On the Logical Foundations of Staged Computation (Invited talk)

Frank Pfenning
Department of Computer Science
Carnegie Mellon University
fp@cs.cmu.edu

ABSTRACT

Dividing a computation into stages and optimizing later phases using information from earlier phases is a familiar technique in algorithm design. In the realm of programming languages, staged computation has found two important realizations: partial evaluation and run-time code generation. A priori, these are fundamentally operational concepts, concerned with how a program executes, but not what it computes.

In this talk we provide a logical foundation for staged computation which is consistent with the operational intuition. We concentrate on run-time code generation which is related to modal logic via an interpretation of constructive proofs as programs. This correspondence yields new insights into issues of language design and leads to a static type system in which staging errors become type errors. We sketch the language PML (for Phased ML), whose design has been directly motivated by our foundational reconstruction, and discuss our ongoing compiler construction effort [1; 2; 3; 4].

REFERENCES

[1] Rowan Davies. A temporal logic approach to binding-time analysis. In E. Clarke, editor, *Proceedings of the Eleventh Annual Symposium on Logic in Computer Science*, pages 184–195, New Brunswick, New Jersey, July 1996. IEEE Computer Society Press.

[2] Rowan Davies and Frank Pfenning. A modal analysis of staged computation. Submitted. Available as Technical Report CMU-CS-99-153, August 1999. A preliminary version appeared in Guy Steele, Jr., editor, *Proceedings of the 23rd Annual Symposium on Principles of Programming Languages*, pages 258–270, St. Petersburg Beach, Florida, January 1996. ACM Press.

[3] Philip Wickline, Peter Lee, and Frank Pfenning. Run-time code generation and modal-ML. In Keith D. Cooper, editor, *Proceedings of the Conference on Programming Language Design and Implementation (PLDI'98)*, pages 224–235, Montreal, Canada, June 1998. ACM Press.

[4] Philip Wickline, Peter Lee, Frank Pfenning, and Rowan Davies. Modal types as staging specifications for run-time code generation. *ACM Computing Surveys*, 30(3es), September 1998.