-Verlag, 1993, pp. 125–138.
[36] M. Krishna Rao, Simple termination of hierarchical combinations of term rewriting systems, in: Proc. STACS, in: LNCS, vol. 789, Springer-Verlag, 1994, pp. 203–223.
[37] M. Krishna Rao, Modular proofs for completeness of hierarchical term rewriting systems, Theoretical Computer Science 151 (2) (1995) 487–512.
[38] M. Krishna Rao, Semi-completeness of hierarchical and super-hierarchical combinations of term rewriting systems, in: P. Mosses, M. Nielsen, M. Schwartzbach (Eds.), Proc. 6th TAPSOFT, in: LNCS, vol. 915, Springer-Verlag, 1995, pp. 379–393.
[39] M. Krishna Rao, Modularity of termination in term graph rewriting, in: H. Ganzinger (Ed.), Proc. 7th RTA, in: LNCS, vol. 1103, Springer-Verlag, 1996, pp. 230–244.

[40] J. Kruskal, Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture, Transactions of the American Mathematical Society 95 (1960) 210–225.
[41] M. Kurihara, I. Kaji, Modular term rewriting systems: termination, confluence and strategies, IEICE (1988) 57–66. Tech. Report COMP88 (143).
[42] M. Kurihara, I. Kaji, Modular term rewriting systems and the termination, Information Processing Letters 34 (1990) 1–4.
[43] M. Kurihara, A. Ohuchi, Modularity of simple termination of term rewriting systems, Journal of IPS, Japan 34 (1990) 632–642.
[44] M. Kurihara, A. Ohuchi, Termination of combination of composable term rewriting systems, in: Proc. 7th AJCAI (Australian Joint Conference on Artificial Intelligence), 1994, pp. 227–234.
[45] M. Kurihara, A. Ohuchi, Modularity in noncopying term rewriting, Theoretical Computer Science 152 (1) (1995) 139–169.
[46] S. Lucas, Context-sensitive computations in functional and functional logic programs, Journal of Functional and Logic Programming 1998 (1) (1998).
[47] S. Lucas, Context-sensitive rewriting strategies, Information and Computation 178 (1) (2002) 294–343.
[48] S. Lucas, Lazy rewriting and context-sensitive rewriting, in: Proc. 10th WFLP, 2001, ENTCS 64 (2002) 234–254. Selected Papers.
[49] C. Lüth, Compositional term rewriting: An algebraic proof of Toyama's theorem, in: H. Ganzinger (Ed.), Proc. 7th RTA, in: LNCS, vol. 1103, Springer-Verlag, 1996, pp. 261–275.
[50] M. Marchiori, Modularity of completeness revisited, in: J. Hsiang (Ed.), Proc. 6th RTA, in: LNCS, vol. 914, Springer-Verlag, 1995, pp. 2–10.
[51] M. Marchiori, On the modularity of normal forms in rewriting, Journal of Symbolic Computation 22 (2) (1996) 143–154.
[52] M. Marchiori, The theory of vaccines, in: P. Degano, R. Gorrieri, A. Marchetti-Spaccamela (Eds.), Proc. 24th ICALP, in: LNCS, vol. 1256, Springer-Verlag, 1997, pp. 660–670.
[53] A. Middeldorp, Modular aspects of properties of term rewriting systems related to normal forms, in: N. Dershowitz (Ed.), Proc. 3rd RTA, in: LNCS, vol. 355, Springer-Verlag, 1989, pp. 263–277.
[54] A. Middeldorp, A sufficient condition for the termination of the direct sum of term rewriting systems, in: Proc. 4th LICS, 1989, pp. 396–401.
[55] A. Middeldorp, Modular properties of term rewriting systems, Ph.D. Thesis, Free University, Amsterdam, 1990.
[56] A. Middeldorp, Modular properties of conditional term rewriting systems, Information and Computation 104 (1) (1993) 110–158.
[57] A. Middeldorp, Completeness of combinations of conditional constructor systems, Journal of Symbolic Computation 17 (1994) 3–21. Prelim. version in M. Rusinowitch and J.-L. Rémy, (Eds.) Proc. 3rd CTRS, 1992, LNCS 656, Springer-Verlag, 1993, pp. 82–96.
[58] A. Middeldorp, H. Zantema, Simple termination of rewrite systems, Theoretical Computer Science 175 (1997) 127–158.
[59] T. Nagaya, Y. Toyama, Decidability for left-linear growing term rewriting systems, Information and Computation 178 (2) (2002) 499–514.
[60] E. Ohlebusch, Combinations of simplifying conditional term rewriting systems, in: M. Rusinowitch, J. Remy (Eds.), Proc. 3rd CTRS, 1992, in: LNCS, vol. 656, Springer-Verlag, 1993, pp. 113–127.
[61] E. Ohlebusch, Modular properties of composable term rewriting systems, Ph.D. Thesis, Universität Bielefeld, 1994. Report 94-01.
[62] E. Ohlebusch, On the modularity of confluence of constructor-sharing term rewriting systems, in: S. Tison (Ed.), Proc. 19th CAAP, in: LNCS, vol. 787, Springer-Verlag, 1994, pp. 261–275.
[63] E. Ohlebusch, On the modularity of termination of term rewriting systems, Theoretical Computer Science 136 (1994) 333–360.
[64] E. Ohlebusch, Modular properties of composable term rewriting systems, Journal of Symbolic Computation 20 (1) (1995) 1–42.
[65] E. Ohlebusch, Modular properties of constructor-sharing conditional term rewriting systems, in: N. Dershowitz, N. Lindenstrauss (Eds.), Proc. 4th CTRS, 1994, in: LNCS, vol. 968, Springer-Verlag, 1995, pp. 296–315.
[66] E. Ohlebusch, Termination is not modular for confluent variable-preserving term rewriting systems, Information Processing Letters 53 (1995) 223–228.
[67] E. Ohlebusch, Advanced Topics in Term Rewriting, Springer-Verlag, 2002.
[68] E. Ohlebusch, Hierarchical termination revisited, Information Processing Letters 84 (4) (2002) 207–214.
[69] D. Plump, Implementing term rewriting by graph reduction: termination of combined systems, in: S. Kaplan, M. Okada (Eds.), Proc. 2nd CTRS, 1990, in: LNCS, vol. 516, Springer-Verlag, 1991, pp. 307–317.
[70] D. Plump, Collapsed tree rewriting: completeness, confluence, and modularity, in: M. Rusinowitch, J.-L. Remy (Eds.), Proc. 3rd CTRS, 1992, in: LNCS, vol. 656, 1993, pp. 97–112.
[71] J. Raoult, J. Vuillemin, Operational and semantic equivalence between recursive programs, Journal of the ACM 27 (4) (1980) 772–796.
[72] M. Rusinowitch, On termination of the direct sum of term rewriting systems, Information Processing Letters 26 (1987) 65–70.
[73] M. Sakai, Innermost terminating right-linear overlay term rewriting systems are terminating. Unpublished manuscript, http://www.sakabe.nuie.nagoya-u.ac.jp/~sakai/papers/sin-sn-031015.pdf, 2003.
[74] F. Schernhammer, B. Gramlich, Termination of lazy rewriting revisited, in: Jürgen Giesl (Ed.), Final Proc. 7th WRS, 2007, ENTCS 204 (2008), 35–51.
[75] M. Schmidt-Schauß, Unification in a combination of arbitrary disjoint equational theories, Journal of Symbolic Computation 1 (1989) 51–99.
[76] M. Schmidt-Schauß, M. Marchiori, S. Panitz, Modular termination of r-consistent and left-linear term rewriting systems, Theoretical Computer Science 149 (2) (1995) 361–374.
[77] J. G. Simonsen, On the modularity of confluence in infinitary term rewriting, in: V. van Oostrom (Ed.), Proc. 15t RTA, in: LNCS, vol. 3091, Springer-Verlag, 2004, pp. 185–199.
[78] J. G. Simonsen, On modularity in infinitary term rewriting, Information and Computation 204 (6) (2006) 957–988.
[79] C. Sternagel, R. Thiemann, Signature extensions preserve termination - an alternative proof via dependency pairs, in: A. Dawar, H. Veith (Eds.), Proc. 24th CSL, in: LNCS, vol. 6247, Springer-Verlag, 2010, pp. 514–528.
[80] Termination competition 2012. http://termination-portal.org/wiki/Termination_Competition_2012.
[81] Y. Toyama, Counterexamples to termination for the direct sum of term rewriting systems, Information Processing Letters 25 (1987) 141–143.
[82] Y. Toyama, On the Church-Rosser property for the direct sum of term rewriting systems, Journal of the ACM 34 (1) (1987) 128–143.
[83] Y. Toyama, Commutativity of term rewriting systems, in: K. Fuchi, L. Kott (Eds.), Proc. FGCS (Programming of Future Generation Computer), vol. II, North-Holland, 1988, pp. 393–407.
[84] Y. Toyama, Confluent term rewriting systems, in: J. Giesl (Ed.), Proc. 16th RTA, in: LNCS, vol. 3467, Springer-Verlag, 2005, page 1.
[85] Y. Toyama, J. Klop, H. Barendregt, Termination for direct sums of left-linear complete term rewriting systems, Journal of the ACM 42 (6) (1995) 1275–1304.
[86] J. van de Pol, Modularity in many-sorted term rewriting systems. Master's Thesis, Utrecht University, 1992.
[87] V. van Oostrom, Confluence by decreasing diagrams, Theoretical Computer Science 121 (1994) 259–280.
[88] V. van Oostrom, Random descent, in: F. Baader (Ed.), Proc. 18th RTA, in: LNCS, vol. 4533, Springer-Verlag, 2007, pp. 314–328.
[89] V. van Oostrom, Confluence by decreasing diagrams — converted, in: A. Voronkov (Ed.), Proc. 19th RTA, in: LNCS, vol. 5117, Springer-Verlag, 2008, pp. 306–320.
[90] V. van Oostrom, Modularity of confluence constructed, in: A. Armando, P. Baumgartner, G. Dowek (Eds.), Proc. 4th IJCAR, in: LNCS, vol. 5195, Springer-Verlag, 2008, pp. 348–363.
[91] H. Zankl, M. Korp, Modular complexity analysis via relative complexity, in: Proc. 21st RTA, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 6, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010, pp. 385–400.
[92] H. Zantema, Termination of term rewriting: interpretation and type elimination, Journal of Symbolic Computation 17 (1994) 23–50.
[93] H. Zantema, Personal communication, 2012.