A monotonic Higher-Order Semantic Path Ordering

Cristina Borralleras[1] and Albert Rubio[2]

[1] Universitat de Vic, Spain
Email: cristina.borralleras@uvic.es
[2] Universitat Politècnica de Catalunya, Barcelona, SPAIN
Email: rubio@lsi.upc.es

There is an increasing use of higher-order rewrite rules in many programming languages and logical systems. As in the first-order case, termination is a fundamental property of most applications of higher-order rewriting. Thus, there exists a need to develop for the higher-order case the kind of semi-automated termination proof techniques that are available for the first-order case.

There have been several attempts at designing methods for proving strong normalization of higher-order rewrite rules based on ordering comparisons. Recently, in [JR99], Dershowitz's recursive path ordering has been extended to a higher-order setting by defining a higher-order recursive path ordering (HORPO) on terms of a typed lambda-calculus generated by a signature of polymorphic higher-order function symbols. This ordering is powerful enough to deal with many non-trivial examples and can be automated. HORPO is the first method which operates on arbitrary higher-order terms, therefore applying to higher-order rewriting based on plain pattern matching, where β-reduction is considered as any other rewrite rule. Furthermore, HORPO can operate as well on terms in η-long β-normal form, and hence it provides as well a method for provig strong normalization of higher-order rewriting "à la Nipkow", based on higher-order pattern matching modulo $\beta\eta$.

However, HORPO inherits the same weaknesses that RPO has in the first-order case. RPO is a *simplification ordering* (a monotonic ordering including the subterm relation), which extends a precedence on function symbols to an ordering on terms. It is simple and easy to use, but unfortunately, it turns out, in many cases, to be a weak termination proving tool. First, there are many term rewrite systems (TRSs) that are terminating but are not contained in any simplification ordering, i.e. they are not *simply terminating*. Second, in many cases the head symbol does not provide enough information to prove the termination of the TRS. Therefore, since HORPO follows the same structure and the same use of a precedence as in RPO (in fact, it reduces to RPO when restricted to first-order case), it is easy to expect that similar weaknesses will appear when proving termination of higher-order rewriting.

To avoid this weakness, in the first-order case, many different so-called *transformation methods* have been developed. By transforming the TRS into a set of ordering constraints, the Arts and Giesl's *dependency pair* method has become a successful general technique for proving termination of (non-simply terminating) TRSs.

As an alternative to transformation methods, more powerful term orderings, like Kamin and Levy's *semantic path ordering* (SPO), can be used. SPO generalizes RPO by replacing the precedence on function symbols by any (well-founded) underlying (quasi-)ordering involving the whole term and not only its head symbol. Although the simplicity of the presentation is kept, this makes the ordering much more powerful. Unfortunately, SPO is not so useful in practice, since, although it is well-founded, it is not, in general, monotonic. Hence, in order to ensure termination, apart from checking that the rules of the rewrite system are included in the ordering, in addition the monotonicity for contexts of the rewrite rules has to be proved.

In a recent work [BFR00], a monotonic version of SPO, called MSPO, has been presented. MSPO overcomes the weaknesses of RPO, is automatable and it is shown to generalize other existing transformation methods.

Due to the fact that RPO and SPO share the same "path ordering nature", our aim is to obtain for SPO and MSPO the same kind of extensions to the higher-order case as it was done for RPO.

In this work we introduce the *higher-order semantic path order* (HOSPO), which generalizes HORPO by replacing the precedence on algebraic function symbols by any underlying (quasi-)ordering on higher-order terms. Then a monotonic version of HOSPO, called MHOSPO, is obtained, which provides, a powerful method for proving termination of higher-order rewriting. The ingredients for MHOSPO are still (quasi-)ordering on higher-order terms, but to ensure that MHOSPO is adequate to prove termination (i.e. it is a higher-order reduction ordering) several properties on these quasi-orderings have to be required. In order to make MHOSPO useful in practice, and even make it automatable, we have analyzed possible cadidates for these ingredients. To illustrate the power of the resulting method several non-trivial examples, which cannot be proved by any HORPO, are shown to be terminating.

References

[BFR00] C. Borralleras, M. Ferreira, and A. Rubio. Complete monotonic semantic path orderings. In David McAllester, ed., *Proc. of the 17th Int. Conf. on Automated Deduction (CADE-17)*, LNAI 1831, pp. 346–364, Springer-Verlag, 2000.

[JR99] J.-P. Jouannaud and A. Rubio. The higher-order recursive path ordering. In *14th IEEE Symposium on Logic in Computer Science (LICS)*, pp. 402–411, 1999.