The termination hierarchy for term rewriting

Hans Zantema
Utrecht University, The Netherlands
e-mail: `hansz@cs.uu.nl`

(Tutorial talk)

A natural way to prove termination of a term rewriting system is to find an interpretation in an algebra equipped with a well-founded order, in such a way that for every rule the interpretation of the left hand side is greater than that of the right hand side. If the operations are strictly monotone in all arguments then indeed termination of the TRS can be concluded; the algebra is then called *well-founded monotone*. The requirement that every left hand side exceeds the corresponding right hand side is called *compatibility*. In [1] a hierarchy of termination was proposed based on the carrier set and the operations in this algebra:

- a TRS is terminating if it is compatible with a well-founded monotone algebra;
- a TRS is simply terminating if it is compatible with a well-founded monotone algebra satisfying $f_A(\ldots, a, \ldots) \geq a$ for all operations f_A and all positions in these operations;
- a TRS is totally terminating if it is compatible with a well-founded monotone algebra in which the order is total;
- a TRS is ω-terminating if the interpretation is in the natural numbers;
- a TRS is polynomially terminating if the interpretation is in the natural numbers and the operations are polynomials.

We restrict to finite signatures. Where alternative definitions of these notions appeared before, they are equivalent to ours except for total termination of string rewriting. The following implications hold:

polynomial termination
$\Longrightarrow$ ω-termination
$\Longrightarrow$ total termination
$\Longrightarrow$ simple termination
$\Longrightarrow$ termination.

It is possible to extend this hierarchy by some extra levels and some ramifications. For instance, termination by recursive path order always implies

ω-termination, and weak normalization and non-loopingness can be seen as levels beyond termination.

None of the implications in the hierarchy holds in the reverse direction: for every implication $A \Rightarrow B$ even a single string rewrite rule can be given satisfying B but not A. Moreover, deciding these properties is hard: in [2] it has been shown that for most implications $A \Rightarrow B$ the question whether A holds for a TRS satisfying B is undecidable. This even holds for single rules ([3]).

More recent we considered some modifications on the notion of well-founded monotone algebra. One original requirement is that all operations f_A are strictly monotone in all arguments. This requirement may be weakened to the combination of weak monotonicity and the condition that $f_A(\ldots, a, \ldots) \geq a$ for all operations f_A and all positions in these operations. It can be shown that compatibility with this kind of an algebra still implies termination, while much more operations are allowed to be chosen. For instance, the operation choosing the maximum of two natural numbers is not strictly monotone in both arguments, but satisfies the new condition. It is a natural question whether this modification on the requirements on the operations affects the various notions in the termination hierarchy. Indeed it does. For instance, the single rewrite rule $f(g((x)) \to g(f(f(x)))$ is not ω-terminating in the old sense, but it allows an interpretation in the natural numbers with the weakened requirements.

Another subtility is the difference between the requirement $f_A(\ldots, a, \ldots) \geq a$ and $f_A(\ldots, a, \ldots) > a$. Without referring to the algebras it can be stated as the question whether adding the rules $f(\ldots, x, \ldots) \to x$ affects the termination behaviour or not. For termination, simple termination and total termination it is easily shown that adding these rules have no effect. Surprisingly however, ω-termination and polynomial termination may be destroyed by adding these rules. For instance, the TRS consisting of the two rules $f(f(x)) \to g(f(x))$ and $g(g(x)) \to f(g(h(x)))$ is ω-terminating, but after adding the single rule $h(x) \to x$ it is not ω-terminating.

References

[1] Zantema, H. Termination of term rewriting: interpretation and type elimination. *Journal of Symbolic Computation 17* (1994), 23–50.

[2] Geser, A., Middeldorp, A., Ohlebusch, E., and Zantema, H. Relative undecidability in term rewriting. In *Proceedings of the Conference of the European Association of Computer Science Logic (CSL96)* (1997), D. van Dalen, Ed., Lecture Notes in Computer Science, Springer Verlag.

[3] Geser, A., Middeldorp, A., Ohlebusch, E., and Zantema, H. Relative undecidability in the termination hierarchy of single rewrite rules. In *Proceedings Theory and Practice of Software Development (TAPSOFT97, CAAP/FASE)* (1997), M. Bidoit and M. Dauchet, Eds., vol. 1214 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 237–248.