Counterexamples to Higher-Order Modularity

October 31, 2005

Abstract

It is shown that, contrary to what popular believe would have it, almost none of the usual rewrite properties is modular for higher-order rewriting.

1 Introduction

A property of rewrite systems is said *modular* when it holds for the disjoint union of two systems if and only if it holds for the systems individually. That is, letting $\mathcal{T}_1 \uplus \mathcal{T}_2 = (\Sigma_1 \uplus \Sigma_2, R_1 \uplus R_2)$ denote the disjoint union of the rewrite systems $\mathcal{T}_i = (\Sigma_i, R_i)$ for $i \in \{1,2\}$, a property P is *modular* if $P(\mathcal{T}_1 \uplus \mathcal{T}_2) \Leftrightarrow P(\mathcal{T}_1)$ & $P(\mathcal{T}_2)$. We show that almost none of the usual properties considered in rewriting such as termination, confluence, or uniqueness of normal forms, is modular for higher-order rewriting, be it for Nipkow's higher-order Pattern Rewrite Systems (PRSs), Klop's Combinatory Reduction Systems (CRSs), or for Yamada's Simply Typed Term Rewriting Systems (STTRSs).

Thoughout we will employ notions, notations and results which can be found in the book *Term Rewriting Systems*, Terese, CUP 2003. We employ x, y, z to range over term variables of base type. This is in concordance with the types variables in first-order term rewrite system (TRSs) get when they are embedded into Nipkow's higher-order pattern term rewrite systems (PRSs). We employ Z, W, V to range over meta-variables which yield a term of base type when supplied with an arbitrary but fixed number of terms of base type. This is in concordance with the types meta-variables in Klop's combinatory reduction systems (CRSs) get when embedded into PRSs.

2 Counterexamples to modularity for PRSs

Termination To set the stage, we briefly recapitulate Toyama's classical counterexample to modularity of termination for TRSs.

Counterexample 1. The single rule TRS

$$f(a,b,x) \to f(x,x,x)$$

is terminating as is routinely shown by the nowadays available termination prooftools. However, combining it with the, also trivially terminating, two-rule TRS

$$\begin{array}{lll} g(x,y) & \to & x \\ g(x,y) & \to & y \end{array}$$

yields a non-terminating system as witnessed by repeating the cycle

$$\underline{f(a,b,g(a,b))} \to f(\underline{g(a,b)},g(a,b),g(a,b)) \to f(a,\underline{g(a,b)},g(a,b)) \to f(a,b,g(a,b))$$

As a first approximation one may say that termination of the first TRS relies on the absence of a term which reduces both to a and b, a feat destroyed by the second TRS by its ability to encode non-deterministic choice. In depth analyses have been performed by, among others, Middeldorp, Ohlebusch, and Gramlich.

Confluence Toyama also obtained the first major positive modularity result for TRSs, modularity of confluence. A simpler proof was given later by him, Middeldorp, Klop, and de Vrijer. In his PhD thesis, Klop showed that confluence is not a modular property for CRSs. His counterexample involved on the one hand a non-left-linear first-order rule and on the other hand the β-rule of the λ-calculus. In the following, the λ-calculus is replaced by the μ-rule, avoiding the encoding of recursion via the fixed-point combinator needed in case of the former, which only distracts from the issue.

Counterexample 2. The first-order TRS consisting of the following two rules

$$\begin{array}{rll} f(x,x) & \to & a \\ f(x,s(x)) & \to & b \end{array}$$

is confluent by Huet's Critical Pair Lemma, since it is terminating and has no critical pairs. However, adjoining the orthogonal hence confluent, single-rule PRS

$$\mu(x.Z(x)) \to Z(\mu(x.Z(x)))$$

yields a non-confluent system as witnessed by

$$a \leftarrow f(\mu(x.s(x)),\mu(x.s(x))) \to f(\mu(x.s(x)),s(\mu(x.s(x)))) \to b$$

As a first approximation one may say that confluence of the TRS relies both on termination and on the absence of a critical pair between the two rules, which in turn relies on non-left-linearity and non-convertibility of t and $s(t)$ for any term t. Both feats are destroyed by the PRS by its ability to encode recursion, as witnessed by taking $t = \mu(x.s(x))$.

Uniqueness of normal forms Middeldorp gave a reduction of modularity of uniqueness of normal forms to modularity of confluence, for first-order TRSs. By the previous subsection such a reduction would be useless in the higher-order case. In fact, uniqueness of normal forms is not modular as shown by the same example employed there: Since the rewrite systems are confluent they both have the uniqueness of normal forms property, but the terms a and b are distininct convertible normal forms in the disjoint union of the TRS and the PRS.

Left-linear completeness Toyama, Klop, and Barendregt showed that left-linearity and confluence suffice to turn termination into a modular property for first-order TRSs. A simpler proof (yielding a slightly stronger result) was later found by Marchiori, Schmidt-Schauß and Panitz. For PRSs left-linear completeness is not modular.

Counterexample 3. Consider a PRS consisting of the single rule

$$f(x.x, xy.Z(x,y)) \to g(Z(a, f(x.Z(x,a), xy.Z(x,y))))$$

where f and g are second-order symbols and a is a first-order symbol. The PRS is confluent, since it is orthogonal, i.e. its only rule is left-linear and non(-self)-overlapping. It is terminating as well, as shown below. However, combining it with the trivially left-linear and complete TRS consisting of the single rule

$$h(x,y) \to x$$

yields a non-terminating combination as witnessed by

$$\underline{f(x.x, xy.h(x,y))} \to g(h(a, f(x.\underline{h(x,a)}, xy.h(x,y)))) \to g(h(a, f(x.x, xy.h(x,y))))$$

noting that it is of the form $t \twoheadrightarrow g(h(a,t))$ for $t = f(x.x, xy.h(x,y))$, thus giving rise to an infinite fixed-point reduction.

As a first approximation one may say termination of the PRS relies on its left-hand side not being embeddable in its right-hand side; if it were then the term substituted for $Z(x,a)$ should be reducible to and therefore identical to x, which would cause the embedded subterm of the rhs to be erased. As is well-known, termination which relies on non-embeddability is easily destroyed by adjoining projection rules, which is indeed the only, but fatal, feat the TRS brings.

To show termination of the PRS, the claim that the PRS does not allow creation of redexes along any reduction, is shown, from which termination follows by the Finite Developments Theorem. To prove the claim, distinguish cases on whether or not the head symbol f of a hypothetical redex in s, has an origin in t, for a given step $t \to s$.

- If it does, then it must have been either in the context part or the substitution part of the step.

  In the former case, f can only not have been the head symbol of a redex in t, if part of the left-hand side is created by the step. This is impossible, since then certainly the outermost symbol g of the right-hand side should be part of the pattern, which in fact it cannot be.

  In the latter case, note that the only possible change the substitution part undergoes in the right-hand side compared to the left-hand side, is the replacement of its formal parameters x and y by actual terms. For the first occurrence of Z, these terms are a and some term having f as head-symbol respectively, neither of which cannot create a redex since a nor f occur in the left-hand side. For the second occurrence of Z, the actual terms are x and a respectively. For the latter we reason as before, while the former will be free in the substitution part hence cannot be part of the pattern. For the third occurrence of Z, we reason in the same way.

- If it does not, then it must be a copy of the occurrence of f in the right-hand side. But that cannot be since it would enforce f to be erased as argued above.

Normalisation Normalisation is modular in the first-order case. Grue Simonsen raised the question if this extends to the higher-order case. It doesn't.

Counterexample 4. The two rule PRS specified by

$$\begin{array}{lcl} f(x.Z(x), y.y) & \to & f(x.Z(x), y.Z(Z(y))) \\ f(x.x, y.Z(y)) & \to & a \end{array}$$

is normalising as shown below. However, combining it with the single-rule TRS

$$g(g(x)) \to x$$

which is trivially normalising, yields a system which is not, as witnessed by the cycle

$$f(x.g(x), y.y) \leftrightarrow f(x.g(x), y.g(g(y)))$$

As in the previous subsection, normalisation of the PRS relies essentially on the lhs of its first rule to be non-embeddable into its rhs; if it were embeddable the term substituted for $Z(Z(y))$ should be reducible to and therefore identical to y, but then the second rule would have been applicable to its lhs as well, causing normalisation nonetheless. As in the previous subsection, non-embeddability is destroyed by adjoining the projection rule of the TRS.

Normalisation of the PRS is shown by induction on terms. The only interesting cases are terms of the form $f(x.t, y.s)$ where t and s are normalising by the induction hypothesis. Thus we may assume w.l.o.g. that t and s are in fact in normal form. To show normalisation, distinguish cases on the shapes of t and s.

- If t is equal to x, then the term reduces to the normal form a.
- If t is distinct from x and s is distinct from y, then the term is in normal form itself already.
- If t is distinct from x and s is equal to y, the term reduces in one step to

  $$f(x.t, y.t[x := t[x := y]])$$

  By the assumption that t is distinct from x, the term in the first argument of the reduct is distinct from x and the term in the second argument is distinct from y, so neither of the rules is applicable at the head.

  By the assumption that t is in normal form, both the terms in the first and second arguments are so as well, since looking at the left-hand sides of the rules one notes that the only way a redex could be created by a substitution requires that one substitutes the bound variable (x or y, respectively), which cannot be.

Orthogonal Acyclicity Klop raised the question whether acyclicity, i.e. the absence of non-empty reductions $t \twoheadrightarrow t$ from a term to itself, is modular for orthogonal rewrite systems. In the higher-order case it isn't.

Counterexample 5. The PRS consisting of the single rule

$$f(xyz.Z(x, y, z), W, V) \to Z(W, Z(V, W, f(xyz.Z(x, y, z), W, V)), V)$$

is shown to be acyclic below. Combining the PRS with the two-rule TRS

$$\begin{array}{lcl} g(a, x, y) & \to & x \\ g(b, x, y) & \to & y \end{array}$$

which is trivially acyclic, yields a cyclic system, as witnessed by

$$\begin{array}{ll} \multicolumn{2}{l}{\underline{f(xyz.g(x,y,z),a,b)}} \\ \to & \underline{g(a,g(b,a,f(xyz.g(x,y,z),a,b)),b)} \\ \to & \underline{g(b,a,f(xyz.g(x,y,z),a,b))} \\ \to & f(xyz.g(x,y,z),a,b) \end{array}$$

The main feat of the PRS is that the lhs of its rule is embeddable in its rhs, but only so in a non-empty context. If it were the case the context could be empty, then the term substituted for Z should collapse both to its second (y) and third (z) argument *only* depending on its first (x) argument, which is impossible. It is exactly this feature which the two selection rules of the TRS bring.

The PRS is proven acyclic by contradiction. To that end, suppose a non-empty cycle σ on t were to exist. W.l.o.g. we may assume such a t to be of minimal size among all terms admitting a non-empty cycle. Moreover, we may assume by the Standardisation Theorem for left-linear PRSs, that σ is standard.

We will employ the notion of *gripping* path due to Melliès. Say a position of an f-symbol in a term *grips* any position of an f-symbol which has a variable bound by the former below it. Then a gripping path is a path w.r.t.. the gripping relation. Note that gripping paths are finite since successive positions are properly ordered by prefix. The property to be exploited is that gripping paths are preserved under taking their (position-wise) origin along any rewrite step. The point is that if the path in the target of a step has some position below the contracted redex—which is the only interesting case—then the first such must either be the position of the head-symbol of the copy of the redex, or be in one of the copies of (the terms substituted for) the meta-variables in the rhs. In either case, the path must proceed completely inside that copy, by the variable convention. But then a corresponding path exists in the source of the step through the corresponding copy.

Now consider the origin of an arbitrary gripping path from the root of t along σ. This origin is again a gripping path in t by the above and by σ being a cycle. Distinguish cases depending on whether its first position, say p, is on some gripping path from the root of t or not.

If it is not, then consider the position q which is closest to the root and above p such that it is not on some gripping path from the root. Per construction, the subterm headed by q, say t', contains no variable bound by an f-symbol above it. Since p descends, also per construction, to the root of t along σ which was assumed to be standard, it follows that t reduces to t' somewhere along σ. Thus there would be a cylce on t' as well, contradicting minimality of t.

If it is, then a further case distinction is made depending on whether p is itself the root or it isn't.

If p is not the root, a contradiction is obtained since then for any gripping path from the root in t, one obtains a longer such path by prefixing its origin along σ by the (non-empty) gripping path from the root to p.

If p is the root, then first note that by the minimality assumption for t, σ must contain some head step. By standardness and the shape of the lhs—which is a pattern which cannot be created—also the first step of σ must be a head step, i.e.

$$\sigma : f(xyz.t', s, u) \to t'[x, y, z := s, t'[x, y, z := u, s, t], u] \twoheadrightarrow f(xyz.t', s, u)$$

and the root descends to the head f-symbol of a fresh copy of t in the rhs. Both y and z must occur in t' otherwise t would be erased. Therefore. by reasoning as above, the (instantiated) copies of t' above the f-symbol must be collapsable to their second respectively third arguments, depending on the respective substitutions s and u for its first argument. However, neither s nor u contains a variable bound outside it, and thus a collapsing reduction from t' could not depend on them. Contradiction.

The proof method employed is less *ad hoc* then one maybe would surmise. For instance, by analogous (technical) reasoning one can prove that a term can only be cyclic w.r.t. the earlier μ-rule if its body is identical to its bound variable.

3 Counterexamples for CRS signature extension

It is easy to check that all the counterexamples to modularity for PRSs presented in the previous section go through for CRSs; all meta-variables occurring in the rules have order at most two. Therefore, neither termination nor confluence, uniqueness of normal forms, left-linear completeness, normalisation, or orthogonal acyclicity is modular for CRSs. Even stronger, it is not immediate they are preserved under *signature extensions*, i.e. under taking disjoint unions where one of the rewrite systems has no rules. Indeed, most are not.

Termination and Normalisation For TRSs termination is preserved under signature extension, as follows by an easy induction on the rank of terms since the fresh function symbols partition any term in the disjoint union into terminating components. For PRSs termination is preserved under signature extension, as follows by replacing any fresh function symbol by a fresh variable of the same type. This fails for (functional) CRSs.

Counterexample 6. The CRS having only a unary function symbol f, and rule

$$f([x][y]Z(x,y)) \quad \to \quad Z([x][y]Z(y,x),[x]x)$$

is shown terminating below. Extending the signature with a binary symbol g yields:

$$\begin{array}{rcl} f([x][y]g(f(x),f(y))) & \to & g(f([x][y]g(f(y),f(x))),f([x]x)) \\ & \to & g(f([x]x),\underline{g(f([x][y]g(f(x),f(y))),f([x]x))}) \end{array}$$

showing non-preservation of termination.

Noting that the rewrite relation for CRSs is defined on *terms* not on *meta-terms*, termination of the first CRS is seen to rely on the absence of function symbols having arity greater than one, a feat destroyed by extending the signature with the binary function symbol g.

Since the CRS is orthogonal and its steps are non-erasing, to prove termination it suffices to prove normalisation.

This is proven by induction on the term t, the only interesting case being when it is of shape $f([x][y]t')$ with t' in normal form. Distinguish cases on whether x or y occurs free in t'.

In the case that neither x nor y occurs free in t', the term t simply reduces to the normal form t' by contracting the root redex.

If x occurs free in t', then y cannot occur free too, since the CRS does not contain function symbols of arity greater than one. Thus, contracting the root redex yields $t'[x := [x][y]t'[x := y]]$. This term can only be not in normal form, if t' is of shape $C[f(x)]$ for some context C in normal form, giving $C[f([x][y]C[f(y)])]$ which reduces to $C[C[f([x]x)]]$, which normalises by the first case.

If y occurs free in t'm then x cannot occur free too, since the CRS does not contain function symbols of arity greater than one. Thus, contracting the root redex yields $t'[y := [x]x]$ which is a normal form since t' is.

Since the CRS is non-erasing termination and normalisation are equivalent for it, so normalisation is not preserved under signature extension either.

For termination, the above really hinges on the absence of a symbol with arity greater than one: If, *e.g.*, a binary symbol were present (as is the case in applicative CRSs), any non-terminating reduction in the CRS with extended signature could be simulated by replacing each fresh n-ary function symbol by $n-1$ copies of the binary symbol and $[x]x$ once. Thus termination is preserved under signature extension for almost any CRS occurring in practice.

For normalisation, things are not as pleasant. For instance, adjoining a binary symbol h with rule $h(x,y) \to x$, normalisation of the resulting CRS follows from that of the original CRS, by first maximally applying the new rule. Still, normalisation is not preserved under extending the signature with g.

Confluence and uniqueness of normal forms Both tor TRSs and PRSs, confluence is preserved under signature extension. For the former this is just a special case of modularity of confluence. For the latter it follows by 'viewing' each fresh function symbol as a variable, as in the previous subsection. For CRSs confluence is not preserved.

Counterexample 7. The CRS given by the rules

$$\begin{array}{rcl} f(f(W)) & \to & f(W) \\ f([x]Z(x)) & \to & f(Z(a)) \\ f([x]Z(x)) & \to & f([x]Z(Z(x))) \end{array}$$

is shown confluent below, but is not so after extending the signature with a unary g:

$$f(g(a)) \leftarrow f([x]g(x)) \to f([x]g(g(x))) \to f(g(g(a)))$$

showing non-preservation of confluence.

The counterexample shows that a CRS can be confluent, i.e. confluent on *closed* terms, but not *meta*-confluent, i.e. not confluent on *meta*-terms. This is analogous to the fact that a TRS can be *ground* confluent, i.e. confluent on *closed* terms, but not confluent, i.e. not confluent on *open* terms.

To show confluence of the CRS, consider its first two rules which are length-decreasing so terminating. By Nipkow's higher-order Critical Pair Lemma—-which also applies to CRSs—and Newman's Lemma it suffices to show its critical pair

$$f([x]Z(x)) \leftarrow f(f([x]Z(x))) \to f(f(Z(a)))$$

can be joined in a common reduct; it can by performing a single step from either term to the common reduct $f(Z(a))$. The third rule is shown redundant by proving that its lhs is convertible to its rhs. This is clear in case x does not occur in the term substituted for Z. Otherwise, from applying the second rule to either side one sees that it suffices to show that any term $f(C[a])$ with C a unary context, reduces to $f(a)$. This is shown by induction on the size of C, applying either the first or the second rule, depending on whether the head-symbol of C is either f or $[x]$, respectively.

By the same example it follows that uniqueness of normal forms is not preserved under signature extension either.

Left-linear completeness Both for TRSs and PRSs, left-linear completeness is preserved under signature extension. For the former this is just a special case of modularity of left-linear completeness. For the latter, it follows again by viewing the fresh function symbols as variables. For CRSs left-linear completeness is not preserved: The counterexample against preservation of termination above is orthogonal, hence left-linear and confluent, so works here as well.

Orthogonal acyclicity Orthogonal acyclicity *is* preserved under signature extension. To see this, consider a hypothetical term t of minimal size in the extended signature, allowing a cycle σ. Consider transforming σ by projecting each function symbol in the extended to its first argument, i.e. by normalising w.r.t. rules of the form $f(\vec{x}) \to x_1$ for f in the extended signature, and by replacing any constant a in the extended signature by $[x]x$. Since these rules constitute an orthogonal CRS themselves, which moreover is orthogonal to the original rules, the transformed reduction cycle is a reduction cycle in the original CRS. By minimality of t, σ contains a head-step, and the head-symbol of t must be in the original signature, so transforming σ yields a reduction cycle which is non-empty. Contradiction.

4 Counterexamples to modularity for STTRSs

The counterexample to modularity for PRSs carry over to STTRSs, but need to be adapted to the fact that the latter do not allow for λ-abstractions. Of course, Toyama's counterexample to modularity of termination for TRSs carries over immediately, since each TRS can be viewed as an STTRS by assigning appropriate types to the function symbols, *e.g.*, the 'product' type $o \times o \to o$ to binary function symbols. Failure of modularity of the other properties is due to the presence of terms of 'function' type, allowing for partial applications.

Confluence and uniqueness of normal forms The μ-counterexample to modularity of confluence for PRSs can be casted without problems to STTRSs.

Counterexample 8. Taking the disjoint union of the confluent STTRS

$\mu Z \to Z(\mu Z)$

where Z has type $o \to o$, with the confluent two-rule STTRS

$$\begin{array}{rcl} fxx & \to & a \\ fx(sx) & \to & b \end{array}$$

yields a non-confluent system as witnessed by

$$a \leftarrow f(\mu s)(\mu s) \to f(\mu s)(s(\mu s)) \to b$$

The same example shows that uniqueness of normal forms is not modular.

Left-linear completeness, normalisation, and orthogonal acyclicity The counterexamples to modularity of the three properties in the title of this subsection for PRSs, can be simplified into a single one for STTRSs.

Counterexample 9. Taking the disjoint union of the STTRS

$$faZ \to f(Za)Z$$

with the 'identity' STTRS

$$gx \to x$$

enables the conversion

$$fag \leftrightarrow f(ga)g$$

Both STTRSs are orthogonal and terminating (note that no redex-creation is possible in either of them). Since the displayed conversion witnesses both cyclicity and non-normalisation of the term fag in their disjoint union, this provides us in one fell swoop with counterexamples to modularity of all three properties—left-linear completeness, normalisation, and orthogonal acyclicity—at the same time.

The counterexample essentially relies on the so-called substitution calculus (in the terminology of van Oostrom and van Raamsdonk) of STTRSs being limp; whereas for PRSs a term substituted for the meta-variable Z may immediately act upon its argument a in the rhs of the first rule (e.g. substituting $x.x$ for Z in first rule of the corresponding PRS would yield a step from $fa(x.x)$ to itself), there are no substitution-calculus steps in STTRSs.

5 Discussion

The properties considered above are the standard meta-theoretical properties of rewrite systems.[1] Their failure to be modular in three of the more common higher-order rewriting formats indicates that the situation in first-order term rewriting, where each *is* modular, is the exception rather than the rule. Technically, the main proof technique for establishing modularity results for first-order TRSs is based on terms in the disjoint union of TRSs being *stratified* in the sense

[1](Orthogonal) acyclicity is still a bit out of the ordinary, but should be important in settings where rewrite systems are used to model non-terminating phenomena.

that each term in the disjoint union has a unique decomposition into layers of components residing in either of the TRSs separately, and moreover that this stratification is *preserved* by rewriting in the sense that the *rank*, i.e. the number of layers, cannot increase along a reduction. In the higher-order case, preservation fails due to the presence of meta-variables which allow for *nesting* in the rhs of rules, causing that the rank may *increase* along a reduction. For that reason and from that perspective, it is seems less useful to try to generalise the standard first-order TRS proof technique of proving a property modular by induction on the *rank* to more general (categorical) settings. After all, the notion of rank is very much restricted to (first-order) terms and will certainly be inappropriate in settings—such as graph rewriting—where the objects do not have a layer structure.[2]

From the above it seems that modularity of properties for first-order TRSs can be considered a kind of syntactic accident. In more general settings like higher-order rewriting, additional conditions guaranteeing the terms in the disjoint union to make semantic sense should be considered. That is, more stringent notion of modules/modularisation should be developed.

Hoping to have contributed a bit to the transfer of *speclore* to folklore, we conclude with a positive exception to the non-modularity rule.

Left-linear confluence For first-order left-linear TRSs, modularity of confluence is a trivial consequence of modularity of confluence for arbitrary TRSs. However, since the latter fails in the higher-order case, one may wonder whether left-linearity would suffice to regain modularity of confluence. Indeed it does.

Theorem 10. *Confluence is modular for left-linear PRSs.*

The idea of the proof, as presented in the PhD thesis of van Oostrom,[3] is to use the Hindley–Rosen Lemma and confluence of each of the PRSs, to reduce confluence of the union to their commutation. The latter holds, because since the signatures are disjoint, and since the rules of the respective PRSs were assumed to be left-linear they are therefore orthogonal *to each other*.

The reason why left-linear confluence *is* modular where the properties considered above all fail to be so, is that the proof relies essentially on mutual orthogonality of the rules of the individual PRSs, and mutual orthogonality is something which is built in so-to-speak when taking *disjoint* unions. Thus, (left-linear) confluence is also modular for other 'structured' rewriting formats such as e.g. graph rewriting.

[2] Of course, one could try to *represent* graphs by means of their spanning tree (term!) and rewrite these, i.e. one could resort to graph term rewriting, but that is not the issue here.

[3] In fact, the more general result is shown there that the (ordinary, non-disjoint) union of two left-linear confluent PRSs is confluent, if the rules are *weakly* orthogonal w.r.t. each other, i.e. all critical pairs are trivial.