Bibliography

[ABL86] R.M. Amadio, K. Bruce, and G. Longo. The finitary projection model for second order lambda calculus and solutions to higher order domain equations. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 122–130, 1986.

[Abr90] S. Abramsky. Abstract interpretation, logical relations and Kan extensions. *J. of Logic and Computation*, 1(1):5–40, 1990.

[AC80] Egidio Astesiano and Gerardo Costa. Nondeterminism and fully abstract models. *RAIRO*, 14(4):323–347, 1980.

[AC91] R.M. Amadio and L. Cardelli. Subtyping recursive types. In *Proc. ACM Symp. Principles of Programming Languages*, pages 104–118, 1991.

[AC94a] M. Abadi and L. Cardelli. A theory of primitive objects: second-order systems. In *Proc. European Symposium on Programming*, pages 1–24. Springer-Verlag, 1994.

[AC94b] M. Abadi and L. Cardelli. A theory of primitive objects: untyped and first-order systems. In *Proc. Theor. Aspects of Computer Software*, pages 296–320. Springer-Verlag LNCS 789, 1994.

[AC95] M. Abadi and L. Cardelli. On subtyping and matching. In *Proc. European Conf. on Object-Oriented Programming*, page ?? ??, 1995.

[ACP+92] M. Abadi, L. Cardelli, B. Pierce, G. Plotkin, and D. Rèmy. Dynamic typing in polymorphic languages. In *Proceedings of the ACM SIGPLAN Workshop on ML and its Applications*, San Francisco, June 1992. ACM.

[ACPP91] M. Abadi, L. Cardelli, B. Pierce, and G. Plotkin. Dynamic typing in a statically typed language. *ACM Transactions on Programming Languages and Systems*, 13(2):237–268, April 1991.

[Acz77] P. Aczel. An introduction to inductive definitions. In *Handbook of Mathematical Logic*, pages 739–782. North-Holland, Amsterdam, 1977.

[AHU83] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *Data Structures and Algorithms.* Addison-Wesley, 1983.

[AJ91] S. Abramsky and T.P. Jensen. A relational approach to strictness analysis for higher-order polymorphic functions. In *Proc. ACM Symp. Principles of Programming Languages*, pages 49–54, 1991.

[AL91] A. Asperti and G. Longo. *Categories, Types and Structures.* MIT Press, 1991.

[All86] L. Allison. *A practical introduction to denotational semantics.* Cambridge Computer Science Texts 23. Cambridge Univ Press, 1986.

[AM75] M.A. Arbib and E.G. Manes. *Arrows, Structures, and Functors: The Categorical Imperative.* Academic Press, 1975.

[Ama91] R.M. Amadio. Recursion over realizability structures. *Information and Computation*, 91(1):55–86, 1991.

[And86] P.B. Andrews. *An introduction to mathematical logic and type theory: to truth through proof.* Academic Press, 1986.

[AO91] K.R. Apt and E.-R. Olderog. *Verification of sequential and concurrent programs.* Springer-Verlag, New York, Berlin, 1991.

[AP90] M Abadi and G.D. Plotkin. A PER model of polymorphism and recursive types. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 355–365, 1990.

[Apt81] K.R. Apt. Ten years of hoare's logic: A survey – part i. *ACM Trans. on Prog. Lang. and Systems*, 3(4):431–483, Oct. 1981.

[AS85] H. Abelson and G.J. Sussman. *Structure and Interpretation of Computer Programs.* MIT Press, 1985.

[ASU86] A.V. Aho, R. Sethi, and J.D. Ullman. *Compilers: principles, techniques, tools.* Addison-Wesley, 1986.

[AU92] A.V. Aho and J.D. Ullman. *Foundations of Computer Science.* Computer Science Press, W.H. Freeman and Co., New York, 1992.

[Bar84] H.P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* North-Holland, Amsterdam, 1984. Second edition.

[Bar90] M. Barr. Fixed points in cartesian closed categories. *Theoretical Computer Science*, 70:65–72, 1990.

[BCDC83] H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *J. Symbolic Logic*, 48(4):931–940, 1983.

[Bee85] M. Beeson. *Foundations of Constructive Mathematics.* Springer-Verlag, Berlin, 1985.

[Bel86] F. Bellegarde. Rewriting systems on FP expressions to reduce the number of sequences yielded. *Science of Computer Programming*, 6:11–34, 1986.

[Ben86] J. Bentley. *Programming Pearls.* Addison-Wesley, 1986.

[BFSS89] E.S. Bainbridge, P.J. Freyd, A. Scedrov, and P.J. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70:35–64, 1989. Corrigendum *ibid.*, 71:431, 1990. Preliminary report in *Logical Foundations of Functional Programming,* ed. G. Huet, Addison-Wesley (1990) 315–327.

[BG66] C. Böhm and W. Gross. Introduction to the CUCH. In E.R. Caianiello, editor, *Automata Theory*, pages 35–65. Academic Press, New York, 1966.

[BHA86] G.L. Burn, C. Hankin, and S. Abramsky. Strictness analysis for higher-order functions. *Science of Computer Programming*, 7:249–278, 1986.

[BJ82] D. Bjorner and C.B. Jones. *Formal Specification and Software Development.* Prentice-Hall, 1982.

[BL90] K. Bruce and G. Longo. A modest model of records, inheritance and bounded quantification. *Information and Computation*, 87(1/2):196–240, 1990. Reprinted in [GM94].

[BM92] K. Bruce and J.C. Mitchell. PER models of subtyping, recursive types and higher-order polymorphism. In *Proc. 19th ACM Symp. on Principles of Programming Languages*, pages 316–327, January 1992.

[BMM90] K. B. Bruce, A. R. Meyer, and J. C. Mitchell. The semantics of second-order lambda calculus. *Information and Computation*, 85(1):76–134, 1990. Reprinted in *Logical Foundations of Functional Programming,* ed. G. Huet, Addison-Wesley (1990) 213–273.

[Boo91] G. Booch. *Object-Oriented Design with Applications.* Benjamin Cummings, Redwood City, CA, 1991.

[Bru94] Kim B. Bruce. A paradigmatic object-oriented programming language: Design, static typing and semantics. *J. Functional Programming*, 4(2):127–206, 1994.

[BT83] J.A. Bergstra and J.V. Tucker. Initial and final algebra semantics for data type specifications: two characterization theorems. *SIAM J. Computing*, 12:366–387, 1983.

[BT88] V. Breazu-Tannen. Combining algebra and higher-order types. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 82–90, 1988.

[BTCGS91] V. Breazu-Tannen, T. Coquand, C.A. Gunter, and A. Scedrov. Inheritance as explicit coercion. *Information and Computation*, 93(1):172–221, 1991. Reprinted in [GM94].

[BTG89] V. Breazu-Tannen and J.H. Gallier. Polymorphic rewriting conserves algebraic strong normalization and confluence. In *16th Int'l Colloq. on Automata, Languages and Programming*, pages 137–159, Berlin, 1989. Springer LNCS 37.

[BW90] M. Barr and C. Wells. *Category theory for computing science*. Prentice Hall International, 1990.

[C$^+$86] R.L. Constable et al. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, 1986.

[Car84] L. Cardelli. A semantics of multiple inheritance. In *Proc. Int. Symp. on Semantics of Data Types, Sophia-Antipolis (France)*, pages 51–68, Berlin, June 1984. Springer LNCS 173.

[Car86] L. Cardelli. Amber. In *Combinators and Func. Programming*, pages 21–47. Springer LNCS 242, Berlin, 1986.

[Car89] F. Cardone. Relational semantics for recursive types and bounded quantification. In *ICALP*, pages 164–178, Berlin, 1989. Springer LNCS 372.

[CD91] P.-L. Curien and R. Di Cosmo. A confluent reduction for the lambda calculus with surjective pairing and terminal object. In *Proc. 1990 ICALP Conference*, pages 291–302, Berlin, 1991. Springer LNCS 510.

[CDCV80] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. Principal type schemes and lambda calculus semantics. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 535–560. Academic Press, 1980.

[CDCZ87] M. Coppo, M. Dezani-Ciancaglini, and M. Zacchi. Type theories, normal forms and D_∞ lambda models. *Information and Computation*, 72:85–116, 1987.

[CDG$^+$88] L. Cardelli, J. Donahue, L. Glassman, M. Jordan, B. Kalsow, and G. Nelson. Modula-3 report. Technical Report SRC-31, DEC systems Research Center, 1988.

[CDJ$^+$89] L. Cardelli, J. Donahue, M. Jordan, B. Kalsow, and G. Nelson. The Modula-3 type system. In *Sixteenth ACM Symp. Principles of Programming Languages*, pages 202–212, 1989.

[CF58] H.B Curry and R. Feys. *Combinatory Logic I*. North-Holland, Amsterdam, 1958.

[CG92] P.-L. Curien and G. Ghelli. Coherence of subsumption, minimum typing and type-checking in $f_\leq$. *Math. Struct. in Comp. Sci.*, 2:55–91, 1992. Reprinted in [GM94].

[CGW89] T. Coquand, C.A. Gunter, and G. Winskel. Domain-theoretic models of polymorphism. *Information and Computation*, 81(2):123–167, 1989.

[CH88] T. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76(2/3):95–120, 1988.

[CH94a] T. Coquand and H. Herbelin. A-translation and looping combinators in pure type systems. *J. Functional Programming*, 4(1):77–88, 1994.

[CH94b] P.-L. Curien and T. Hardin. Theoretical Pearl: Yet yet a counterexample for $\lambda + sp$. *J. Functional Prog.*, 4(1):113–115, 1994.

[Chu32] A. Church. A set of postulates for the foundation of logic. *Ann. of Math.*, 33:346–366, 1932. Second paper with same title in Vol. 33, pages 839–864, of same journal.

[Chu36] A. Church. An unsolvable problem of elementary number theory. *Amer. J. Math.*, 58:345–363, 1936.

[Chu40] A. Church. A formulation of the simple theory of types. *J. Symbolic Logic*, 5:56–68, 1940.

[Chu41] A. Church. *The Calculi of Lambda Conversion*. Princeton Univ. Press, 1941. Reprinted 1963 by University Microfilms Inc., Ann Arbor, MI.

[CMMS94] L. Cardelli, S. Martini, J. Mitchell, and A. Scedrov. An extension of System F with subtyping. *Information and Computation*, 109:4–56, 1994. Preliminary version appeared *Proc. Theor. Aspects of Computer Software,* Springer LNCS 526, September 1991, pages 750–770.

[Coo78] S.A. Cook. Soundness and completeness of an axiom system for program verification. *SIAM J. Computing*, 7:129–147, 1978.

[Coo89] W.R. Cook. *A Denotational Semantics of Inheritance*. PhD thesis, Brown University, 1989.

[Coq86] T. Coquand. An analysis of Girard's paradox. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 227–236, June 1986.

[CP92] R.L. Crole and A.M. Pitts. New foundations for fixpoint computations: FIX-hyperdoctrines and the FIX-logic. *Information and Computation*, 98(2):171–210, 1992.

[CR36] A. Church and J.B. Rosser. Some properties of conversion. *Trans. Amer. Math. Soc.*, 39:472–482, 1936.

[Cur86] P.-L. Curien. *Categorical combinators, sequential algorithms, and functional programming*. Pitman, London, and John Wiley and Sons, New York, 1986. second edition???

[Cut80] N.J. Cutland. *Computability: An introduction to recursive function theory*. Cambridge Univ. Press, Cambridge, 1980.

[CW85] L. Cardelli and P. Wegner. On understanding types, data abstraction, and polymorphism. *Computing Surveys*, 17(4):471–522, 1985.

[CZ86] M. Coppo and M. Zacchi. Type inference and logical relations. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 218–226, June 1986.

[DB80] N.G. De Bruijn. A survey of the project Automath. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 579–607. Academic Press, 1980.

[Dij75] E.W. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs. *CACM*, 18(8):453–457, 1975.

[Dij76] E.W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B*, pages 243–320. North-Holland, Amsterdam, 1990.

[DKM84] C. Dwork, P. Kanellakis, and J.C. Mitchell. On the sequential nature of unification. *J. Logic Programming*, 1:35–50, 1984.

[DM82] L. Damas and R. Milner. Principal type schemes for functional programs. In *Proc. 9th ACM Symposium on Principles of Programming Languages*, pages 207–212, 1982.

[Don79] J. Donahue. On the semantics of data type. *SIAM J. Computing*, 8:546–560, 1979.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1*. Springer-Verlag, Berlin, 1985.

[End72] H.B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 1972.

[Ers71] Yu. L. Ershov. Computable numerations of morphisms. *Algebra i Logika*, 10(3):247–308, 1971.

[Ers72] Yu. L. Ershov. Computable functions of finite types. *Algebra i Logika*, 11(4):367–437, 1972.

[Ers76] Yu. L. Ershov. Hereditarily effective operations. *Algebra i Logika*, 15(6):642–654, 1976.

[ES90] M. Ellis and B. Stroustrop. *The Annotated C^{++} Reference Manual.* Addison-Wesley, 1990.

[Fef79] S. Feferman. Constructive theories of functions and classes. In *Logic Colloquium '78*, pages 159–224, Amsterdam, 1979. North-Holland.

[FGSS88] P. Freyd, J.-Y. Girard, A. Scedrov, and P.J. Scott. Semantic parametricity in polymorphic lambda calculus. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 274–279, July 1988.

[FHM94] K. Fisher, F. Honsell, and J.C. Mitchell. A lambda calculus of objects and method specialization. *Nordic J. Computing (*formerly *BIT)*, 1:3–37, 1994. Preliminary version appeared in *Proc. IEEE Symp. on Logic in Computer Science,* 1993, 26–38.

[Fit69] M.C. Fitting. *Intuitionistic Logic, Model Theory and Forcing.* North-Holland, Amsterdam, 1969.

[Flo67] R.W. Floyd. Assigning meaning to programs. In J.T. Schwartz, editor, *Proc. Symp. in Applied Mathematics*, pages 19–32. AMS, 1967.

[FLO83] S. Fortune, D. Leivant, and M. O'Donnell. The expressiveness of simple and second order type structures. *JACM*, 30(1):151–185, 1983.

[Fre88] P. Freyd. Polynat in PER, 1988.

[Fri75] H. Friedman. Equality between functionals. In R. Parikh, editor, *Logic Colloquium*, pages 22–37. Springer-Verlag, Berlin, 1975.

[FRMS92] P. Freyd, G. Rosolini, P. Mulry, and D.S. Scott. Extensional PER's. *Information and Computation*, 98(2):211–227, 1992. Preliminary version appeared in *Proc. IEEE Symp. on Logic in Computer Science,* IEEE, 1990, 346–354.

[FS87] P. Freyd and A. Scedrov. Some semantic aspects of polymorphic lambda calculus. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 315–319, June 1987.

[FS90] P. Freyd and A. Scedrov. *Categories, Allegories.* Mathematical Library, North-Holland, 1990.

[Gab81] D.M. Gabbay. *Semantical Investigations in Heyting's Intuitionistic Logic.* D. Reidel, 1981.

[Gan56] R.O. Gandy. On the axiom of extensionality – part i. *J. Symbolic Logic*, 21:??–??, 1956.

[Gan80] R.O. Gandy. An early proof of normalization by A.M. Turing. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 453–455. Academic Press, 1980.

[GHM78] J.V. Guttag, E. Horowitz, and D.R. Musser. Abstract data types and software validation. *Comm. ACM*, 21(12):1048–1064, 1978.

[Gir71] J.-Y. Girard. Une extension de l'interpretation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J.E. Fenstad, editor, *2nd Scandinavian Logic Symposium*, pages 63–92. North-Holland, Amsterdam, 1971.

[Gir72] J.-Y. Girard. Interpretation fonctionelle et elimination des coupures de l'arithmetique d'ordre superieur. These D'Etat, Universite Paris VII, 1972.

[Gir86] J.-Y. Girard. The system F of variable types, fifteen years later. *Theor. Comp. Sci.*, 45(2):159–192, 1986.

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[GM94] C.A. Gunter and J.C. Mitchell, editors. *Theoretical aspects of object-oriented programming.* MIT Press, Cambridge, MA, 1994.

[GMW79] M.J. Gordon, R. Milner, and C.P. Wadsworth. *Edinburgh LCF.* Springer LNCS 78, Berlin, 1979.

[Göd58] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.

[Gog78] J.A. Goguen. Order-sorted algebras. Technical Report 14, Computer Science Dept., UCLA, 1978.

[Gol79] R. Goldblatt. *Topoi: The categorical analysis of logic.* North-Holland, 1979.

[Gor79] M.J.C. Gordon. *The Denotational Description of Programming Languages.* Springer-Verlag, Berlin, 1979.

[Gor93] M. J. C. Gordon. *Introduction to HOL: A Theorem Proving Environment.* Cambridge University Press, 1993.

[GR83] A. Goldberg and D. Robson. *Smalltalk–80: The language and its implementation.* Addison Wesley, 1983.

[Grä68] G. Grätzer. *Universal Algebra.* Van Nostrand, 1968.

[GS90] C.A. Gunter and D.S. Scott. Semantic domains. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B*, pages 633–674. North-Holland, Amsterdam, 1990.

[GTW78] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In R.T. Yeh, editor, *Current Trends in Programming Methodology*, volume 4. Prentice-Hall, 1978.

[Gun92] C.A. Gunter. *Semantics of Programming Languages: Structures and Techniques.* MIT Press, Cambridge, MA, 1992.

[Gut77] J.V. Guttag. Abstract data types and the development of data structures. *Comm. ACM*, 20(6):396–404, 1977.

[H+92] P. Hudak et al. Report on the programming language Haskell. *SIGPLAN Notices*, 27(5):Section R, 1992.

[Hal60] P.R. Halmos. *Naive Set Theory.* D. Van Nostrand, 1960.

[Har84] D. Harel. Dynamic logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, II: Extensions of Classical Logic*, pages 497–604. Reidel, Boston, 1984.

[Hay84] C.T. Haynes. A theory of data type representation independence. In *Proc. Int. Symp. on Semantics of Data Types, Sophia-Antipolis (France)*, pages 157–176, Berlin, 1984.

[Hen50] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, June 1950.

[Hen63] L. Henkin. A propositional theory of types. *Fund. Math.*, 52:323–344, 1963.

[Hen90a] F. Henglein. A lower bound for full polymorphic type inference: Girard/reynolds typability is DEXPTIME-hard. Technical Report RUU-CS-90-14, Univerity of Utrecht, 1990.

[Hen90b] M. Hennessy. *The Semantics of Programming Languages: An elementary introduction using Structured Operational Semantics.* Wiley, 1990.

[HF92] P. Hudak and J. Fasel. A gentle introduction to Haskell. *SIGPLAN Notices*, 27(5):Section T, 1992.

[HHP87] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 194–204, June 1987. To appear in *J. Assoc. Comput. Machinery.*

[Hin69] J.R. Hindley. The principal type-scheme of an object in combinatory logic. *Trans. AMS*, 146:29–60, 1969.

[Hin83] J.R. Hindley. The completeness theorem for typing lambda terms. *Theor. Comp. Sci.*, 22:1–17, 1983.

[HL80] J.R. Hindley and G. Longo. Lambda calculus models and extensionality. *Z. Math. Logik Grundlag Math*, 26:289–310, 1980.

[HM90] B.T. Howard and J.C. Mitchell. Operational and axiomatic semantics of PCF. In *ACM Conference on LISP and Functional Programming*, pages 298–306, 1990.

[HM93] R. Harper and J.C. Mitchell. On the type structure of Standard ML. *ACM Trans. Programming Lang. and Systems*, 15(2):211–252, 1993. Earlier version appears as "The Essence of ML" in *Proc. 15th ACM Symp. on Principles of Programming Languages,* 1988, pp. 28–46.

[HM94] F. Henglein and H. Mairson. The complexity of type inference for higher-order typed lambda calculi. *J. Functional Programming*, 4(4):435–478, 1994.

[HM95a] R. Harper and G. Morrisett. Compiling polymorphism using intensional type analysis. In *Proc. 22nd ACM Symp. on Principles of Programming Languages*, page to appear, January 1995.

[HM95b] M. Hoang and J.C. Mitchell. Lower bounds for type inference with subtyping. In *ACM Symp. Principles of Programming Languages*, pages 176–185, 1995.

[HMT87] R. Harper, R. Milner, and M. Tofte. A type discipline for program modules. In *TAPSOFT '87*, Berlin, 1987. Springer LNCS 250.

[HN88] S. Hayashi and H. Nakano. *PX – A Computational Logic.* MIT Press, 1988.

[HO80] G. Huet and D.C. Oppen. Equations and rewrite rules: a survey. In R. Book, editor, *Formal Languages: Perspectives and Open Problems.* Academic Press, 1980. Revised version appears as SRI International Tech. Report CSL-111, 1980.

[Hoa69] C.A.R. Hoare. An axiomatic basis for computer programming. *CACM*, 12(10):576–580, 1969.

[Hoa72] C.A.R. Hoare. Proof of correctness of data representations. *Acta Informatica*, 1:271–281, 1972.

[Hoa95] M. Hoang. *Type Inference and Program Evaluation in the Presence of Subtyping.* PhD thesis, Stanford University, 1995.

[Hor84] E. Horowitz. *Fundamentals of Programming Languages.* Computer Science Press, 1984.

[How73] W. Howard. Hereditarily majorizable functionals. In *Mathematical Investigation of Intuitionistic Arithmetic and Analysis*, pages 454–461. Springer LNM 344, Berlin, 1973.

[How80] W. Howard. The formulas-as-types notion of construction. In *To H.B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism*, pages 479–490. Academic Press, 1980.

[How87] D.J. Howe. The computational behavior of Girard's paradox. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 205–214, June 1987.

[How92] B.T. Howard. *Fixed points and extensionality in typed functional programming languages.* PhD thesis, Stanford University, 1992.

[HS86] J.R. Hindley and J.P. Seldin. *Introduction or Combinators and Lambda Calculus.* London Mathematical Society Student Texts 1, Cambridge University Press, 1986.

[HU79] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[Hue80] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *J. Assoc. Comput. Mach.*, 27(4):797–821, 1980.

[Hyl76] J.M.E. Hyland. A syntactic characterization of the equality in some models of the lambda calculus. *J. London Math. Society*, 2(12):361–370, 1976.

[Hyl82] J.M.E. Hyland. The effective topos. In *The L.E.J. Brouwer Centenary Symposium*, pages 165–216. North-Holland, Amsterdam, 1982.

[Hyl88] J.M.E. Hyland. A small complete category. *Ann. Pure and Applied Logic*, 40, 1988. Lecture delivered at the conference Church's Thesis: Fifty Years Later, Zeiss(NL), June 1986.

[Jac75] G. Jacopini. A condition for identifying two elements of whatever model of combinatory logic. In C. Böhm, editor, *Proc. Lambda calculus and computer science theory*, pages 213–219. Springer LNCS 37, 1975.

[Joh77] P. Johnstone. *Topos Theory.* Academic Press, 1977.

[Kam83] S. Kamin. Final datatypes and their specification. *ACM Trans. Prog. Languages and Systems*, 5:97–121, 1983.

[KB70] D.E. Knuth and P.B. Bendix. Simple word problems in universal algebra. In J. Leech, editor, *Computational problems in abstract algebra*, pages 263–297. Pergamon Press, Elmsford, N.Y, 1970.

[KJ35] S.C. Kleene and Rosser J.B. The inconsistency of certain formal logics. *Ann. of Math.*, 36:630–636, 1935.

[Kle36] S.C. Kleene. Lambda definability and recursiveness. *Duke Math. J.*, 2:340–353, 1936.

[Kle45] S.C. Kleene. On the interpretation of intuitionistic number theory. *J. Symbolic Logic*, 10:109–124, 1945.

[Kle71] S.C. Kleene. Realizability: A retrospective survey. In *Cambridge Summer School in Mathematical Logic*, pages 95–112. Springer LNM 337, Berlin, 1971.

[Klo80] J.W. Klop. *Combinatory Reduction Systems.* PhD thesis, University of Utrecht, 1980. Published as Mathematical Center Tract 129.

[Klo87] J.W. Klop. Term rewriting: a tutorial. *EATCS Bulletin*, 32:143–182, 1987.

[KM89] P.C. Kanellakis and J.C. Mitchell. Polymorphic unification and ML typing. In *16th ACM Symposium on Principles of Programming Languages*, pages 105–115, 1989.

[KMM91] P.C. Kanellakis, H.G. Mairson, and J.C. Mitchell. Unification and ML type reconstruction. In *Computational Logic, Essays in Honor of Alan Robinson*, pages 444–478. MIT Press, 1991.

[Koy82] C.P.J. Koymans. Models of the lambda calculus. *Information and Control*, 52(3):306–323, 1982.

[KR77] A. Kock and G.E. Reyes. Doctrines in categorical logic. In *Handbook of Mathematical Logic*, pages 283–316. North-Holland, Amsterdam, 1977.

[KR78] B.W. Kernighan and D.M. Ritchie. *The C programming language.* Prentice-Hall Software Series, 1978.

[Kre59] G. Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North-Holland, Amsterdam, 1959.

[Kri65] S.A. Kripke. Semantical analysis of intuitionistic logic i. In *Formal Systems and Recursive Functions*, pages 92–130. (Proc. 8th Logic Colloq. Oxford 1963) North-Holland, Amsterdam, 1965.

[KT90] D. Kozen and J. Tiuryn. Logics of programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B*, pages 789–840. North-Holland, Amsterdam, 1990.

[KTU88] A.J. Kfoury, J. Tiuryn, and P. Urzyczyn. A proper extension of ML with effective type assignment. In *Proc. 15th ACM Symp. Principles of Programming Languages*, pages 58–69, 1988.

[KTU89] A. Kfoury, J. Tiuryn, and P. Urzyczyn. Computational consequences and partial solutions of a generalized unification problem. In *Proc. IEEE Symp. on Logic in Computer Science*, June 1989.

[KTU90a] A. Kfoury, J. Tiuryn, and P. Urzyczyn. The undecidability of the semi-unification problem. In *Proc. 22nd Annual ACM Symp. on Theory of Computation (STOC), Baltimore, Maryland*, pages 468–476, May 1990.

[KTU90b] A.J. Kfoury, J. Tiuryn, and P. Urzyczyn. ML typability is Dexptime-complete. In *Proc. 15th Colloq. on Trees in Algebra and Programming*, pages 206–220. Springer LNCS 431, 1990. To appear in *J. Assoc. Comput. Machinery* under title, "An Analysis of ML Typability".

[L+81] B. Liskov et al. *CLU Reference Manual.* Springer LNCS 114, Berlin, 1981.

[Lam80] J. Lambek. From lambda calculus to cartesian closed categories. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 375–402. Academic Press, 1980.

[Lan63] P.J. Landin. The mechanical evaluation of expressions. *Comput. J.*, 6:308–320, 1963.

[Lan65] P.J. Landin. A correspondence between algol 60 and church's lambda notation. *CACM*, 8:89–101; 158–165, 1965.

[Lan66] P.J. Landin. The next 700 programming languages. *CACM*, 9:157–166, 1966.

[Lau65] H. Lauchli. Intuitionistic propositional calculus and definably non-empty terms. *Journal of Symbolic Logic*, 30:263, 1965.

[Lau70] H. Lauchli. An abstract notion of realizability for which intuitionistic predicate calculus is complete. In *Intuitionism and Proof Theory: Proc. of the Summer Conference at Buffalo N.Y.*, pages 227–234, 1970.

[Law69] F.W. Lawvere. Diagonal arguments and cartesian closed categories. In *Category Theory, Homology Theory and their Applications II*, pages 143–145. Springer LNM 92, 1969.

[Les92] P. Lescanne. Termination of rewrite systems by elementary interpretations. In H. Kirchner and G. Levi, editors, *Proceeding 3rd Conference on Algebraic and Logic Programming, Volterra (Italy)*, number 632 in Lecture Notes in Computer Science, pages 21–36. Springer-Verlag, September 1992.

[Lév75] J.-J. Lévy. An algebraic interpretation of the λ-β-k-calculus and a labeled λ-calculus. In C. Böhm, editor, *Proc. Lambda calculus and computer science theory*, pages 147–165. Springer LNCS 37, 1975.

[LG86] B. Liskov and J. Guttag. *Abstraction and Specification in Software Development.* MIT Press, 1986.

[Lip77] R. Lipton. A necessary and sufficient condition for the existence of hoare logics. In *Proc. 18th IEEE Symp. on Foundations of Computer Science*, pages 1–6, 1977.

[LM92] P.D. Lincoln and J.C. Mitchell. Algorithmic aspects of type inference with subtypes. In *Proc. 19th ACM Symp. on Principles of Programming Languages*, pages 293–304, January 1992.

[LMS93] G. Longo, K. Milstead, and S. Soloviev. The genericity theorem and the notion of parametricity in the polymorphic lambda calculus. In *Proc. IEEE Symp. Logic in Computer Science*, pages 6–14, 1993.

[Löb75] M.H. Löb. Embedding first-order predicate logic in fragments of intuitionistic logic. Technical Report 75-8, Mathematisch Institut, Amsterdam, 1975.

[LS86] J. Lambek and P.J. Scott. *Introduction to Higher-Order Categorical Logic.* Cambridge University Press, Cambridge, U.K, 1986.

[LSAS77] B. Liskov, A. Snyder, R. Atkinson, and C. Schaffert. Abstraction mechanisms in clu. *Comm. ACM*, 20:564–576, 1977.

[LZ74] B. Liskov and S. Zilles. Programming with abstract data types. *ACM Sigplan Notices*, 9:50–59, 1974.

[Mac71] S. MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1971.

[Mac85] D.B. MacQueen. Modules for Standard ML. *Polymorphism*, 2(2), 1985. 35 pages. An earlier version appeared in Proc. 1984 ACM Symp. on Lisp and Functional Programming.

[Mac86] D.B. MacQueen. Using dependent types to express modular structure. In *Proc. 13th ACM Symp. on Principles of Programming Languages*, pages 277–286, 1986.

[Mai90] H.G. Mairson. Deciding ml typability is complete for deterministic exponential time. In *Proc. 17th ACM Symp. Principles of Programming Languages*, pages 382–401, January 1990.

[Mal70] A Malcev. Algorithms and recursive functions. Technical report, ???, 1970. Translation Woolters, Ne, 1970.???

[Mar73] P. Martin-Löf. An intuitionistic theory of types: Predicative part. In H. E. Rose and J. C. Shepherdson, editors, *Logic Colloquium, '73*, pages 73–118, Amsterdam, 1973. North-Holland.

[Mar76] G. Markowsky. Chain-complete posets and directed sets with applications. *Algebra Universalis*, 6:53–68, 1976.

[Mar82] P. Martin-Löf. Constructive mathematics and computer programming. In *Sixth International Congress for Logic, Methodology, and Philosophy of Science*, pages 153–175, Amsterdam, 1982. North-Holland.

[Mar84] P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, Napoli, 1984.

[McC60] J. McCarthy. Recursive functions of symbolic expressions and their computation by machine. *Comm. Assoc. Comput. Mach.*, 3(3):184–195, 1960.

[McC61] J. McCarthy. A basis for mathematical theory of computation. In *Proc. Western Joint Computer Conf*, pages 225–238, May 1961. Later version in P. Braffort and D. Hirschberg (eds.), *Computer Programming and Formal Systems*, North-Holland, Amsterdam, pages 33–70, 1963.

[McC63] J. McCarthy. Towards a mathematical theory of computation. In *Proc. IFIP Congress 62*, pages 21–28, Amsterdam, 1963. North-Holland.

[McC78] J. McCarthy. History of LISP. *ACM Sigplan Notices*, 13(8):217–223, 1978.

[McC79] N. McCracken. *An Investigation of a Programming Language with a Polymorphic Type Structure*. PhD thesis, Syracuse Univ., 1979.

[McC84] N. McCracken. A finitary retract model for the polymorphic lambda calculus. Manuscript, 1984.

[Men64] E. Mendelson. *Introduction to Mathematical Logic*. D. Van Nostrand Co., Princeton, 1964.

[Mes89] J. Meseguer. Relating models of polymorphism. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 228–241, January 1989.

[Mey82] A.R. Meyer. What is a model of the lambda calculus ? *Information and Control*, 52(1):87–122, 1982.

[Mey88] B. Meyer. *Object-Oriented Software Construction*. Prentice-Hall, 1988.

[Mey92a] A.R. Meyer. MIT 6.830 course notes. Unpublished, 1992.

[Mey92b] B. Meyer. *Eiffel: The Language*. Prentice-Hall, 1992.

[MH88] J.C. Mitchell and R. Harper. The essence of ML. In *Proc. 15th ACM Symp. on Principles of Programming Languages*, pages 28–46, January 1988.

[Mil77] R. Milner. Fully abstract models of typed lambda calculi. *Theoretical Computer Science*, 4(1):1–22, 1977.

[Mil78] R. Milner. A theory of type polymorphism in programming. *JCSS*, 17:348–375, 1978.

[Mil85a] R. Milner. The Standard ML core language. *Polymorphism*, 2(2), 1985. 28 pages. An earlier version appeared in Proc. 1984 ACM Symp. on Lisp and Functional Programming.

[Mil85b] R. Milner. The Standard ML core language. *Polymorphism*, 2(2), 1985. 28 pages. An earlier version appeared in Proc. 1984 ACM Symp. on Lisp and Functional Programming.

[Mit86a] J.C. Mitchell. Abstract realizability for intuitionistic and relevant implication (abstract). *Journal Symbolic Logic*, 51(3):851–852, 1986.

[Mit86b] J.C. Mitchell. Representation independence and data abstraction. In *Proc. 13th ACM Symp. on Principles of Programming Languages*, pages 263–276, January 1986.

[Mit86c] J.C. Mitchell. A type-inference approach to reduction properties and semantics of polymorphic expressions. In *ACM Conference on LISP and Functional Programming*, pages 308–319, August 1986. Reprinted with minor revisions in *Logical Foundations of Functional Programming,* ed. G. Huet, Addison-Wesley (1990) 195–212.

[Mit88] J.C. Mitchell. Polymorphic type inference and containment. *Information and Computation*, 76(2/3):211–249, 1988. Reprinted in *Logical Foundations of Functional Programming,* ed. G. Huet, Addison-Wesley (1990) 153–194.

[Mit91a] J.C. Mitchell. On the equivalence of data representations. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 305–330. Academic Press, 1991.

[Mit91b] J.C. Mitchell. Type inference with simple subtypes. *J. Functional Programming*, 1(3):245–286, 1991.

[MM85] J.C. Mitchell and A.R. Meyer. Second-order logical relations. In *Logics of Programs*, pages 225–236, Berlin, June 1985. Springer-Verlag LNCS 193.

[MM91] J.C. Mitchell and E. Moggi. Kripke-style models for typed lambda calculus. *Ann. Pure and Applied Logic*, 51:99–124, 1991. Preliminary version in *Proc. IEEE Symp. on Logic in Computer Science,* 1987, pages 303–314.

[MMMS87] A. R. Meyer, J. C. Mitchell, E. Moggi, and R. Statman. Empty types in polymorphic lambda calculus. In *Proc. 14th ACM Symp. on Principles of Programming Languages*, pages 253–262, January 1987. Reprinted with minor revisions in *Logical Foundations of Functional Programming,* ed. G. Huet, Addison-Wesley (1990) 273–284.

[MO86] J.C. Mitchell and M.J. O'Donnell. Realizability semantics for error-tolerant logics. In *Theoretical Aspects of Reasoning About Knowledge*, pages 362–382. Morgan Kaufman, 1986.

[Mon76] J. D. Monk. *Mathematical Logic*, volume 37 of *Graduate Texts in Mathematics.* Springer-Verlag, Berlin, 1976.

[Mor73] J.H. Morris. Types are not sets. In 1^{st} *ACM Symp. on Principles of Programming Languages*, pages 120–124, 1973.

[Mor90] C. Morgan. *Programming from Specifications.* Prentice–Hall, 1990.

[Mos74] Y.N. Moschovakis. *Elementary induction on abstract structures.* north-Holland, Amsterdam, 1974.

[MP88] J.C. Mitchell and G.D. Plotkin. Abstract types have existential types. *ACM Trans. on Programming Languages and Systems*, 10(3):470–502, 1988. Preliminary version appeared in *Proc. 12th ACM Symp. on Principles of Programming Languages,* 1985.

[MPS86] D. MacQueen, G Plotkin, and R. Sethi. An ideal model for recursive polymorphic types. *Information and Control*, 71(1/2):95–130, 1986.

[MR77] M. Makkai and G.E. Reyes. *First-order categorical logic.* Springer LNM 611, Berlin, 1977.

[MR86] A.R. Meyer and M.B. Reinhold. Type is not a type. In *Proc. 13th ACM Symp. on Principles of Programming Languages*, pages 287–295, January 1986.

[MR92] Q. Ma and J.C. Reynolds. Types, abstraction, and parametric polymorphism, part 2. In *Mathematical Foundations of Programming Semantics, Proceedings 1991*, pages 1–40. Springer-Verlag LNCS 598, 1992.

[MS55] J.R. Myhill and J.C. Shepherdson. Effective operations on partial recursive functions. *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik*, 1, 1955.

[MS76] R.E. Milne and C. Strachey. *A theory of programming language semantics.* Chapman and Hall, London, and Wiley, New York, 1976.

[MS88] A.R. Meyer and K. Sieber. Towards fully abstract semantics for local variables: preliminary report. In *Proc. 15th ACM Symp. on Principles of Programming Languages*, pages 191–203, January 1988.

[MS89] J.C. Mitchell and P.J. Scott. Typed lambda calculus and cartesian closed categories. In *Categories in Computer Science and Logic, Proc. Summer Research Conference, Boulder, Colorado, June, 1987*, volume 92 of *Contemporary Mathematics*, pages 301–316. Amer. Math. Society, 1989.

[MS93] J.C. Mitchell and A. Scedrov. Notes on sconing and relators. In E. Boerger et al., editor, *Computer Science Logic '92, Selected Papers*, pages 352–378. Springer LNCS 702, 1993. Available by anonymous ftp from host ftp.cis.upenn.edu and the file pub/papers/scedrov/rel.dvi.

[MT91] Robin Milner and Mads Tofte. *Commentary on Standard ML.* MIT Press, 1991.

[MTH90] Robin Milner, Mads Tofte, and Robert Harper. *The Definition of Standard ML.* MIT Press, 1990.

[Mul87] K. Mulmuley. *Full abstraction and semantic equivalence.* The MIT Press, 1987.

[MV95] J.C. Mitchell and R. Viswanathan. Standard ml-nj weak polymorphism and imperative constructs. *Information and Computation*, ??:??, 1995. Submitted to special issue on *LICS 93.* A preliminary version appeared as M. Hoang, J.C. Mitchell and R. Viswanathan, Standard ML-NJ weak polymorphism and imperative constructs, in *Proc. IEEE Symp. on Logic in Computer Science,* 1993, 15–25.

[MW90] Z. Manna and R. Waldinger. *The logical basis for computer programming (Vol. 1 and 2).* Addison-Wesley, 1985,1990.

[Ned73] R.P. Nederpelt. *Strong Normalization in a typed lambda calculus with lambda structured types.* PhD thesis, Technological Univ. Eindhoven, 1973.

[Nel91] G. Nelson, editor. *Systems programming with Modula-3.* Prentice-Hall, 1991.

[New42] M.H.A. Newman. On theories with a combinatorial definition of 'equivalence'. *Ann. Math.*, 43(2):223–243, 1942.

[O'H90] P.W. O'Hearn. *The semantics of non-interference: a natural approach.* PhD thesis, Queen's University, Kingston, Canada, 1990.

[Ole82] F.J Oles. *A category-theoretic approach to the semantics of programming languages.* PhD thesis, Syracuse University, Syracuse, N.Y, 1982.

[Ole85] F.J Oles. Type algebras, functor categories and block structure. In *Algebraic Methods in Semantics*, pages 543–573. Cambridge Univ. Press, 1985.

[OT92] P. W. O'Hearn and R. D. Tennent. Semantics of local variables. In M. P. Fourman, P. T. Johnstone, and A. M. Pitts, editors, *Applications of Categories in Computer Science*, volume 177 of *London Mathematical Society Lecture Note Series*, pages 217–238. Cambridge University Press, Cambridge, England, 1992.

[OT93] P.W. O'Hearn and R.D. Tennent. Relational parametricity and local variables. In *Proc. 20th ACM Symp. Principles of Programming Languages*, pages 171–184, 1993.

[Par72] D. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 5(12):1053–1058, December 1972.

[Pau87] L.C. Paulson. *Logic and computation: interactive proof with Cambridge LCF.* Cambridge Univ. Press, 1987.

[Pey87] Simon L. Peyton Jones. *The Implementation of Functional Programming Languages.* Prentice–Hall, 1987.

[Pho90a] W. Phoa. *Domain theory in realizability toposes.* PhD thesis, Cambridge, 1990. Available as University of Edinburgh Dept. of Computer Science report CST-82-91 and ECS-LFCS-91-171.

[Pho90b] W. Phoa. Effective domains and intrinsic structure. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 366–377, 1990.

[Pie91] B. Pierce. *Basic category theory for computer scientists.* MIT Press, 1991.

[Pit87] A.M. Pitts. Polymorphism is set-theoretic, constructively. In *Proceedings Summer Conf. on Category Theory and Computer Science*, pages 12–39, Berlin, 1987. Springer LNCS 283.

[Pit89] A.M. Pitts. Non-trivial power types can't be subtypes of polymorphic types. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 6–13, 1989.

[Pit99] A. M. Pitts. Categorical logic. In *Handbook of Logic in Computer Science, Volume VI - Logical Methods in Computer Science.* Oxford University Press, 199? To appear.

[Plo73] G.D. Plotkin. Lambda-definability and logical relations. Technical Report (Memo.) SAI-RM-4, University of Edinburgh, School of Artificial Intelligence, 1973.

[Plo75] G.D. Plotkin. Call-by-name, call-by-value and the lambda calculus. *Theoretical Computer Science*, 1:125–159, 1975.

[Plo77] G.D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[Plo80] G.D. Plotkin. Lambda definability in the full type hierarchy. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 363–373. Academic Press, 1980.

[Plo81] G.D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Aarhus University Computer Science Department, 1981.

[Plo82] G.D. Plotkin. Notes on completeness of the full continuous hierarchy. Unpublished manuscript, 1982.

[Plo85] G.D. Plotkin. Denotational semantics with partial functions. Lecture notes, C.S.L.I. Summer School, Stanford, 1985.

[Pra65] D. Prawitz. *Natural Deduction.* Almquist and Wiksell, Stockholm, 1965.

[PW78] M.S. Paterson and M.N. Wegman. Linear unification. *JCSS*, 16:158–167, 1978.

[Rey74a] J.C. Reynolds. On the relation between direct and continuation semantics. In *Second Colloq. Automata, Languages and Programming*, pages 141–156, Berlin, 1974. Springer-Verlag LNCS.

[Rey74b] J.C. Reynolds. Towards a theory of type structure. In *Paris Colloq. on Programming*, pages 408–425, Berlin, 1974. Springer-Verlag LNCS 19.

[Rey81] J.C. Reynolds. The essence of Algol. In de Bakker and van Vliet, editors, *Algorithmic Languages*, pages 345–372, Amsterdam, 1981. IFIP, North-Holland.

[Rey83] J.C. Reynolds. Types, abstraction, and parametric polymorphism. In *Information Processing '83*, pages 513–523. North-Holland, Amsterdam, 1983.

[Rey84] J.C. Reynolds. Polymorphism is not set-theoretic. In *Proc. Int. Symp. on Semantics of Data Types, Sophia-Antipolis (France)*, pages 145–156, Berlin, 1984. Springer LNCS 173.

[Rie94] J.G. Riecke. Statman's 1-section theorem. *Information and Computation*, ?:to appear, 1994.

[Rob65] J.A. Robinson. A machine oriented logic based on the resolution principle. *JACM*, 12(1):23–41, 1965.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967.

[Ron] S. Ronchi Della Rocca. An unification semi-algorithm for intersection type schemes. Manuscript, ???

[Ros92] G. Rosolini. An exper model for quest. In S. Brookes, M. Main, A. Melton, M. Mislove, and D.Schmidt, editors, *Mathematical Foundations of Programming Semantics*, volume 598 of *Lecture Notes in Computer Science*, pages 436–445. Springer-Verlag, 1992.

[RP93] J.C. Reynolds and G.D. Plotkin. On functors expressible in the polymorphic lambda calculus. *Information and Computation*, 105:1–29, 1993. Reprinted in *Logical Foundations of Functional Programming,* ed. G. Huet, Addison-Wesley (1990) 127–152.

[Rus03] B. Russell. *The Principles of Mathematics.* Cambridge, 1903.

[Sal89] A. Salveson. *Polymorphism and Monomorphism in Martin-Löf's Type Theory.* PhD thesis, Institutt for Informatikk, University of Oslo, March 1989.

[Saz76] V.Y. Sazonov. Expressibility of functions in D. Scott's LCF language. *Algebra i Logika*, 15(3):308–330, 1976. Translation.

[Sce88] A. Scedrov. Kleene computable functionals and the higher-order existence property. *J. Pure Appl. Algebra*, 52:313–320, 1988.

[Sch86] D.A. Schmidt. *Denotational Semantics.* Allyn and Bacon, 1986.

[Sco69] D.S. Scott. A type–theoretic alternative to CUCH, ISWIM, OWHY. Manuscript, 1969.

[Sco76] D. Scott. Data types as lattices. *Siam J. Computing,* 5(3):522–587, 1976.

[Sco80a] D.S. Scott. Relating theories of the lambda calculus. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 403–450. Academic Press, 1980.

[Sco80b] D.S. Scott. A space of retracts. Manuscript, Merton College, Oxford., 1980.

[See84] R.A.G. Seely. Locally cartesian closed categories and type theory. *Math. Proc. Camb. Phil. Soc.*, 95:33–48, 1984.

[See87] R.A.G. Seely. Categorical semantics for higher-order polymorphic lambda calculus. *J. Symbolic Logic*, 52:969–989, 1987.

[Set89] R. Sethi. *Programming Languages: Concepts and Constructs.* Addison-Wesley, 1989.

[Sie92] Kurt Sieber. Reasoning about sequential functions via logical relations. In M. P. Fourman, P. T. Johnstone, and A. M. Pitts, editors, *Proc. LMS Symposium on Applications of Categories in Computer Science, Durham 1991*, volume 177 of *LMS Lecture Note Series*, pages 258–269. Cambridge University Press, 1992.

[SNGM89] G. Smolka, W. Nutt, J.A. Goguen, and J. Meseguer. Order-sorted equational computation. In *Resolution of equations in algebraic structures, Vol. 2,* pages 299–367. Academic Press, New York, 1989.

[Sny86] A. Snyder. Encapsulation and inheritance in object-oriented programming languages. In *Proc. ACM Symp. on Object-Oriented Programming Systems, Languages, and Applications*, pages 38–46, October 1986.

[SP82] M. Smyth and G.D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM J. Computing*, 11:761–783, 1982.

[Spi88] J.M. Spivey. *Understanding Z.* Cambridge University Press, 1988.

[SS71] D.S. Scott and C. Strachey. Toward a mathematical semantics for computer languages. In J. Fox, editor, *Proc. Symp. Computers and Automata.* Polytechnic Inst. of Brooklyn Press, 1971. Also Technical Monograph PRG-6, Programming Research Group, Oxford University.

[SS75] G.L. Steele and G.J. Sussman. Scheme: an interpreter for the extended lambda calculus. Technical Report 349, MIT Artificial Intelligence Laboratory, 1975.

[SS82] A. Scedrov and P.J. Scott. A note on the Friedman slash and Freyd covers. In *The L. E. J. Brouwer Centenary Symposium*, pages 443–452. North-Holland, Amsterdam, 1982.

[Sta79] R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9:67–72, 1979.

[Sta81] R. Statman. Number theoretic functions computable by polymorphic programs. In 22^{nd} *IEEE Symp. on Foundations of Computer Science*, pages 279–282, 1981.

[Sta82] R. Statman. Completeness, invariance and λ-definability. *J. Symbolic Logic*, 47(1):17–26, 1982.

[Sta85a] R. Statman. Equality between functionals, revisited. In *Harvey Friedman's Research on the Foundations of Mathematics*, pages 331–338. North-Holland, Amsterdam, 1985.

[Sta85b] R. Statman. Logical relations and the typed lambda calculus. *Information and Control*, 65:85–97, 1985.

[Ste72] S. Stenlund. *Combinators, λ-terms and Proof Theory.* Reidel, Dordrecht-Holland, 1972.

[Ste84] G.L. Steele. *Common Lisp: The language.* Digital Press, 1984.

[Sto77] J.E. Stoy. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory.* MIT Press, 1977.

[Sto88] A. Stoughton. *Fully Abstract Models of Programming Languages.* Pitman, London, and John Wiley and Sons, New York, 1988.

[Sto91a] A. Stoughton. Interdefinability of parallel operations in pcf. *Theoretical Computer Science*, 79:357–358, 1991.

[Sto91b] H. Stoyan. The influence of the designer on the design – john mccarthy and LISP. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 409–426. Academic Press, 1991.

[Str66] C. Strachey. Towards a formal semantics. In T.B. Steel, editor, *Formal Language Description Languages for Computer Programming*, pages 198–220. North-Holland, 1966.

[Str67] C. Strachey. Fundamental concepts in programming languages. Lecture Notes, International Summer School in Computer Programming, Copenhagen, August 1967.

[Str86] B. Stroustrop. *The C^{++} Programming Language.* Addison-Wesley, 1986.

[SW87] B. Shriver and P. Wegner, editors. *Research Directions in Object-oriented Programming.* MIT Press, 1987.

[Tai67] W.W. Tait. Intensional interpretation of functionals of finite type. *J. Symbolic Logic*, 32:198–212, 1967.

[Tay87] P. Taylor. *Recursive Domains, Indexed Category Theory and Polymorphism.* PhD thesis, Mathematics Dept., Cambridge University, 1987.

[Ten81] R.D. Tennent. *Principles of Programming Languages.* Prentice-Hall International, 1981.

[Ten86] R.D. Tennent. Functor-category semantics of programming languages and logics. In *Category Theory and Computer Science*, pages 206–224, Berlin, 1986. Springer LNCS 240.

[Ten91] R.D. Tennent. *Semantics of Programming Languages.* Prentice-Hall International, 1991.

[Tof88] M. Tofte. *Operational Semantics and Polymorphic Type Inference.* PhD thesis, Edinburgh University, 1988. Available as Edinburgh University Laboratory for Foundations of Computer Science Technical Report ECS–LFCS–88–54.

[Tof90] M. Tofte. Type inference for polymorphic references. *Information and Computation*, 89(1):1–34, 1990.

[Toy87] Y. Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *J. Assoc. Computing Machinery*, 34:128–143, 1987.

[Tro73] A.S. Troelstra. *Mathematical Investigation of Intuitionistic Arithmetic and Analysis.* Springer LNM 344, Berlin, 1973.

[Tur37] A.M. Turing. Computability and lambda definability. *J. Symbolic Logic*, 2:153–163, 1937.

[Tur85] D.A. Turner. Miranda: a non-strict functional language with polymorphic types. In *IFIP Int'l Conf. on Functional Programming and Computer Architecture, Nancy*, pages 1–16, Berlin, 1985. Springer LNCS 201.

[Tys88] J. Tyszkiewicz. Complexity of type inference in finitely typed lambda calculus. Master's thesis, University of Warsaw, 1988.

[Ull94] J.D. Ullman. *Elements of ML programming.* Prentice Hall, 1994.

[US 80] US Dept. of Defense. *Reference Manual for the Ada Programming Language.* GPO 008-000-00354-8, 1980.

[vD80] D.T. van Dalen. *The language theory of Automath.* PhD thesis, Technological Univ. Eindhoven, 1980.

[Wad76] C. Wadsworth. The relation between computational and denotational properties for Scott's D^∞ models. *Siam J. Comput.*, 5(3):488–521, 1976.

[Wad89] P. Wadler. Theorems for free! In *Proc. ACM Conf. Functional Programming and Computer Architecture*, pages 347–359, 1989.

[Wan79] M. Wand. Final algebra semantics and datatype extensions. *J. Comput. System Sci.*, 19:27–44, 1979.

[Wan87] M. Wand. A simple algorithm and proof for type inference. *Fundamenta Informaticae*, 10:115–122, 1987.

[Wel94] J. Wells. Typability and type checking in the second-order lambda-calculus are equivalent and undecidable. In *Proc. IEEE Symp. on Logic in Computer Science*, pages 176–185, 1994.

[Win93] G. Winskel. *The formal semantics of programming languages.* MIT Press, 1993.

[Wir90] M. Wirsing. Algebraic specification. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B*, pages 675–788. North-Holland, Amsterdam, 1990.

[WSH77] J. Welsh, W. Sneeringer, and C.A.R Hoare. Ambiguities and insecurities in pascal. *Software – Practice and Experience*, 7(6):685–696, 1977.

[YT87] A. Yonezawa and M. Tokoro, editors. *Object-oriented Concurrent Programming.* MIT Press, 1987.

Index