Bibliography

Aagaard, M. and Leeser, M. (1994) Verifying a logic synthesis tool in Nuprl: A case study in software verification. In v. Bochmann, G. and Probst, D. K. (eds.), *Computer Aided Verification: Proceedings of the Fourth International Workshop, CAV'92*, Volume 663 of *Lecture Notes in Computer Science*, Montreal, Canada, pp. 69–81. Springer Verlag.

Abramsky, S. (1990) The lazy lambda-calculus. In Turner, D. A. (ed.), *Research Topics in Functional Programming*, Year of Programming series, pp. 65–116. Addison-Wesley.

Adel'son-Vel'skii, G. M. and Landis, E. M. (1962) An algorithm for the organization of information. *Soviet Mathematics Doklady*, **3**, 1259–1262.

Backus, J. (1978) Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Communications of the ACM*, **21**, 613–641.

Barendregt, H. P. (1984) *The Lambda Calculus: Its Syntax and Semantics*, Volume 103 of *Studies in Logic and the Foundations of Mathematics.* North-Holland.

Barwise, J. (1989) Mathematical proofs of computer correctness. *Notices of the American Mathematical Society*, **7**, 844–851.

Beckert, B. and Posegga, J. (1995) leanT^AP: Lean, tableau-based deduction. *Journal of Automated Reasoning*, **15**, 339–358. Also available on the Web from `http://i12www.ira.uka.de/~posegga/LeanTaP.ps.Z`.

Boehm, H. J., Cartwright, R., O'Donnel, M. J., and Riggle, M. (1986) Exact real arithmetic: a case study in higher order programming. In *Conference Record of the 1986 ACM Symposium on LISP and Functional Programming*, pp. 162–173. Association for Computing Machinery.

Boizumault, P. (1993) *The implementation of Prolog.* Princeton series in computer science. Princeton University Press. Translated from 'Prolog: l'implantation' by A. M. Djamboulian and J. Fattouh.

Boyer, R. S. and Moore, J. S. (1979) *A Computational Logic.* ACM Monograph Series. Academic Press.

Burge, W. H. (1975) *Recursive Programming Techniques.* Addison-Wesley.

Church, A. (1936) An unsolvable problem of elementary number-theory. *American Journal of Mathematics*, **58**, 345–363.

Church, A. (1940) A formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, **5**, 56–68.

Church, A. (1941) *The calculi of lambda-conversion*, Volume 6 of *Annals of Mathematics Studies.* Princeton University Press.

Clocksin, W. F. and Mellish, C. S. (1987) *Programming in Prolog* (3rd ed.). Springer-Verlag.

Curry, H. B. (1930) Grundlagen der Kombinatorischen Logik. *American Journal of Mathematics*, **52**, 509–536, 789–834.

Davenport, J. H. (1981) *On the integration of algebraic functions*, Volume 102 of *Lecture Notes in Computer Science.* Springer-Verlag.

Davenport, J. H., Siret, Y., and Tournier, E. (1988) *Computer algebra: systems and algorithms for algebraic computation.* Academic Press.

Davis, M. D., Sigal, R., and Weyuker, E. J. (1994) *Computability, complexity, and languages: fundamentals of theoretical computer science* (2nd ed.). Academic Press.

de Bruijn, N. G. (1972) Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae*, **34**, 381–392.

DeMillo, R., Lipton, R., and Perlis, A. (1979) Social processes and proofs of theorems and programs. *Communications of the ACM*, **22**, 271–280.

Dijkstra, E. W. (1976) *A Discipline of Programming.* Prentice-Hall.

Enderton, H. B. (1972) *A Mathematical Introduction to Logic.* Academic Press.

Frege, G. (1893) *Grundgesetze der Arithmetik begriffsschrift abgeleitet.* Jena. Partial English translation by Montgomery Furth in 'The basic laws of arithmetic. Exposition of the system', University of California Press, 1964.

Girard, J.-Y., Lafont, Y., and Taylor, P. (1989) *Proofs and Types*, Volume 7 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press.

Gordon, A. D. (1994) *Functional Programming and Input/Output.* Distinguished Dissertations in Computer Science. Cambridge University Press.

Gordon, M. J. C. (1988) *Programming Language Theory and its Implementation: applicative and imperative paradigms.* Prentice-Hall International Series in Computer Science. Prentice-Hall.

Gordon, M. J. C., Milner, R., and Wadsworth, C. P. (1979) *Edinburgh LCF: A Mechanised Logic of Computation*, Volume 78 of *Lecture Notes in Computer Science*. Springer-Verlag.

Henson, M. C. (1987) *Elements of functional languages.* Blackwell Scientific.

Hindley, J. R. and Seldin, J. P. (1986) *Introduction to Combinators and λ-Calculus*, Volume 1 of *London Mathematical Society Student Texts*. Cambridge University Press.

Hudak, P. (1989) Conception, evolution, and application of functional programming languages. *ACM Computing Surveys*, **21**, 359–411.

Huet, G. (1980) Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM*, **27**, 797–821.

Kleene, S. C. (1935) A theory of positive integers in formal logic. *American Journal of Mathematics*, **57**, 153–173, 219–244.

Lagarias, J. (1985) The $3x + 1$ problem and its generalizations. *The American Mathematical Monthly*, **92**, 3–23. Available on the Web as `http://www.cecm.sfu.ca/organics/papers/lagarias/index.html`.

Landin, P. J. (1966) The next 700 programming languages. *Communications of the ACM*, **9**, 157–166.

Lindemann, F. (1882) Über die Zahl π. *Mathematische Annalen*, **120**, 213–225.

Mairson, H. G. (1990) Deciding ML typability is complete for deterministic exponential time. In *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages (POPL)*, San Francisco, pp. 382–401. Association for Computing Machinery.

Mairson, H. G. (1991) Outline of a proof theory of parametricity. In Hughes, J. (ed.), *1991 ACM Symposium on Functional Programming and Computer Architecture*, Volume 523 of *Lecture Notes in Computer Science*, Harvard University, pp. 313–327.

Martelli, A. and Montanari, U. (1982) An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, **4**, 258–282.

Mauny, M. (1995) Functional programming using CAML Light. Available on the Web from `http://pauillac.inria.fr/caml/tutorial/index.html`.

Ménissier-Morain, V. (1994) *Arithmétique exacte, conception, algorithmique et performances d'une implémentation informatique en précision arbitraire*. Thèse, Université Paris 7.

Michie, D. (1968) "Memo" functions and machine learning. *Nature*, **218**, 19–22.

Milner, R. (1978) A theory of type polymorphism in programming. *Journal of Computer and Systems Sciences*, **17**, 348–375.

Mycroft, A. (1981) Abstract interpretation and optimising transformations of applicative programs. Technical report CST-15-81, Computer Science Department, Edinburgh University, King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK.

Neumann, P. G. (1995) *Computer-related risks*. Addison-Wesley.

Oppen, D. (1980) Prettyprinting. *ACM Transactions on Programming Languages and Systems*, **2**, 465–483.

Paulson, L. C. (1983) A higher-order implementation of rewriting. *Science of Computer Programming*, **3**, 119–149.

Paulson, L. C. (1991) *ML for the Working Programmer*. Cambridge University Press.

Pelletier, F. J. (1986) Seventy-five problems for testing automatic theorem provers. *Journal of Automated Reasoning*, **2**, 191–216. Errata, JAR 4 (1988), 235–236.

Peterson, I. (1996) *Fatal Defect : Chasing Killer Computer Bugs*. Arrow.

Potts, P. (1996) Computable real arithmetic using linear fractional transformations. Unpublished draft for PhD thesis, available on the Web as `http://theory.doc.ic.ac.uk/~pjp/pub/phd/draft.ps.gz`.

Raphael, B. (1966) The structure of programming languages. *Communications of the ACM*, **9**, 155–156.

Reade, C. (1989) *Elements of Functional Programming*. Addison-Wesley.

Reynolds, J. C. (1993) The discoveries of continuations. *Lisp and Symbolic Computation*, **6**, 233–247.

Robinson, J. A. (1994) Logic, computers, Turing and von Neumann. In Furukawa, K., Michie, D., and Muggleton, S. (eds.), *Machine Intelligence 13*, pp. 1–35. Clarendon Press.

Schönfinkel, M. (1924) Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, **92**, 305–316. English translation, 'On the building blocks of mathematical logic' in van Heijenoort (1967), pp. 357–366.

Schwichtenberg, H. (1976) Definierbare Funktionen im λ-Kalkül mit Typen. *Arkhiv für mathematische Logik und Grundlagenforschung*, **17**, 113–114.

Stickel, M. E. (1988) A Prolog Technology Theorem Prover: Implementation by an extended Prolog compiler. *Journal of Automated Reasoning*, **4**, 353–380.

Turing, A. M. (1936) On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society (2)*, **42**, 230–265.

Turing, A. M. (1937) Correction to Turing (1936). *Proceedings of the London Mathematical Society (2)*, **43**, 544–546.

van Heijenoort, J. (ed.) (1967) *From Frege to Gödel: A Source Book in Mathematical Logic 1879–1931*. Harvard University Press.

Whitehead, A. N. (1919) *An Introduction to Mathematics*. Williams and Norgate.

Whitehead, A. N. and Russell, B. (1910) *Principia Mathematica (3 vols)*. Cambridge University Press.

Winskel, G. (1993) *The formal semantics of programming languages: an introduction*. Foundations of computing. MIT Press.

Wittgenstein, L. (1922) *Tractatus Logico-Philosophicus*. Routledge & Kegan Paul.

Wright, A. (1996) Polymorphism for imperative languages without imperative types. Technical Report TR93-200, Rice University.