Higher-Order Recursive Path Orderings à la carte*

Jean-Pierre Jouannaud
LRI, Bat. 490
CNRS/Univ. de Paris Sud
91405 Orsay, FRANCE
Jean-Pierre.Jouannaud@lri.fr
and
LIX, École Polytechnique
91400 Palaiseau, FRANCE

Albert Rubio
Edif. C6-219,
Univ. Politècnica de Catalunya
Pau Gargallo 5
08028 Barcelona, SPAIN
rubio@lsi.upc.es

Rewrite rules are increasingly used in programming languages and logical systems, with two main goals: defining functions by pattern matching; describing rule-based decision procedures. The use of rules in logical systems is subjected to three main meta-theoretic properties : subject reduction, local confluence, and strong normalization. The first two are usually easy. The last one is difficult, requiring the use of sophisticated proof techniques based, for example, on Tait and Girard's reducibility predicate technique. Our ambition is to remedy this situation by developping for the higher-order case the kind of semi-automated termination proof techniques that are available for the first-order case, of which the most popular one is Dershowitz's recursive path ordering.

Our contribution to this program is a reduction ordering for typed higher-order terms following a typing discipline including ML-like polymorphism and sort constructors, which conservatively extends β-reductions for higher-order terms on the one hand, and on the other hand Dershowitz's recursive path ordering for first-order unisorted terms. In the latter, the precedence rule allows to decrease from the term $s = f(s_1, \ldots, s_n)$ to the term $g(t_1, \ldots, t_n)$, provided that (i) f is bigger than g in the given precedence on function symbols, and (ii) s is bigger than every t_i. For typing reasons, in our ordering the latter condition becomes: (ii) for every t_i, either s is bigger than t_i or some s_j is bigger than or equal to t_i. Indeed, we can instead allow t_i to be obtained from the subterms of s by computability preserving operations. Here, computability refers to Tait and Girard's strong normalization proof technique which we have used to show that our ordering is well-founded.

In a preliminary version of this work presented at the Federated Logic Conference in Trento, our ordering could only compare terms of equal types (after

*This work was partly supported by the RNRT project CALIFE and the CICYT project HEMOSS ref. TIC98-0949-C02-01.

identifying sorts such as Nat or List). In the present version, our ordering is capable of ordering terms of *decreasing types*, the ordering on types being simply a slightly weakened form of Dershowitz's recursive path ordering. This yields a very elegant presentation of the whole machinery by integrating both orderings into a single one operating on terms and types as well. This presentation should in turn be considered as the key missing stone on the way towards the definition of a recursive path ordering for dependent type calculi.

Several other improvements have been made to the preliminar version, which allow to prove a great variety of practical examples. To hint at the strength of our ordering, let us mention that the polymorphic version of Gödel's recursor for the natural numbers is easily oriented. And indeed, our ordering is polymorphic in the sense that a single comparison allows to prove the termination property of all monomorphic instances of a polymorphic rewrite rule. Many non-trivial examples can be carried out which examplify the expressive power of these orderings.

In the litterature, one can find several attempts at designing methods for proving strong normalization of higher-order rewrite rules based on ordering comparisons. These orderings are either quite weak, or need an important user interaction. Besides, they operate on terms in η-long β-normal form, hence apply only to the higher-order rewriting "à la Nipkow", based on higher-order pattern matching modulo $\beta\eta$. To our knowledge, our ordering is the first to operate on arbitrary higher-order terms, therefore applying to the other kind of rewriting, based on plain pattern matching and having β-reduction as an additional rule. On the other hand, a very simple modification of the ordering can be used to prove strong normalization of higher-order rewrite rules operating on terms in η-long β-normal forms, and indeed we show that this is true for any well-founded higher-order rewrite ordering containing β-reductions.