References

Books on compiling functional languages

S. L. Peyton-Jones, *The implementation of functional programming languages*, Prentice-Hall, 1987.

A. Appel, *Compiling with continuations*, Cambridge University Press, 1992.

Closures and flow analyses

P. J. Landin, *The mechanical evaluation of expressions*, The Computer Journal, 1964.

M. Wand and P. Steckler, *Selective and lightweight closure conversion*, Principles of Programming Languages, 1994.

O. Shivers, *Control-flow analysis of higher-order languages*, PhD thesis, CMU, 1991.

N. Heintze, *Set-based analysis of ML programs*, Lisp and Functional Programming, 1994.

J. M. Ashley, *The effectiveness of flow analysis for inlining*, International Conference on Functional Programming, 1997.

J. Palsberg and P. O'Keefe, *A type system equivalent to flow analysis*, Principles of Programming Languages, 1995.

A. Aiken and E. Wimmers, *Type inclusion constraints and type inference*, Functional Programming and Computer Architecture, 1993.

J. Eifrig, S. Smith, V. Trifonov, *Sound polymorphic type inference for objects*, Object-Oriented Programming, Systems, Languages and Applications, 1995.

A. Dimock, R. Muller, F. Turbak, J. B. Wells, *Strongly typed flow-directed representation transformation*, International Conference on Functional Programming, 1997.

Data representations

D. Tarditi, G. Morrisett, P. Cheng, C. Stone, R. Harper, P. Lee, *TIL: a type-directed optimizing compiler for ML*, Programming Language Design and Implementation, 1996.

A. Ohori, T. Takamizawa, *An unboxed operational semantics for ML polymorphism*, Lisp and Symbolic Computation, 1997.

X. Leroy, *Unboxed objects and polymorphic typing*, Principles of Programming Languages, 1992.

Z. Shao and A. Appel, *A type-based compiler for Standard ML*, Programming Language Design and Implementation, 1995.

S. L. Peyton-Jones, J. Launchbury, *Unboxed values as first-class citizens in a non-strict functional language*, Functional Programming and Computer Architecture, 1991.

J. Goubault, *Generalized boxing, congruences and partial inlining*, Static Analysis Symposium, 1994.

Code generation

S. Muchnick, *Advanced compiler design and implementation*, Morgan Kaufmann, 1997.

I. Piumarta, F. Riccardi, *Optimizing direct-threaded code by selective inlining*, Programming Language Design and Implementation, 1998.

M. Pouzet, *The program compaction revisited: the functional framework*, EURO-PAR, 1995.

S. L. Peyton-Jones, T. Nordin, D. Oliva, `C--`*: a portable assembly language*, Workshop Implementing Functional Languages, LNCS, 1998.

Garbage collection

P. Wilson, *Uniprocessor garbage collection techniques*, 1996, available from `http://www.cs.utexas.edu/users/wilson`.

P. Wilson et al, *Dynamic storage allocation: a survey and critical review*, 1995, available from `http://www.cs.utexas.edu/users/wilson`.