An Algebraic Machine for Imperative Programs

Jan Heering
(Joint work with Jan Bergstra, T.B. Dinesh, and John Field)

CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands,
e-mail: jan@cwi.nl

Abstract

In [3] Field introduced PIM, an algebraic machine for the analysis, transformation, and optimization of imperative programs. For these purposes the algebraic framework is particularly attractive since it provides a uniform setting for modelling the standard dynamic semantics of a language (implemented as a confluent term rewriting system) as well as arbitrary semantics preserving program transformation and partial evaluation (using equations of the complete logic).

Whereas the standard semantics involves only rewriting of PIM ground terms, program transformation and partial evaluation amount to rewriting of PIM terms containing (meta)variables. The appropriate notion of completeness in this setting is not the well-known completeness of equational logic, but ω-completeness [4, 2] of the specification of PIM. Making an algebraic specification ω-complete usually requires hard work (if it is possible at all), and PIM is no exception.

References

[1] J.A. Bergstra, T.B. Dinesh, J. Field, and J. Heering, The PIM logic for imperative programs, in preparation.

[2] J.A. Bergstra and J. Heering, Which data types have ω-complete initial algebra specifications?, *Theoretical Computer Science*, **124** (1994) 149–168.

[3] J. Field, A simple rewriting semantics for realistic imperative programs and its application to program analysis, in: *Proceedings of the 1992 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, Yale University Technical Report YALEU/DCS/RR-909, 1992, 98–107.

[4] J. Heering, Partial evaluation and ω-completeness of algebraic specifications, *Theoretical Computer Science*, **43** (1986) 149–167.

Proceedings of ASF+SDF95. A workshop on Generating Tools from Algebraic Specifications. May 11 & 12, 1995, CWI, Amsterdam, M.G.J. van den Brand, A. van Deursen, T.B. Dinesh, J.F.Th. Kamperman & E. Visser (eds.) Technical Report P9504, Programming Research Group, University of Amsterdam