Bibliography

Abadi, Martín, & Cardelli, Luca. 1996. *A Theory of Objects*. Berlin, Heidelberg, and New York: Springer-Verlag.

Abelson, Harold, & Sussman, Gerald Jay. 1985. *The Structure and Interpretation of Computer Programs*. Cambridge, MA: MIT Press.

Abelson, Harold, & Sussman, Gerald Jay. 1996. *Structure and Interpretation of Computer Programs*. Second edition. Cambridge, MA.: McGraw Hill.

Appel, Andrew W., & Jim, Trevor. 1989. Continuation-Passing, Closure-Passing Style. *Pages 293–302 of: Conf. Rec. 16th ACM Symposium on Principles of Programming Languages*.

Arnold, Ken, & Gosling, James. 1998. *The Java Programming Language*. Second edition. The Java Series. Reading, MA: Addison-Wesley.

Backus, J. W., *et al.*. 1957. The Fortran Automatic Coding System. *Pages 188–198 of: Western Joint Computer Conference*.

Barendregt, Henk P. 1981. *The Lambda Calculus: Its Syntax and Semantics*. Amsterdam: North-Holland.

Barendregt, Henk P. 1991. *The Lambda Calculus*. Revised edition. Studies in Logic and the Foundations of Mathematics, no. 103. Amsterdam: North-Holland.

Bergin, T. J., & Gibson, R. G. (eds.). 1996. *History of Programming Languages*. New York, NY: Addison-Wesley.

Birtwistle, G. M., Dahl, O. J., Myhrhaug, B., & Nygaard, K. 1973. *Simula Begin*. Philadelphia: Auerbach.

Braffort, P., & Hirschberg, D. (eds.). 1963. *Computer Programming and Formal Systems*. Amsterdam: North-Holland.

Church, Alonzo. 1941. *The Calculi of Lambda Conversion*. Princeton, NJ: Princeton University Press. Reprinted 1963 by University Microfilms, Ann Arbor, MI.

Clinger, William, *et al.*. 1985. *The Revised Revised Report on Scheme or The Uncommon Lisp*. Technical Memo AIM-848. Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Clinger, William, Rees, Jonathan, *et al.*. 1991. The Revised^4 Report on the Algorithmic Language Scheme. *ACM Lisp Pointers*, **4**(3), 1–55.

Clinger, William D. 1998. Proper Tail Recursion and Space Efficiency. *Pages 174–185 of: Proceedings of the ACM SIGPLAN '98 Conference on Programming Language Design and Implementation*.

Clinger, William D., Hartheimer, Anne H., & Ost, Eric. 1999. Implementation Strategies for First-class Continuations. *Higher Order and Symbolic Computation*, **12**, 7–45.

Clocksin, William F., & Mellish, Christopher S. 1994. *Programming in Prolog*. Fourth edition. Berlin, Heidelberg, and New York: Springer-Verlag.

Danvy, Olivier, & Filinski, Andrzej. 1992. Representing Control: A study of the CPS transformation. *Mathematical Structures in Computer Science*, **2**(4), 361–391.

Dybvig, R. Kent. 1987. *The Scheme Programming Language*. Englewood Cliffs, NJ: Prentice-Hall.

Dybvig, R. Kent. 1996. *The Scheme programming language: ANSI Scheme*. Second edition. Upper Saddle River, NJ 07458, USA: Prentice-Hall PTR.

Ellis, Margaret A., & Stroustrup, Bjarne. 1992. *The Annotated C++ Reference Manual*. Reading: Addison-Wesley.

Federhen, Scott. 1980. *A Mathematical Semantics for PLANNER*. Master's Thesis, University of Maryland.

Felleisen, Matthias, & Friedman, Daniel P. 1996. *The Little MLer*. MIT Press.

Fischer, Michael J. 1972. Lambda-Calculus Schemata. *Pages 104–109 of: Proceedings ACM Conference on Proving Assertions about Programs*. SIGPLAN Notices, 7(1), January 1972.

Fischer, Michael J. 1993. Lambda-Calculus Schemata. *Lisp and Symbolic Computation*, **6**(3/4), 259–288.

Flatt, Matthew, Krishnamurthi, Shriram, & Felleisen, Matthias. 1998 (Jan.). Classes and Mixins. *Pages 171–183 of: Proceedings ACM Symposium on Principles of Programming Languages*.

Friedman, Daniel P. 1974. *The Little LISPer*. Palo Alto, CA: Science Research Associates.

Friedman, Daniel P., & Felleisen, Matthias. 1996. *The Little Schemer*. Fourth edition. MIT Press.

Gamma, Erich, Helm, Richard, Johnson, Ralph, & Vlissides, John. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA, USA: Addison Wesley.

Goldberg, A., & Robson, D. 1983. *Smalltalk-80: The Language and its Implementation*. Reading, MA: Addison-Wesley.

Gosling, James, Joy, Bill, & Steele, Guy L. 1996. *The Java Language Specification*. The Java Series. Reading, MA, USA: Addison-Wesley.

Hankin, Chris. 1994. *Lambda Calculi: A Guide for Computer Scientists*. Graduate Texts in Computer Science, vol. 3. Oxford: Clarendon Press.

Haynes, Christopher T., Friedman, Daniel P., & Wand, Mitchell. 1986. Obtaining Coroutines with Continuations. *J. of Computer Languages*, **11**(3/4), 143–153.

Hewitt, Carl. 1977. Viewing Control Structures as Patterns of Passing Messages. *Artificial Intelligence*, **8**, 323–364.

Hieb, Robert, Dybvig, R. Kent, & Bruggeman, Carl. 1990. Representing Control in the Presence of First-class Continuations. *Pages 66–77 of: Proceedings of the ACM SIGPLAN '90 Conference on Programming Language Design and Implementation*.

Hindley, R. 1969. The Principal Type-Scheme of an Object in Combinatory Logic. *Transactions of the American Mathematical Society*, **146**, 29–60.

Hofstadter, Douglas R. 1979. *Gödel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books.

Horowitz, Ellis. 1987. *Programming Languages: A Grand Tour*. Third edition. Rockville, Maryland: Computer Science Press.

Hudak, Paul, *et al.*. 1990. *Report on the Programming Language HASKELL*. Technical Report YALEU/DCS/RR-777. Yale University, CS Dept.

Hughes, R. J. M. 1982. Super Combinators: A New Implementation Method for Applicative Languages. *Pages 1–10 of: Proc. 1982 ACM Symposium on Lisp and Functional Programming*.

IEEE. 1991. *IEEE Standard for the Scheme Programming Language, IEEE Standard 1178-1990*. IEEE Computer Society, New York.

Kelsey, Richard, Clinger, William, & Rees, Jonathan. 1998. Revised^5 Report on the Algorithmic Language Scheme. *Higher-Order and Symbolic Computation*, **11**(1), 7–104. Also appeared in *SIGPLAN Notices* 33:9, September 1998.

Kiczales, G., des Rivières, J., & Bobrow, D. G. 1991. *The Art of the Meta-Object Protocol*. Cambridge (MA), USA: MIT Press.

Knuth, Donald E., & Pardo, L. T. 1977. The Early Development of Programming Languages. *Pages 419–493 of*: Belzer, J., Holzman, A. G., & Kent, D. (eds.), *Encyclopedia of Computer Science and Technology*, vol. 6. New York: Marcel Dekker.

Kranz, David A., Kelsey, Richard, Rees, Jonathan A., Hudak, Paul, Philbin, James, & Adams, Norman I. 1986. Orbit: An Optimizing Compiler for Scheme. *Pages 219–223 of: Proceedings SIGPLAN '86 Symposium on Compiler Construction*.

Liskov, Barbara, Snyder, Alan, Atkinson, R., & Schaffert, Craig. 1977. Abstraction Mechanisms in CLU. *Communications of the ACM*, **20**, 564–576.

McCarthy, John. 1960. Recursive Functions of Symbolic Expressions and their Computation by Machine, Part I. *Communications of the ACM*, **3**, 184–195.

McCarthy, John. 1962. Towards a Mathematical Science of Computation. *Pages 21–28 of:* Popplewell (ed.), *Information Processing 62*. Amsterdam: North-Holland.

McCarthy, John, *et al.*. 1965. *LISP 1.5 Programmer's Manual*. Cambridge, MA: MIT Press.

Milner, R. 1978. A Theory of Type Polymorphism in Programming. *Journal of Computer and Systems Science*, **17**, 348–375.

Milner, Robin, Tofte, Mads, & Harper, Robert. 1989. *The Definition of Standard ML*. Cambridge, MA: MIT Press.

Milner, Robin, Tofte, Mads, Harper, Robert, & MacQueen, David B. 1997. *The Standard ML Programming Language (Revised)*. MIT Press.

Morris, Jr., James H. 1968. *Lambda Calculus Models of Programming Languages*. Ph.D. thesis, MIT, Cambridge, MA.

Naur, P., *et al.*. 1963. Revised Report on the Algorithmic Language ALGOL 60. *Communications of the ACM*, **5**(1), 1–17.

Nielson, Flemming, Nielson, Hanne Riis, & Hankin, Chris. 1999. *Principles of Program Analysis*. Berlin, Heidelberg, and New York: Springer-Verlag.

Okasaki, Chris. 1998. *Purely Functional Data Structures*. Cambridge, UK: Cambridge University Press.

Parnas, David L. 1972. A Technique for Module Specification with Examples. *Communications of the ACM*, **15**(5), 330–336.

Paulson, Laurence C. 1996. *ML for the Working Programmer*. Second edition. New York, NY: Cambridge University Press.

Peyton Jones, Simon L. 1987. *The Implementation of Functional Programming Languages*. Prentice-Hall International.

Plotkin, Gordon D. 1975. Call-by-Name, Call-by-Value and the λ-Calculus. *Theoretical Computer Science*, **1**, 125–159.

Pratt, Terrence W., & Zelkowitz, Marvin V. 1996. *Programming Languages: Design and Implementation*. Third edition. Englewood Cliffs, NJ: Prentice-Hall.

Rees, Jonathan A., Clinger, William D., *et al.*. 1986. Revised^3 Report on the Algorithmic Language Scheme. *SIGPLAN Notices*, **21**(12), 37–79.

Reynolds, John C. 1972. Definitional Interpreters for Higher-Order Programming Languages. *Pages 717–740 of: Proceedings ACM National Conference*. Reprinted as (Reynolds, 1998).

Reynolds, John C. 1975. User-Defined Types and Procedural Data Structures as Complementary Approaches to Data Abstraction. *In: Conference on New Directions on Algorithmic Languages*. IFIP WP 2.1, Munich.

Reynolds, John C. 1993. The Discoveries of Continuations. *Lisp and Symbolic Computation*, **6**(3/4), 233–248.

Reynolds, John C. 1998. Definitional Interpreters for Higher-Order Programming Languages. *Higher-Order and Symbolic Computation*, **11**(4), 363–397. Originally appeared as (Reynolds, 1972).

Robinson, J. Alan. 1965. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, **12**, 23–41.

Sabry, Amr, & Felleisen, Matthias. 1993. Reasoning about Programs in Continuation-Passing Style. *Lisp and Symbolic Computation*, **6**(3/4), 289–360.

Sabry, Amr, & Wadler, Philip. 1997. A Reflection on Call-by-Value. *ACM Transactions on Programming Languages and Systems*, **19**(6), 916–941.

Scott, Michael L. 2000. *Programming Language Pragmatics*. San Francisco: Morgan Kaufmann.

Sethi, Ravi. 1996. *Programming Languages: Concepts and Constructs*. Second edition. Reading, MA.: Addison-Wesley.

Smith, Brian C. 1982 (Jan.). *Reflection and Semantics in a Procedural Language*. Technical Report MIT/LCS/TR-272. Massachusetts Institute of Technology, Cambridge, MA.

Smith, Brian C. 1984. Reflection and Semantics in Lisp. *Pages 23–35 of: Conf. Rec. 11th ACM Symposium on Principles of Programming Languages*.

Springer, George, & Friedman, Daniel P. 1989. *Scheme and the Art of Programming*. New York, NY: McGraw-Hill.

Steel, T. B. (ed.). 1966. *Formal Language Description Languages for Computer Programming*. Amsterdam, NL: North-Holland. Proceedings of the IFIP Working Conference, Vienna.

Steele, Guy L. 1978. *Rabbit: A Compiler for Scheme*. Artificial Intelligence Laboratory Technical Report 474. Massachusetts Institute of Technology, Cambridge, MA.

Steele, Guy L. 1990. *Common Lisp: the Language*. Second edition. Burlington MA: Digital Press.

Steele, Guy L., & Sussman, Gerald Jay. 1978. *The Revised Report on SCHEME*. Artificial Intelligence Memo 452. Massachusetts Institute of Technology, Cambridge, MA.

Stoy, Joseph E. 1977. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*. Cambridge, MA: MIT Press.

Strachey, Christopher, & Wadsworth, C. P. 1974. *Continuations: A Mathematical Semantics for Handling Full Jumps*. Technical Monograph PRG-11. Oxford University Computing Laboratory. Reprinted as (Strachey & Wadsworth, 2000).

Strachey, Christopher, & Wadsworth, C. P. 2000. Continuations: A Mathematical Semantics for Handling Full Jumps. *Higher-Order and Symbolic Computation*, **13**(1/2), 135–152.

Sussman, Gerald J., & Steele, Guy L. 1975. *SCHEME: An Interpreter for Extended Lambda Calculus*. Artificial Intelligence Memo 349. Massachusetts Institute of Technology, Cambridge, MA. Reprinted as (Sussman & Steele, 1998).

Sussman, Gerald J., & Steele, Guy L. 1998. SCHEME: An Interpreter for Extended Lambda Calculus. *Higher-Order and Symbolic Computation*, **11**(4), 405–439. Originally appeared as (Sussman & Steele, 1975).

Ullman, J. D. 1998. *Elements of ML Programming*. ML97 edition. Prentice-Hall.

Wand, Mitchell. 1980a. Continuation-Based Multiprocessing. *Pages 19–28 of:* Allen, J. (ed.), *Conference Record of the 1980 LISP Conference*. Palo Alto, CA: The Lisp Company, republished by ACM. Reprinted as (Wand, 1999).

Wand, Mitchell. 1980b. Continuation-Based Program Transformation Strategies. *Journal of the ACM*, **27**, 164–180.

Wand, Mitchell. 1999. Continuation-Based Multiprocessing. *Higher-Order and Symbolic Computation*, **12**(3), 285–299. Originally appeared as (Wand, 1980a).

Wexelblat, R. L. 1981. *History of Programming Languages*. New York: Academic Press.