Orthogonality and sequentiality in substructural Linear HRSs

Connor Lane Smith

cls@clsmith.net

Abstract

Linear HRSs are higher-order rewriting systems having the substructural linear λ-calculus as a substitution calculus. We explore the properties of orthogonality and sequentiality, focusing on multiplicative and additive conjunction, and how they differ in behaviour to 'intuitionistic' HRSs.

Keywords and phrases lambda calculus, term rewriting, higher-order rewriting, linear logic

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Linear HRSs are higher-order rewriting systems having the substructural linear λ-calculus [1] as a substitution calculus [8, 9] over which rewrites take place. Here we explore certain interesting properties, especially orthogonality and sequentiality, where they differ in behaviour to 'intuitionistic' HRSs [5]. We begin with only linear implication '$\multimap$', and later introduce multiplicative conjunction '$\otimes$' and additive conjunction '&'. Linear HRSs permitting exponentiation '!' have been explored in the context of optimal sharing graphs [6].

▶ **Definition 1.** Given a signature Σ of symbols, a *linear* λ*-term* $t : \tau$ is derived as follows:

$$\frac{}{x : \sigma \vdash x : \sigma} \qquad \frac{\Gamma \uplus \{x : \sigma\} \vdash t : \tau}{\Gamma \vdash \lambda x^{\sigma}.t : \sigma \multimap \tau} \qquad \frac{\Gamma \vdash t_1 : \sigma \multimap \tau \quad \Delta \vdash t_2 : \sigma}{\Gamma \uplus \Delta \vdash t_1 t_2 : \tau} \qquad \frac{F : \tau \in \Sigma}{\vdash F : \tau}$$

▶ **Definition 2.** A *pattern* is a β-normal term in which a free variable occurrence ξ may only be applied to (terms η-equivalent to) pairwise-distinct bound variables. We call these free variables 'metavariables'. Moreover, a pattern must have a symbol, e.g. F, at its head.

▶ **Definition 3.** A Linear HRS comprises a set of rules $\Gamma \vdash l \to r : \alpha$, i.e. $\Gamma \vdash l : \alpha$ and $\Gamma \vdash r : \alpha$, where l is a pattern and α a base type, inducing a rewrite relation $C[l^{\theta}] \to C[r^{\theta}]$.

2 Locality

▶ **Definition 4.** A redex is *external* to a term if it is 'persistently outermost'. That is, its residuals remain outermost no matter which other redexes in the term are contracted.

▶ **Definition 5.** A rewriting system is *local* if any term which is not in normal form contains at least one external redex.

A first-order TRS is local if it is orthogonal [7]. However, the same is not true of higher-order rewriting. In an orthogonal HRS, the weakening of a variable may activate a new outermost redex, so a reducible term need not necessarily have an external redex.

▶ **Example 6.** Consider the following HRS:

$$G\zeta \to A \qquad F(\lambda x.\xi) \to B$$

Given a term $F(\lambda x.G(Gx))$, whose outermost redex is $G(Gx)$, one may reduce the inner redex Gx to yield $F(\lambda x.GA)$, with a new outermost redex at its root having been activated by the weakening of x. The initial term therefore had no external redex.

2.1 Overlappingness

The constraints placed on variables in the linear λ-calculus has a significant impact on the overlappingness of patterns. In an HRS, applying a metavariable to a variable means that the variable may occur in the metavariable's substitute, but it equally may not. This means that any two metavariable occurrences will always be unifiable, for example by mapping both metavariables to a single closed term, and so two patterns that differ only in their metavariable occurrences will always overlap. In contrast, in a Linear HRS, a variable to which a metavariable is applied *must* occur in the substitute, and indeed nowhere else. This means two patterns may only overlap if intersecting metavariable occurrences are applied to the same sets of variables.

▶ **Example 7.** The following rules overlap in an HRS, but not in a Linear HRS.

$$F(\lambda x.G(\xi x)\zeta) \to C(\xi A)\zeta \qquad F(\lambda x.G\zeta(\xi x)) \to C\zeta(\xi A)$$

In an HRS, ξ and ζ may be mapped to a single closed term t, i.e. $\{\xi \mapsto \lambda x.t, \zeta \mapsto t\}$, unifying the terms. In a Linear HRS the terms ξx and ζ are not unifiable as the substitute cannot weaken x, which must occur in the substitute of the former yet cannot in that of the latter.

In HRSs, the overlappingness of two patterns can be determined even after replacing all metavariable occurrences with holes. This is not true in Linear HRSs, which may have patterns that are non-overlapping only by the arguments to their metavariable occurrences.

2.2 Extendedness

An HRS is described as 'fully-extended' if each metavariable occurrence is applied to all bound variables in scope. Full extendedness is an important property in HRSs because a fully-extended orthogonal HRS is local [7].

Orthogonal Linear HRSs are local whether or not they are fully extended: as no variable is permitted to weaken, there is no opportunity for non-local behaviour through the activation of a redex by the weakening of a variable. The reducibility of a term may still be determined by the occurrence of a variable inside one subterm or another, but this cannot change during reduction. Variable occurrences are invariant to a term's equivalence class; a variable may only disappear due to a substitution originating from a rewrite local to that variable's binder.

▶ **Example 8.** The orthogonal Linear HRS in Example 7 is local even though it is not fully-extended.

2.3 'Almost overlappingness'

If the Linear HRS in Example 7 were an intuitionistic HRS then its rules would not cover all possibilities of variable occurrence, as x might occur in both subterms, but in a Linear HRS the two rules are exhaustive.

We cannot however trivially unify these two rules into a single rule, as $F(\lambda x.G(\xi x)(\zeta x))$ is not a legal linear λ-term, so it would seem we are essentially forced to distinguish the two cases despite their common reduct. That two rules may be non-overlapping yet have common reducts for all redexes, by construction, is only possible in an HRS if the right-hand side is closed, as otherwise they must be at best *almost* non-overlapping.

▶ **Definition 9.** A rewriting system is *almost non-overlapping* if any critical pair is a trivial overlay.

We may call these rules 'almost overlapping', as they would be almost *non*-overlapping were it not for the ability of linear variables to suppress the overlap between ξx and ζ.

Almost non-overlapping rules may be resolved simply by removing one of the overlapping rules. 'Almost overlapping' rules are not quite so easily resolved, though they can be if we introduce a type operator for multiplicative conjunction into the substitution calculus. These effectively allow us to specify that a variable may occur in either of two subterms, but that we need not discriminate between the two.

▶ **Definition 10.** Multiplicative conjunction is added to the linear λ-calculus like so:

$$\frac{\Gamma \vdash t_1 : \tau_1 \quad \Delta \vdash t_2 : \tau_2}{\Gamma \uplus \Delta \vdash t_1 \otimes t_2 : \tau_1 \otimes \tau_2} \qquad \frac{\Gamma \vdash s : \sigma_1 \otimes \sigma_2 \quad \Delta \uplus \{x : \sigma_1, y : \sigma_2\} \vdash t : \tau}{\Gamma \uplus \Delta \vdash \text{let } s \text{ be } x \otimes y \text{ in } t : \tau}$$

▶ **Example 11.** The Linear HRS in Example 7 may be joined into a single rule by making use of the 'let-$\otimes$-expression':

$$F(\lambda x.\,\text{let } \xi x \text{ be } y \otimes z \text{ in } Gyz) \to \text{let } \xi A \text{ be } y \otimes z \text{ in } Cyz$$

Consequently, we are only able to substitute a single term for x no matter which subterm it occurs in — hence the single right-hand metavariable occurrence ξA.

3 Non-left-linearity

Linear HRSs with only multiplicative conjunction are trivially left-linear: each variable must occur exactly once on the left-hand side. The same is true of the right-hand side, so metavariable erasure is also forbidden. Neither is true once we introduce additive conjunction.

▶ **Definition 12.** Additive conjunction is added to the linear λ-calculus like so:

$$\frac{\Gamma \vdash t_1 : \tau_1 \quad \Gamma \vdash t_2 : \tau_2}{\Gamma \vdash \langle t_1, t_2 \rangle : \tau_1 \,\&\, \tau_2} \qquad \frac{\Gamma \vdash t : \tau_1 \,\&\, \tau_2}{\Gamma \vdash \text{fst } t : \tau_1} \qquad \frac{\Gamma \vdash t : \tau_1 \,\&\, \tau_2}{\Gamma \vdash \text{snd } t : \tau_2}$$

▶ **Example 13.** An 'if-then-else' expression can now be written, where $\text{If} : \mathbb{B} \multimap (\tau \,\&\, \tau) \multimap \tau$,

$$\text{If True } \xi \to \text{fst } \xi \qquad \text{If False } \xi \to \text{snd } \xi$$

Additive conjunction permits a variable to occur more than once *syntactically*, even if it is linear 'semantically'. This has the perhaps slightly unusual consequence that a Linear HRS need not necessarily be left- or right-linear in the traditional syntactic sense.

▶ **Example 14.** Consider Huet's [3] classic non-overlapping, yet non-left-linear, TRS:

$$F(x, x) \to A \qquad F(x, G(x)) \to B \qquad C \to G(C)$$

A very similar Linear HRS is possible, which like the original TRS is non-confluent:

$$F\langle x, x\rangle \to Ax \qquad F\langle x, Gx\rangle \to Bx \qquad C \to GC$$

We are forced here to include the free variable x on the right-hand side of the first two rules, which means that unlike the original it is not weakly normalising — but it is non-confluent:

$$F\langle C, C\rangle \to AC \to A(GC) \to \cdots$$
$$F\langle C, C\rangle \to F\langle C, GC\rangle \to BC \to B(GC) \to \cdots$$

3.1 Erasure

We could restore weak normalisation to the previous example by including the additive conjunctive identity, which allows us to 'syntactically' erase a variable, although the reduct is still considered to 'use' the variable for the purpose of linearity.

$$\frac{}{\Gamma \vdash \langle\rangle : \top}$$

▶ **Example 15.** Returning to Example 14, where A and B are both of type $\top \multimap 0$:

$$F\langle x, x\rangle \to A\langle\rangle \qquad F\langle x, Gx\rangle \to B\langle\rangle \qquad C \to GC$$

However, unless handled carefully, $\top$ is at risk of putting the most interesting properties of substructural term rewriting in jeopardy.

▶ **Example 16.** Consider again the Linear HRS in Example 7, which is non-overlapping solely due to the linearity of bound variables. If we were to find ourselves with a term $F(\lambda x.G(D\langle\rangle)(D\langle\rangle))$, which of the two rules would this match?

There are two possible constructions of this term. The first derivation constructs the first $\langle\rangle$ with x in its type context, as it would if x had begun in that subterm and were erased during reduction, and the second derivation likewise constructs the second $\langle\rangle$ with x in its type context. In either case the variable *is* properly accounted for in the construction $x : 0 \vdash \langle\rangle : \top$, but if we permit either occurrence to substitute x, no matter whether it was in its type context or not, then the two rules do now overlap in this case.

The key is whether or not we consider the expression $\Gamma \vdash \langle\rangle : \top$ to be constrained after construction to substitutions over the type context Γ. If we do not then the resulting expression may be used to 'swallow' a substitute for any variable, even one that does not occur in Γ, and so we lose the ability to define rules that overlap except for their metavariables' extendedness. If we do apply this constraint then it must be maintained during reduction, so that we can only construct a new $\langle\rangle$-expression from an old if substitution allows, e.g.:

$$\frac{\Gamma \uplus \{x : \sigma\} \vdash \langle\rangle : \top \quad \Delta \vdash t : \sigma}{\Gamma \uplus \Delta \vdash \langle\rangle[t/x] \equiv \langle\rangle : \top}$$

It may therefore make sense to label a term $\langle\rangle$ with its type context Γ — i.e. $\langle\rangle_\Gamma$ — so that its substitutions may be correctly restricted to those variables in Γ.

3.2 Variable containment

Although additive conjunction proves useful for rules such as 'If' in Example 13, and is manageable even when used non-left-linearly, if used to the their full potential they can serve to undermine the 'natural' variable containment provided by the linear λ-calculus and require that we place restrictions on metavariable usage so that we may have a meaningful rewrite relation.

The most obvious problem is a rule like $F\langle\rangle \to \xi$, in which the metavariable ξ does not occur on the left-hand side, as it is 'used' in the construction of $\langle\rangle$. Another, more subtle problem, is a rule like $F(\mathrm{fst}\ \xi) \to \mathrm{snd}\ \xi$, where the metavariable ξ does occur on both sides, yet the value of its second half as used on the right-hand side is undefined by the matching of the left-hand side. In either case, a metavariable in the reduct may be instantiated to any term having the exact same free variables as the term in the redex — a form of term narrowing peculiar to the linear type system.

It would be heavy-handed, though, to simply forbid these expressions from the left-hand side altogether. Although they can be used, if unconstrained, to break the rewrite relation, they can also be used to construct certain terms on the right-hand side that would not otherwise be possible.

▶ **Example 17.** Consider the following Linear HRS, where $F : (0 \,\&\, 0) \multimap 0$:

$$F\langle C(\text{fst } \xi), C(\text{snd } \xi)\rangle \to F\xi$$

It would not be possible to use two metavariables ξ_1 and ξ_2 to replace fst ξ and snd ξ, as the two subterms in $\langle t_1, t_2\rangle$ must have the same set of free variables, which would not hold.

4 Sequentiality

Sequentiality [4] may be defined for HRSs in much the same way as it is for TRSs, except for the added complication of variables, which if the system is non-fully-extended may undermine the property in the same way as with non-locality. Yet, since in the linear λ-calculus variable occurrence is invariant in reduction, the presence of a bound variable in a term may be determined for good when its root is constructed. If variable occurrence may be checked 'out of band', as if each symbol were labelled when constructed with the variables occurring in each of its arguments, then bound variables cease to pose a threat to sequentiality: we are simply matching both on symbols and on their static labels of variable occurrences.

A system like that in Example 7 would then distinguish between $G_{x:1}$ and $G_{x:2}$, indicating whether the bound variable x occurs in its first or second subterm, so that $F(\lambda x.G_{x:i}\Box_1\Box_2)$ has $\Box_i$ — either $\Box_1$ or $\Box_2$ — as a sequential index. Using a mechanism like this, or indeed one that actually performs the required occurrence check to determine whether G ought to be labelled as $G_{x:1}$ or $G_{x:2}$ as it goes along, sequentiality may then proceed as normal. Of course, other rules may require similar labellings, so we must support checking not only $G_{x:1}$ but also G_ϕ for any mapping $\phi : \Gamma \to \mathbb{N}$ where $\Gamma \vdash Gt_1t_2 : \tau$.

If a similar system were attempted for non-fully-extended intuitionistic HRSs then the labelling would undermine the locality of the rewrite step, as the weakening of a variable may cause a relabelling of symbols all the way up to the root of the term. But with Linear HRSs a relabelling may only occur downwards, triggered by a substitution local to a variable's binder, and so sequential reduction remains a meaningful reduction strategy.

Similar to overlappingness, non-full-extendedness allows for two terms to be sequential only because of their occurring free variables, even if they are non-overlapping.

▶ **Example 18.** Consider 'Gustave's TRS' [2], which is non-overlapping yet non-sequential:

$$F(A, B, x) \to r_1 \qquad F(x, A, B) \to r_2 \qquad F(B, x, A) \to r_3$$

A Linear HRS counterpart to this may have variable occurrences that alter its sequentiality:

$$F(\lambda x.\lambda y.\, x \otimes y \otimes \xi) \to r_1 \qquad F(\lambda x.\lambda y.\, \xi \otimes x \otimes y) \to r_2 \qquad F(\lambda x.\lambda y.\, y \otimes \xi \otimes x) \to r_3$$

Although this has the same construction in terms of its atoms and metavariables, and so the terms are pattern matched in the same way, the fact that those atoms are variables which may be subject to a variable occurrence check means that we can determine their location without having to inspect the subterm in such a way that it must be a sequential index.

References

1 Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

2 Gérard Berry. Bottom-up computation of recursive programs. *Revue Française d'Automatique, Informatique et Recherche Opérationnelle*, 10:47–82, 1976.

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

4 Gérard Huet and Jean-Jacques Lévy. Computations in orthogonal rewriting systems, I and II. *Computation Logic – Essays in Honor of A. Robinson*, pages 394–443, 1991.

5 Tobias Nipkow. Higher-order critical pairs. In *Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science*, 1991.

6 Connor Lane Smith. *Optimal Sharing Graphs for Substructural Higher-Order Rewriting Systems*. PhD thesis, University of Kent, 2017.

7 'Terese'. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

8 Vincent van Oostrom. *Confluence for abstract and higher-order rewriting*. PhD thesis, Vrije Universiteit Amsterdam, 1994.

9 Femke van Raamsdonk. *Confluence and normalisation for higher-order rewriting*. PhD thesis, Vrije Universiteit Amsterdam, 1996.