# A fully syntactic AC-RPO*

Albert Rubio

Universitat Politècnica de Catalunya, Barcelona, Spain.
E-mail: rubio@lsi.upc.es

Rewrite-based methods with built-in associativity and commutativity (AC) properties for some of the operators are well-known to be crucial in theorem proving and programming. Therefore a lot of work has been done on the development of suitable *AC-compatible* reduction or simplification orderings. An essential additional property of the ordering that is needed in order to preserve the completeness of most rewrite-based theorem proving techniques (modulo AC) is *AC-totality*, i.e. the totality on (AC-different) ground terms.

Since the initial attempts, it has always been an aim to obtain AC-compatible versions of Dershowitz' *recursive path ordering* [Der82], as it is simple, easy to automatize and use, and normally orients the rules in an adequate direction. In [RN95] we gave the first RPO-based AC-total and AC-compatible reduction ordering without any restriction on the number of AC-symbols or on the precedence over the signature. Unfortunately, although being defined in terms of RPO, it does not behave like RPO; e.g. it does not orient the distributivity rule in the "right" (i.e. distributing) way, since a transformation on the terms is applied before using RPO (this approach, with different transformations, is also used in [BP85] among others). Therefore, a better approach seems to be to directly apply an RPO-like scheme, treating as the only special case the *AC-equal-top case*, that is, when both terms to be compared are headed by the same AC-symbol. In this direction the first AC-compatible simplification ordering with an RPO scheme was defined in [KSZ90] and the first one AC-total on ground terms in [KS97]. Other simpler proposals for AC-orderings with RPO scheme were given in [Rub97] and in [KS98].

However, all these AC-orderings need to interpret terms (apart from *flattening*) in some way, which makes their behaviour less intuitive, unlike it happens with the standard RPO, whose simple fully syntactic definition has been an important reason for its success.

We propose the first fully syntactic AC-RPO, i.e., no interpretation is needed apart from flattening. It is very simple, and hence easy to implement, and its behaviour is intuitive as for the standard RPO. The ordering is AC-total, and defined uniformly for both ground and non-ground terms, as well as for partial precedences.

Moreover, precisely due to the fact that it is not interpretation-based, it is the first AC-RPO that can deal *incrementally* with partial precedences, i.e. if $s \succ t$, then $s \succ t$ under any extension of the precedence. This aspect is essential, together with its intuitive behaviour, for interactive applications like Knuth-Bendix completion. Of course, previously existing orderings could work with

* Partially supported by the ESPRIT working group CCL-II, ref. WG # 22457.

partial precedences, but in a useless way, simply by considering an arbitrarily chosen total extension of the partial precedence, and hence loosing incrementality.

In order to introduce the concepts smoothly we present the ordering in three steps, first for ground terms and total precedences, then for terms with variables and total precedences and finally for terms with variables and partial precedences, each definition strictly extending the previous one. For this reason we prove all properties only for the last one, showing that it is indeed an AC-compatible simplification ordering.

All this results are reported in [Rub98].

## References


[BP85] L. Bachmair and D. A. Plaisted. Termination orderings for associative-commutative rewriting systems. *Journal of Symbolic Computation*, 1:329–349, 1985.

[Der82] Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

[KS97] D. Kapur and G. Sivakumar. A total, ground path ordering for proving termination of ac-rewrite systems. In H. Comon, editor, *8th International Conference on Rewriting Techniques and Applications*, LNCS, 1997.

[KS98] D. Kapur and G. Sivakumar. A recursive path ordering for proving associative-commutative termination, 1998.

[KSZ90] D. Kapur, G. Sivakumar, and H. Zhang. A new method for proving termination of ac-rewrite systems. In *Conf. Found. of Software Technology and Theoretical Computer Science*, LNCS 472, pages 134–148, New Delhi, India, December 1990. Springer-Verlag.

[RN95] Albert Rubio and Robert Nieuwenhuis. A total AC-compatible ordering based on RPO. *Theoretical Computer Science*, 142(2):209–227, May 15, 1995.

[Rub97] Albert Rubio. A total AC-compatible ordering with RPO scheme. Technical Report UPC-LSI-97, Univ. Polit. Catalunya, September 1997.

[Rub98] Albert Rubio. A fully syntactic AC-RPO. Technical Report UPC-LSI-97, Univ. Polit. Catalunya, December 1998.


This article was processed using the LaTeX macro package with LLNCS style