Literaturverzeichnis

[Ackermann, 1954] W. Ackermann. *Solvable cases of the decision problem*. North–Holland, Amsterdam, 1954.

[Aczel, 1978] Peter Aczel. The type theoretic interpretation of constructive set theory. In A. MacIntyre, L. Pacholski, and J. Paris, editors, *Logic Colloquium '77*, pages 55–66. North–Holland, Amsterdam, 1978.

[Allen *et.al.*, 1990] Stuart F. Allen, Robert L. Constable, Douglas J. Howe, and William E. Aitken. The semantics of reflected proof. In John C. Mitchell, editor, *LICS-90 — Proceedings of the Fifth Annual Symposium on Logic in Computer Science, 1990, Philadelphia, PA, June 1990*, pages 95–106. IEEE Computer Society Press, 1990.

[Allen, 1987a] Stuart Allen. A non-type-theoretic definition of Martin-Löf's types. In David Gries, editor, *LICS-87 — Proceedings of the Second Annual Symposium on Logic in Computer Science, Ithaca, New York, USA, June 1987*, pages 215–224. IEEE Computer Society Press, 1987.

[Allen, 1987b] Stuart Allen. *A non-type-theoretic semantics for type-theoretic language*. PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, December 1987. TR 87-866.

[Altenkirch *et.al.*, 1994] Thorsten Altenkirch, Veronica Gaspes, Bengt Nordström, and Björn von Sydow. *A user's guide to ALF*. University of Göteborg, 1994.

[Amadio & Cardelli, 1990] Roberto M. Amadio and Luca Cardelli. Subtyping recursive types. Technical Report 62, DIGITAL Systems Research Center, 130 Lytton Avenue, Palo Alto, CA 94301, August 1990.

[Anderson, 1970] R. Anderson. Completeness results for E-resolution. In *Proceedings of the Spring Joint Computer Conference*, volume 36, pages 636–656. AFIPS Press, 1970.

[Andrews, 1971] Peter B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, 36(3):414–432, September 1971.

[Andrews, 1986] Peter B. Andrews. *An Introduction to mathematical logic and Type Theory: To Truth through Proof*. Academic Press, London, Orlando, 1986.

[Backhouse *et.al.*, 1988a] Roland C. Backhouse, P. Chisholm, and Grant Malcolm. Do it yourself type theory (part I). *EATCS Bulletin*, 34:68–110, 1988. Also in *Formal aspects of computing* 1:19-84, 1989.

[Backhouse *et.al.*, 1988b] Roland C. Backhouse, P. Chisholm, and Grant Malcolm. Do it yourself type theory (part II). *EATCS Bulletin*, 35:205–245, 1988.

[Backhouse, 1984] Roland C. Backhouse. A note on subtypes in Martin-Löf's Theory of Types. Technical Report CSM-70, Computer Science Department, University of Essex, England, November 1984.

[Backhouse, 1985] Roland C. Backhouse. Algorithm development in Martin-Löf's Type Theory. Technical report, Computer Science Department, University of Essex, England, 1985.

[Backhouse, 1989] Roland C. Backhouse. Constructive Type Theory - an introduction. In Manfred Broy, editor, *Constructive Methods in Computer Science*, volume 55 of *NATO ASI Series, Series F: Computer & System Sciences*, pages 9–62. Springer Verlag, Berlin, Heidelberg, New-York, 1989.

[Balzer *et.al.*, 1983] Robert Balzer, T. E. Cheatham jr, and Cordell Green. Software technology in the 1990's: Using a new paradigm. *Computer*, pages 39–45, November 1983.

[Balzer, 1985] Robert Balzer. A 15 year perspective on automatic programming. *IEEE Transactions on Software Engineering*, SE-11(11):1257–1268, November 1985.

[Barendregt, 1981] Henk P. Barendregt. *The Lambda Calculus. Its syntax and semantics*, volume 103 of *Studies in Logic and The Foundations of Mathematics*. North–Holland, Amsterdam, 1981. (revised version 1984).

[Barstow, 1979] David R. Barstow. *Knowledge-Based Program Construction*. Elsevier Science Publishers B.V. (North–Holland) Amsterdam, New York, 1979.

[Basin, 1989] David A. Basin. Building theories in NuPRL. In A. R. Meyer and M. A. Taitslin, editors, *Logic at Botik 89*, number 363 in Lecture Notes in Computer Science, pages 12–25. Springer Verlag, Berlin, Heidelberg, New-York, 1989.

[Bates & Constable, 1985] Joseph L. Bates and Robert L. Constable. Proofs as programs. *ACM Transactions on Programming Languages and Systems*, 7(1):113–136, January 1985.

[Bates, 1981] Joseph L. Bates. *A logic for correct Program Development*. PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, 1981.

[Bauer & others, 1988] F. L. Bauer *et al.* *The Munich Project CIP: Volume II*, volume 292 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Heidelberg, New-York, 1988.

[Bibel & Hörnig, 1984] Wolfgang Bibel and K. M. Hörnig. LOPS - a system based on a strategical approach to program synthesis. In Alan W. Biermann, G. Guiho, and Y. Kodratoff, editors, *Automatic program construction techniques*, chapter 3, pages 69–89. MacMillan, New York, 1984.

[Bibel *et.al.*, 1978] Wolfgang Bibel, Ulrich Furbach, and J. F. Schreiber. Strategies for the synthesis of algorithms. In K. Alber, editor, *Proceedings 5th GI Conference on Programming Languages*, number 12 in Informatik Fachberichte, pages 97–109, Berlin, 1978. Springer Verlag, Berlin, Heidelberg, New-York.

[Bibel *et.al.*, 1994] W. Bibel, S. Brüning, U. Egly, and T. Rath. Komet. In Alan Bundy, editor, *Proceedings of the 12^{th} Conference on Automated Deduction, 1994*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 783–787. Springer Verlag, Berlin, Heidelberg, New-York, 1994. System description.

[Bibel, 1978] Wolfgang Bibel. On strategies for the synthesis of algorithms. In D. Sleeman, editor, *Proceedings of AISB/GI Conference on Artificial Intelligence, Leeds University, England*, pages 22–27, 1978.

[Bibel, 1980] Wolfgang Bibel. Syntax–directed, semantics–supported program synthesis. *Artificial Intelligence*, 14(3):243–261, October 1980.

[Bibel, 1983] Wolfgang Bibel. Matings in matrices. *Communications of the Association for Computing Machinery*, 26(11):844–852, November 1983.

[Bibel, 1987] Wolfgang Bibel. *Automated Theorem Proving*. Vieweg Verlag, Braunschweig, second edition, 1987.

[Bibel, 1991] Wolfgang Bibel. Toward predicative programming. In Michael R. Lowry and Robert McCartney, editors, *Automating Software Design*, pages 405–424, Menlo Park, CA, 1991. AAAI Press / The MIT Press.

[Bibel, 1992] Wolfgang Bibel. *Deduktion – Automatisierung der Logik*, volume 6.2 of *Handbuch der Informatik*. R. Oldenbourg, München, Wien, 1992.

[Bishop & Bridges, 1985] Errett Bishop and Douglas Bridges. *Constructive Analysis*. Springer Verlag, Berlin, Heidelberg, New-York, 1985.

[Bishop, 1967] Errett Bishop. *Foundations of Constructive Analysis.* McGraw-Hill, New York, 1967. revised and extended version: [Bishop & Bridges, 1985].

[Bjorner *et.al.*, 1988] D. Bjorner, A.P. Ershov, and N.D.Jones, editors. *Partial evaluation and mixed computation.* North–Holland, Amsterdam, 1988.

[Blaine & Goldberg, 1990a] Lee Blaine and Allen Goldberg. DTRE—a semi-automatic transformation system. Technical report, Kestrel Institute, October 1990.

[Blaine & Goldberg, 1990b] Lee Blaine and Allen Goldberg. Verifiably correct data type refinement. Technical report, Kestrel Institute, November 1990.

[Bläsius *et.al.*, 1981] K. Bläsius, N. Eisinger, J. Siekmann, G. Smolka, A. Herold, and C. Walther. The Markgraf Karl refutation procedure. In *IJCAI-81 — Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, August 1981*, pages 511–518, Los Altos CA, 1981. Morgan Kaufmann.

[Bledsoe, 1977] W. Bledsoe. Non-resolution theorem proving. *Artificial Intelligence*, 9(1):1–35, 1977.

[Boyer & Moore, 1979] Robert S. Boyer and J. Strother Moore. *A computational Logic.* Academic Press, London, New York, 1979.

[Bridges, 1979] Douglas Bridges. *Constructive Functional Analysis.* Pitman, 1979.

[Brouwer, 1908a] L. E. J. Brouwer. De onbetrouwbaarheid der logische principes. *Tijdschrift voor wijsbegeerte*, 2:152–158, 1908. English translation: “The unreliability of the logical principles”, in [Brouwer, 1975], pp. 107–112.

[Brouwer, 1908b] L. E. J. Brouwer. Over de grondslagen der wiskunde. *Nieuw Arch. Wisk*, 2(8):326–328, 1908. English translation: “On the foundations of mathematics”, in [Brouwer, 1975], pp. 105–107.

[Brouwer, 1924a] L. E. J. Brouwer. Intuitionistische Zerlegung mathematischer Grundbegriffe. *Jahresbericht Deutscher Mathematiker Verein*, 33:251–256, 1924.

[Brouwer, 1924b] L. E. J. Brouwer. Über die Bedeutung des Satzes vom ausgeschlossenen Dritten in der Mathematik, insbesondere in der Funktionentheorie. *J. reine und angewandte Mathematik*, pages 1–7, 1924.

[Brouwer, 1925] L. E. J. Brouwer. Zur Begründung der intuitionistischen Mathematik I. *Mathematische Annalen*, 93:244–257, 1925.

[Brouwer, 1926] L. E. J. Brouwer. Zur Begründung der intuitionistischen Mathematik II. *Mathematische Annalen*, 95:453–472, 1926.

[Brouwer, 1927] L. E. J. Brouwer. Zur Begründung der intuitionistischen Mathematik III. *Mathematische Annalen*, 96:451–488, 1927.

[Brouwer, 1975] L. E. J. Brouwer. *Collected Works*, volume 1. North–Holland, Amsterdam, 1975.

[Broy & Pepper, 1981] Manfred Broy and Peter Pepper. Program development as a formal activity. *IEEE Transactions on Software Engineering*, SE-7(1):14–22, January 1981.

[Bruijn, 1980] N. G. De Bruijn. A survey of the project AUTOMATH. In J.P. Seldin and J.R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 579–606. Academic Press, London, New York, 1980.

[Bundy *et.al.*, 1990] Alan Bundy, Frank van Harmelen, Christian Horn, and Alan Smaill. The Oyster-Clam system. In M. E. Stickel, editor, *Proceedings of the 10^{th} Conference on Automated Deduction, Kaiserslautern, Germany, July 1990*, volume 449 of *Lecture Notes in Computer Science*, pages 647–648. Springer Verlag, Berlin, Heidelberg, New-York, 1990.

[Bundy, 1989] Alan Bundy. Automatic guidance of program synthesis proofs. In *Proceedings of the Workshop on Automating Software Design, IJCAI-89, Kestrel Institute, Palo Alto*, pages 57–59, 1989.

[Burstall & Darlington, 1977] R. M. Burstall and J. Darlington. A transformation system for developing recursive programs. *Journal of the Association for Computing Machinery*, 24(1):44–67, January 1977.

[Cardone & Coppo, 1990] F. Cardone and M. Coppo. Two extendsions of Curry's type inference system. In Piergiorgio Odifreddi, editor, *Logic and Computer Science*, pages 19–76. Academic Press, London, 1990.

[Chisholm, 1985] Paul Chisholm. Derivation of a parsing algorithm in Martin-Löf's theory of types. Technical report, Department of Computer Science, Heriot-Watt University, Edinburgh, Scotland, 1985.

[Chisholm, 1987] Paul Chisholm. Derivation of a parsing algorithm in Martin-Löf's Type Theory. *Science of Computer Programming*, 8:1–42, 1987.

[Chou & Gao, 1990] Shang-Ching Chou and Xiao-Shan Gao. Ritt-Wu's decomposition algorithm and geometry theorem proving. In M. E. Stickel, editor, *Proceedings of the 10^{th} Conference on Automated Deduction, Kaiserslautern, Germany, July 1990*, volume 449 of *Lecture Notes in Computer Science*, pages 207–220. Springer Verlag, Berlin, Heidelberg, New-York, 1990.

[Church & Rosser, 1936] Alonzo Church and B. J. Rosser. Some properties of conversion. *Trans. Am. Math. Soc.*, 39:472–482, 1936.

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

[Clark & Sickel, 1977] Keith L. Clark and S. Sickel. Predicate logic: A calculus for the formal derivation of programs. In Raj Reddy, editor, *IJCAI-77 — Proceedings of the 5^{th} International Joint Conference on Artificial Intelligence, Cambridge, MA, August 1977*, pages 419–420. Morgan Kaufman, Los Altos, CA, 1977.

[Cleaveland, 1987] Walter Rance Cleaveland. *Type-theoretic models of concurrency.* PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, 1987. TR 87-837.

[Constable & Howe, 1990a] Robert L. Constable and Douglas J. Howe. Implementing metamathematics as an approach to automatic theorem proving. In R. B. Banerjee, editor, *Formal techniques in Artificial Intelligence, a sourcebook*, pages 45–75. Elsevier Science Publishers B.V. (North–Holland) Amsterdam, 1990.

[Constable & Howe, 1990b] Robert L. Constable and Douglas J. Howe. Nuprl as a general logic. In Piergiorgio Odifreddi, editor, *Logic and Computer Science*, pages 77–90. Academic Press, London, 1990.

[Constable & Mendler, 1985] Robert L. Constable and Nax Paul Mendler. Recursive definitions in Type Theory. In Rohit Parikh, editor, *Proceedings of the Logics of Programs Conference, Brooklyn, NY, USA, June 1985*, volume 193 of *Lecture Notes in Computer Science*, pages 61–78. Springer Verlag, Berlin, Heidelberg, New-York, 1985.

[Constable & Smith, 1987] Robert L. Constable and Scott Fraser Smith. Partial objects in constructive Type Theory. In David Gries, editor, *LICS-87 — Proceedings of the Second Annual Symposium on Logic in Computer Science, Ithaca, New York, USA, June 1987*, pages 183–193. IEEE Computer Society Press, 1987.

[Constable & Smith, 1988] Robert L. Constable and Scott Fraser Smith. Computational foundations of basic recursive function theory. In Yuri Gurevich, editor, *LICS-88 — Proceedings of the Third Annual Symposium on Logic in Computer Science, Edinburgh, Scotland, July 1988*, pages 360–371. IEEE Computer Society Press, 1988.

[Constable & Smith, 1993] Robert L. Constable and Scott Fraser Smith. Computational foundations of basic recursive function theory. *Theoretical Computer Science*, 121:89–112, 1993.

[Constable *et.al.*, 1982] Robert L. Constable, Scott D. Johnson, and Carl D. Eichenlaub. *Introduction to the PL/CV2 Programming Logic*, volume 135 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Heidelberg, New-York, 1982.

[Constable *et.al.*, 1985] Robert L. Constable, Todd B. Knoblock, and Joseph L. Bates. Writing programs that construct proofs. *Journal of Automated Reasoning*, 1:285–326, 1985.

[Constable *et.al.*, 1986] Robert L. Constable, Stuart F. Allen, H. Mark Bromley, W. Rance Cleaveland, J. F. Cremer, Robert W. Harper, Douglas J. Howe, Todd B. Knoblock, Nax Paul Mendler, Prakash Panangaden, Jim T. Sasaki, and Scott F. Smith. *Implementing Mathematics with the NuPRL proof development system.* Prentice Hall, Englewood Cliffs, NJ, 1986.

[Constable, 1983] Robert L. Constable. Programs as proofs. *Information Processing Letters*, 16(3):105–112, 1983.

[Constable, 1984] Robert L. Constable. Mathematics as programming. In Edmund Clarke and Dexter Kozen, editors, *Logics of Programs*, number 164 in Lecture Notes in Computer Science, pages 116–128. Springer Verlag, Berlin, Heidelberg, New-York, 1984.

[Constable, 1985] Robert L. Constable. Constructive mathematics as a programming logic I: Some principles of theory. *Annals of Discrete Mathematics*, 24:21–38, 1985.

[Constable, 1988] Robert L. Constable. Themes in the development of programming logics circa 1963-1987. *Annual Reviews in Computer Science*, pages 147–165, 1988.

[Constable, 1989] Robert L. Constable. Assigning meaning to proofs: a semantic basis for problem solving environments. In Manfred Broy, editor, *Constructive Methods in Computer Science*, volume 55 of *NATO ASI Series, Series F: Computer & System Sciences*, pages 63–94. Springer Verlag, Berlin, Heidelberg, New-York, 1989.

[Coquand & Huet, 1985] Thierry Coquand and Gerard Huet. Constructions: A higher order proof system for mechanizing mathematics. In *EUROCAL'85*, pages 151–184, Linz, Austria, April 1985.

[Coquand & Huet, 1988] Thierry Coquand and Gerard Huet. The calculus of constructions. *Information and Computation*, 76:95–120, 1988.

[Coquand & Paulin, 1988] Thierry Coquand and Christine Paulin. Inductively defined types. In P. Martin-Löf and G. Mints, editors, *COLOG-88: Proceedings of the International Conference on Computer Logic, Tallin, USSR, December 1988*, number 417 in Lecture Notes in Computer Science, pages 50–66. Springer Verlag, Berlin, Heidelberg, New-York, 1988.

[Coquand, 1990] Thierry Coquand. Metamathematical investigations of a calculus of constructions. In Piergiorgio Odifreddi, editor, *Logic and Computer Science*, pages 91–122. Academic Press, London, 1990.

[Costas, 1984] J. Costas. A study of a class of detection waveforms having nearly ideal range – doppler amibiguity properties. In *Proceedings of the IEEE*, volume 72, pages 996–1009, 1984.

[Cousineau & Huet, 1990] Guy Cousineau and Gerard Huet. The CAML primer. Rapports Techniques 122, Institut National de Recherche en Informatique et en Automatique, September 1990.

[Curry *et.al.*, 1958] Haskell B. Curry, R. Feys, and W. Craig. *Combinatory Logic*, volume 1. North–Holland, Amsterdam, 1958.

[Curry, 1970] Haskell B. Curry. *Outline of a formalist philosophy of mathematics.* North–Holland, Amsterdam, 1970.

[Dahl *et.al.*, 1972] O.-J. Dahl, Edsger W. Dijkstra, and C.A.R. Hoare. *Structured Programming*, volume 8 of *A.P.I.C. Studies in Data Processing.* Academic Press, London, 1972.

[Damas & Milner, 1982] Luis Damas and Robin Milner. Principal type schemes for functional programs. In *Proceedings of the 9th ACM Symposium on Principles of Programming Languages*, pages 207–212, 1982.

[Darlington, 1975] John Darlington. Application of program transformation to program synthesis. In *IRIA Symposium on Proving and Improving programs*, pages 133–144, Arc-et-Senans, France, 1975.

[Davis & Putnam, 1960] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the Association for Computing Machinery*, 7:201–215, 1960.

[de la Tour & Kreitz, 1992] Thierry Boy de la Tour and Christoph Kreitz. Building proofs by analogy via the Curry-Howard isomorphism. In A. Voronkov, editor, *LPAR'92, Proceedings of the Conference on Logic Programming and Automated Reasoning, St.Petersburgh, Russia, July 1992*, number 624 in Lecture Notes in Computer Science, pages 202–213. Springer Verlag, Berlin, Heidelberg, New-York, 1992.

[Dershowitz, 1985] Nachum Dershowitz. Synthesis by completion. In A. Joshi, editor, *IJCAI-85 — Proceedings of the 9th International Joint Conference on Artificial Intelligence, Los Angeles, August 1985*, pages 208–214, 1985.

[Dijkstra, 1976] Edsger W. Dijkstra. *A discipline of Programming.* Prentice Hall, Englewood Cliffs, NJ, 1976.

[Dummett, 1977] Michael Dummett. *Elements of Intuitionism.* Oxford Logic Series. Clarenden Press, Oxford, 1977.

[Felty & Miller, 1988] Amy Felty and Dale Miller. Specifying theorem provers in a higher-order logic programming language. In E. Lusk and R. Overbeek, editors, *Proceedings of the 9th Conference on Automated Deduction, Argonne, Illinois, May 1988*, volume 310 of *Lecture Notes in Computer Science*, pages 61–80. Springer Verlag, Berlin, Heidelberg, New-York, 1988.

[Felty & Miller, 1990] Amy Felty and Dale Miller. Encoding a dependent-type λ-calculus in a logic programming language. In M. E. Stickel, editor, *Proceedings of the 10th Conference on Automated Deduction, Kaiserslautern, Germany, July 1990*, volume 449 of *Lecture Notes in Computer Science*, pages 221–235. Springer Verlag, Berlin, Heidelberg, New-York, 1990.

[Felty, 1991] Amy Felty. Encoding dependent types in an intuitionistic logic. In Gérard Huet and Gordon Plotkin, editors, *Logical Frameworks*, pages 215–251. Cambridge University Press, 1991.

[Franova & Kodratoff, 1991] Marta Franova and Yves Kodratoff. Solving "How to clear a block" with constructive matching methodology. In John Mylopoulos and Ray Reiter, editors, *IJCAI-91 — Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sidney, August 1991*, pages 232–237. Morgan Kaufmann Publishers, 1991.

[Franova, 1985] Marta Franova. A methodology for automatic programming based on the constructive matching strategy. In *EUROCAL 85*, pages 568–570. Springer Verlag, Berlin, Heidelberg, New-York, 1985.

[Frege, 1879] Gottlob Frege. *Begriffsschrift.* Verlag Louis Nebert, Halle, 1879.

[Frege, 1892] Gottlob Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892.

[Freyd, 1990] Peter Freyd. Recursive types reduced to inductive types. In John C. Mitchell, editor, *LICS-90 — Proceedings of the Fifth Annual Symposium on Logic in Computer Science, 1990, Philadelphia, PA, June 1990*, pages 498–507. IEEE Computer Society Press, 1990.

[Fribourg, 1990] L. Fribourg. Extracting logic programs from proofs that use extended PROLOG execution and induction. In *Proceedings of the 7th International Conference on Logic Programming, Jerusalem*, pages 685–699. MIT Press, 1990.

[Gallier, 1986] Jean H. Gallier. *Logic for Computer Science: Foundations of Automated Theorem Proving.* Harper and Row, New York, 1986.

[Galmiche, 1990] Didier Galmiche. Constructive system for automatic program synthesis. *Theoretical Computer Science*, 71:227–239, 1990.

[Gentzen, 1935] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English Translation: "Investigations into logical deduction" in [Szabo, 1969] pages 68-131.

[Girard *et.al.*, 1989] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[Girard, 1971] Jean-Yves Girard. Une extension de l'interpretation de godel a l'analyse, et son application a l'elimination des coupures dans l'analyse at la theorie des types. In J. Fenstad, editor, *Proceedings of the second Scandinavian Logic Symposium*, pages 63–92. North–Holland, Amsterdam, 1971.

[Girard, 1972] Jean-Yves Girard. *Interpretation fonctionelle et elimination des coupures dans l'arithmetique d'ordre superieur.* PhD thesis, Universite Paris VII, 1972.

[Girard, 1986] Jean-Yves Girard. The system F of variable types: Fifteen years later. *Theoretical Computer Science*, 45:159–192, 1986.

[Girard, 1987] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Giunchiglia & Smaill, 1989] Fausto Giunchiglia and Alan Smaill. Reflection in constructive and non-constructive automated reasoning. In Harvey Abramson and M. H. Rogers, editors, *Meta-Programming in Logic Programming*, chapter 6, pages 123–140. MIT Press, Cambridge Mass., 1989.

[Gordon *et.al.*, 1979] Michael J. Gordon, Robin Milner, and Christopher P. Wadsworth. *Edinburgh LCF: A mechanized Logic of Computation.* Number 78 in Lecture Notes in Computer Science. Springer Verlag, Berlin, Heidelberg, New-York, 1979.

[Gordon., 1985] M. Gordon. HOL: A machine oriented formalization of higher order logic. Technical Report 68, Cambridge University, 1985.

[Gordon., 1987] M. Gordon. HOL: A proof generating system for higher order logic. Technical Report 103, Cambridge University, 1987.

[Green, 1969] Cordell C. Green. An application of theorem proving to problem solving. In *IJCAI-69 — Proceedings of the 1st International Joint Conference on Artificial Intelligence, Washington, DC, May 1969*, pages 219–239, 1969.

[Gries, 1981] David Gries. *The science of programming.* Springer Verlag, Berlin, Heidelberg, New-York, 1981.

[Guttag & Horning, 1978] J. V. Guttag and J. J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, pages 27–52, 1978.

[Harper & Pollack, 1989] Robert Harper and Robert Pollack. Type checking, universe polymorphism and typical ambiguity in the calculus of constructions. In J. Diaz and F. Orejas, editors, *TAPSOFT '89 - Proceedings of the International Joint Conference on Theory and Practice of Software Development, Barcelona, Spain, March 1989*, number 352 in Lecture Notes in Computer Science, pages 241–256. Springer Verlag, Berlin, Heidelberg, New-York, 1989.

[Harper, 1985] Robert W. Harper. *Aspects of the implementation of Type Theory.* PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, April 1985. TR 85-675.

[Hayashi, 1986] S. Hayashi. PX: a system extracting programs from proofs. In *Proceedings of the IFIP Confereence on Formal Description of Programming Concepts*, pages 399–424, 1986.

[Hayes, 1987] I. Hayes. *Specification Case Studies.* Prentice-Hall, 1987.

[Heisel *et.al.*, 1988] M. Heisel, W. Reif, and W. Stephan. Implementing verification strategies in the KIV system. In E. Lusk and R. Overbeek, editors, *Proceedings of the 9^{th} Conference on Automated Deduction, Argonne, Illinois, May 1988*, number 310 in Lecture Notes in Computer Science, pages 131–140. Springer Verlag, Berlin, Heidelberg, New-York, 1988.

[Heisel *et.al.*, 1990] M. Heisel, W. Reif, and W. Stephan. Tactical theorem proving in program verification. In M. E. Stickel, editor, *Proceedings of the 10^{th} Conference on Automated Deduction, Kaiserslautern, Germany, July 1990*, number 449 in Lecture Notes in Computer Science, pages 117–131. Springer Verlag, Berlin, Heidelberg, New-York, 1990.

[Heisel *et.al.*, 1991] Maritta Heisel, Wolfgang Reif, and Werner Stephan. Formal software development in the KIV system. In Michael R. Lowry and Robert D. McCartney, editors, *Automating Software Design*, pages 547–576, Menlo Park, CA, 1991. AAAI Press / The MIT Press.

[Heisel, 1989] Maritta Heisel. A formalization and implementation of Gries's program development method with the KIV environment. Technical Report 3/89, Fakultät für Informatik, Universität Karlsruhe, 1989.

[Heyting, 1931] Arend Heyting. Die intuitionistische Grundlegung der Mathematik. *Erkenntnis*, 2:106–115, 1931.

[Heyting, 1934] A. Heyting. *Mathematische Grundlagenforschung: Intuitionismus – Beweistheorie.* Springer Verlag, Berlin, Heidelberg, New-York, 1934.

[Heyting, 1971] Arend Heyting. *Intuitionism: An Introduction.* North–Holland, Amsterdam, 3^{rd} edition, 1971.

[Hilbert & Bernays, 1934] David Hilbert and P. Bernays. *Grundlagen der Mathematik*, volume 1. Springer Verlag, Berlin, Heidelberg, New-York, 1934.

[Hilbert & Bernays, 1939] David Hilbert and P. Bernays. *Grundlagen der Mathematik*, volume 2. Springer Verlag, Berlin, Heidelberg, New-York, 1939.

[Hindley & Seldin, 1986] J. Roger Hindley and Jonathan P. Seldin. *Introduction to combinators and λ-calculus.* Cambridge University Press, 1986.

[Hindley, 1969] J. Roger Hindley. The principal type scheme of an object in combinatory logic. *Transactions of the American Mathematical Society*, 146:29–60, 1969.

[Hindley, 1983] J. Roger Hindley. The completeness theorem for typing λ-terms. *Theoretical Computer Science*, 22:1–17, 1983.

[Hoare, 1972] Charles Antony Richard Hoare. Notes on data structuring. In *[Dahl* et.al.*, 1972]*, pages 83–174. Academic Press, London, 1972.

[Hoare, 1975] C. A. R. Hoare. Recursive data structures. *International Journal of Computer and Information Sciences*, 4(2):105–132, June 1975.

[Hogger, 1978] C. J. Hogger. Goal-oriented derivation of logic programs. In *7th Symposium on the Mathematical foundations of Computer Science*, Zakopane, Poland, September 3–8 1978. Polish Academy of Science.

[Hogger, 1981] C. J. Hogger. Derivation of logic programs. *Journal of the Association for Computing Machinery*, 28(2):372–392, April 1981.

[Horn, 1988] Christian Horn. Interactive program and proof development with NurPRL. Seminarbericht 97, Sektion Mathematik der Humboldt-Universität zu Berlin, PSF 1297, 1086 Berlin, 1988.

[Howard, 1980] W. Howard. The formulas-as-types notion of construction. In J. Roger Hindley and Jonathan P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, London, New York, 1980. First published 1969.

[Howe, 1986] Douglas J. Howe. Implementing number theory: An experiment with NuPRL. In J. H. Siekmann, editor, *Proceedings of the 8^{th} Conference on Automated Deduction, Oxford, England, 1986*, volume 230 of *Lecture Notes in Computer Science*, pages 404–415. Springer Verlag, Berlin, Heidelberg, New-York, 1986.

[Howe, 1987] Douglas J. Howe. The computational behaviour of girard's paradox. In David Gries, editor, *LICS-87 — Proceedings of the Second Annual Symposium on Logic in Computer Science, Ithaca, New York, USA, June 1987*, pages 205–214. IEEE Computer Society Press, 1987.

[Howe, 1988] Douglas J. Howe. *Automating reasoning in an implementation of constructive Type Theory.* PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, June 1988. TR 88-925.

[Huet, 1986] Gérard Huet. Deduction and computation. In Wolfgang Bibel and Ph. Jorrand, editors, *Fundamentals of Artificial Intelligence — An Advanced Course*, volume 232 of *Lecture Notes in Computer Science*, pages 39–74. Springer Verlag, Berlin, Heidelberg, New-York, 1986.

[Jackson, 1993a] Paul Jackson. *NuPRL ML Manual.* Cornell University. Department of Computer Science, 1993.

[Jackson, 1993b] Paul Jackson. *The Nuprl Proof Development System, Version 4.1: Introductory Tutorial.* Cornell University. Department of Computer Science, 1993.

[Jackson, 1993c] Paul Jackson. *The Nuprl Proof Development System, Version 4.1: Reference Manual and User's Guide.* Cornell University. Department of Computer Science, 1993.

[Johnson & Feather, 1991] W. Lewis Johnson and Martin S. Feather. Using evolution transformations to construct specifications. In Michael R. Lowry and Robert D. McCartney, editors, *Automating Software Design*, pages 65–92, Menlo Park, CA, 1991. AAAI Press / The MIT Press.

[Johnson, 1983] Scott D. Johnson. *A Computer System for checking Proofs.* UMI Press, New York, 1983. Ph.D. Thesis.

[Jones & Shaw, 1990] C. B. Jones and R. C. Shaw. *Case Studies in Systematic Software Development.* Prentice-Hall, 1990.

[Kant & Barstow, 1981] Elaine Kant and David Barstow. The refinement paradigm: the interaction of coding and efficiency knowledge in program synthesis. *IEEE Transactions on Software Engineering*, 7(5):458–471, 1981.

[Kelly & Nonnenmann, 1991] Van E. Kelly and Uwe Nonnenmann. Reducing the complexity of formal specification acquisition. In Michael R. Lowry and Robert D. McCartney, editors, *Automating Software Design*, pages 41–64, Menlo Park, CA, 1991. AAAI Press / The MIT Press.

[Kneale & Kneale, 1962] William Kneale and Martha Kneale. *The Development of Logic*. Clarendon Press, Oxford, 1962.

[Knoblock, 1987] Todd B. Knoblock. *Metamathematical extensibility in Type Theory*. PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, December 1987. TR 87-892.

[Knuth, 1968] D. E. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison Wesley, Reading, MA, 1968.

[Knuth, 1972] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison Wesley, Reading, MA, 1972.

[Knuth, 1975] D. E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison Wesley, Reading, MA, 1975.

[Kolmogorov, 1932] A. N. Kolmogorov. Zur Deutung der intuitionistischen Logik. *Mathematische Zeitschrift*, 35:58–65, 1932.

[Kreitz, 1986] Christoph Kreitz. Constructive automata theory implemented with the NuPRL proof development system. Technical Report TR 86-779, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, August 1986.

[Kreitz, 1992] Christoph Kreitz. *METASYNTHESIS: Deriving Programs that Develop Programs*. Thesis for Habilitation, Technische Hochschule Darmstadt, FG Intellektik, 1992. Forschungsbericht AIDA–93–03.

[Krieg-Brückner *et.al.*, 1986] B. Krieg-Brückner, B. Hoffmann, H. Ganzinger, M. Broy, R. Wilhelm, U. Möricke, B. Weisgerber, A.D. McGettrick, I.G. Campbell, and G. Winterstein. PROgram development by SPECification and TRAnsformation. In *Proceedings ESPRIT Conference*, Brussels, 1986.

[Krieg-Brückner, 1989] B. Krieg-Brückner. Algebraic specifications and functionals for transformational program and meta-program development. In J. Diaz and F. Orejas, editors, *TAPSOFT '89 - Proceedings of the International Joint Conference on Theory and Practice of Software Development, Barcelona, Spain, March 1989*, number 352 in Lecture Notes in Computer Science, pages 36–59. Springer Verlag, Berlin, Heidelberg, New-York, 1989.

[Lafontaine, 1990] Christine Lafontaine. Formalization of the VDM reification in the DEVA meta-calculus. the human-leucocyte-antigen case study. In *IFIP Working Conference on Programming Concepts and Methods*, 1990.

[Lakatos, 1976] Imre Lakatos. *Proofs and Refutations*. Cambridge University Press, Cambridge, 1976.

[Lau & Prestwich, 1990] K.K. Lau and S.D. Prestwich. Top-down synthesis of recursive logic procedures from first-order logic specifications. In *Proceedings of the 7th International Conference on Logic Programming*, pages 667–684. MIT Press, 1990.

[Leivant, 1990] Daniel Leivant. Contracting proofs to programs. In Piergiorgio Odifreddi, editor, *Logic and Computer Science*, pages 279–328. Academic Press, London, 1990.

[Letz *et.al.*, 1992a] R. Letz, J. Schumann, S. Bayerl, and W. Bibel. SETHEO: A high-performance theorem prover. *Journal of Automated Reasoning*, 8:183–212, 1992.

[Letz *et.al.*, 1992b] Reinhold Letz, Johannes Schumann, Stephan Bayerl, and Wolfgang Bibel. SETHEO: A high-performance theorem prover. *Journal of Automated Reasoning*, 8:183–212, 1992.

[Lowry & Duran, 1989] Michael R. Lowry and R. Duran. Knowledge-based software engineering. In Avron Barr, Paul R. Cohen, and Edward A. Feigenbaum, editors, *The Handbook of Artificial Intelligence, Vol. IV*, chapter XX. Addison Wesley, Reading, MA, 1989.

[Lowry, 1987] Michael R. Lowry. Algorithm synthesis through problem reformulation. In *AAAI-87 — Proceedings of the 6^{th} AAAI National Conference on Artificial Intelligence, Seattle, July 1987*, Seattle, WA, July 13–17, 1987. Technical Report KES.U.87.10, Kestrel Institute, August 1987.

[Lowry, 1988] Michael R. Lowry. The structure and design of local search algorithms. In M. R. Lowry, R. McCartney, and D. R. Smith, editors, *Proceedings of the Workshop on Automating Software Design. AAAI-88, St. Paul, MN, August 25, 1988*, pages 88–94, August 1988.

[Lowry, 1989] Michael R. Lowry. *Algorithm Synthesis through Problem Reformulation.* PhD thesis, Stanford University, 1989.

[Lowry, 1991] Michael R. Lowry. Automating the design of local search algorithms. In Michael R. Lowry and Robert D. McCartney, editors, *Automating Software Design*, pages 515–546, Menlo Park, CA, 1991. AAAI Press / The MIT Press.

[Luo & Pollack, 1992] Zhaohui Luo and Robert Pollack. *The LEGO Proof Development System: A user's manual.* University of Edinburgh, 1992. LFCS Report ECS-LFVS-92-211.

[Luo, 1989] Zhaohui Luo. ECC: an extended calculus of constructions. In Rohit Parikh, editor, *LICS-89 — Proceedings of the Fourth Annual Symposium on Logic in Computer Science, 1989*, pages 385–395. IEEE Computer Society Press, 1989.

[Luo, 1990] Zhaohui Luo. *An extended calculus of constructions.* PhD thesis, University of Edinburgh, 1990.

[Manna & Waldinger, 1975] Zohar Manna and Richard J. Waldinger. Knowledge and reasoning in program synthesis. *Artificial Intelligence*, 6(2):175–208, 1975.

[Manna & Waldinger, 1979] Zohar Manna and Richard J. Waldinger. Synthesis: Dreams $\Rightarrow$ programs. *IEEE Transactions on Software Engineering*, SE-5(4):294–328, July 1979.

[Manna & Waldinger, 1980] Zohar Manna and Richard J. Waldinger. A deductive approach to program synthesis. *ACM Transactions on Programming Languages and Systems*, 2(1):90–121, 1980.

[Manna & Waldinger, 1985] Zohar Manna and Richard J. Waldinger. *The logical basis for computer programming*, volume I: Deductive Reasoning. Addison Wesley, Reading, MA, 1985.

[Martin-Löf, 1970] Per Martin-Löf. A theory of types. unpublished manuscript, 1970.

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

[Martin-Löf, 1982] Per Martin-Löf. Constructive mathematics and computer programming. In *6-th International Congress for Logic, Methodology and Philosophy of Science, 1979*, pages 153–175. North–Holland, Amsterdam, 1982.

[Martin-Löf, 1984] Per Martin-Löf. *Intuitionistic Type Theory*, volume 1 of *Studies in Proof Theory Lecture Notes.* Bibliopolis, Napoli, 1984.

[Martin-Löf, 1988] Per Martin-Löf. Mathematics of infinity. In P. Martin-Löf and G. Mints, editors, *COLOG-88: Proceedings of the International Conference on Computer Logic, Tallin, USSR, December 1988*, number 417 in Lecture Notes in Computer Science, pages 146–197. Springer Verlag, Berlin, Heidelberg, New-York, 1988.

[Mauny, 1991] Michel Mauny. Functional programming using CAML. Technical Report 129, Institut National de Recherche en Informatique et en Automatique, May 1991.

[Mendler *et.al.*, 1986] Paul Francis Mendler, Prakash Panangaden, and Robert L. Constable. Infinite objects in Type Theory. Technical Report TR 86-743, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, March 1986.

[Mendler, 1987a] Paul Francis Mendler. *Inductive definition in Type Theory.* PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, September 1987. TR 87-870.

[Mendler, 1987b] Paul Francis Mendler. Recursive types and type constraints in second order lambda calculus. In David Gries, editor, *LICS-87 — Proceedings of the Second Annual Symposium on Logic in Computer Science, Ithaca, New York, USA, June 1987*, pages 30–36. IEEE Computer Society Press, 1987.

[Milner, 1978] Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17(3):348–375, December 1978.

[Minsky, 1963] M. Minsky. Steps toward artificial intelligence. *Computers and Thought*, pages 406–450, 1963.

[Morris, 1969] J. B. Morris. E-resolution: an extension of resolution to include the equality relation. In *IJCAI-69 — Proceedings of the 1st International Joint Conference on Artificial Intelligence, Washington, DC, May 1969*, pages 287–294. Morgan Kaufmann, 1969.

[Nelson & Oppen, 1979] Greg Nelson and Derek. C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, October 1979.

[Nelson & Oppen, 1980] Greg Nelson and Derek. C. Oppen. Fast decision procedures based on congruence closure. *Journal of the Association for Computing Machinery*, 27(2):356–364, April 1980.

[Nerode, 1988] A. Nerode. Some lectures on intuitionistic logic. In Piergiorgio Odifreddi, editor, *Logic and Computer Science*, volume 1429 of *Lecture Notes in Mathematics*, pages 12–59. Springer Verlag, Berlin, Heidelberg, New-York, 1988.

[Neugebauer *et.al.*, 1989] G. Neugebauer, Bertram Fronhöfer, and Christoph Kreitz. XPRTS - an implementation tool for program synthesis. In D. Metzing, editor, *GWAI-89 — Proceedings of the 13th German Workshop on Artificial Intelligence, Eringerfeld, September 1989*, number 216 in Informatik Fachberichte, pages 348–357. Springer Verlag, Berlin, Heidelberg, New-York, 1989.

[Newell *et.al.*, 1963] A. Newell, M. Shaw, and H. Simon. Empirical explorations with the logic theory machine. *Computers and Thought*, pages 109–133, 1963.

[Nordström & Smith, 1984] Bengt Nordström and Jan M. Smith. Propositions and specifications of programs in Martin-Löfs Type Theory. *BIT*, 24:288–301, 1984.

[Nordström *et.al.*, 1990] Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löfs Type Theory. An introduction.* Clarendon Press, Oxford, 1990.

[Paige & Koenig, 1982] Robert Paige and Shaye Koenig. Finite differencing of computable expressions. *ACM Transactions on Programming Languages and Systems*, 4(3):402–454, July 1982.

[Paige, 1981] Robert A. Paige. *Formal Differentiation - A Program Synthesis Technique.* Artificial Intelligence, No. 6. UMI Research Press, Ann Arbor, MI, October 1981.

[Partsch & R., 1983] H. Partsch and Steinbrüggen R. Program transformation systems. *Computing Surveys*, 15(3):199–236, September 1983.

[Paulin-Mohring, 1989] Christine Paulin-Mohring. Extracting F_ω's programs from proofs in the Calculus of Constructions. In *Proceedings of the 16th ACM Symposium on Principles of Programming Languages*, pages 89–104, 1989.

[Paulson, 1986] Lawrence C. Paulson. Constructing recursion operators in intuituionistic type theory. *Journal of Symbolic Computation*, 2:325–355, 1986.

[Paulson, 1987] Lawrence C. Paulson. *Logic and Computation: Interactive Proof with Cambridge LCF*. Cambridge University Press, 1987.

[Paulson, 1989] Lawrence C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–397, 1989.

[Paulson, 1990] Lawrence C. Paulson. Isabelle: The next 700 theorem provers. In Piergiorgio Odifreddi, editor, *Logic and Computer Science*, pages 361–386. Academic Press, London, 1990.

[Pitts, 1989] Andrew M. Pitts. Non-trivial power types can't be subtypes of polymorphic types. In Rohit Parikh, editor, *LICS-89 — Proceedings of the Fourth Annual Symposium on Logic in Computer Science, 1989*, pages 6–13. IEEE Computer Society Press, 1989.

[Pollack, 1994] Robert Pollack. *The theory of LEGO – a proof checker for the extendend calculus of constructions*. PhD thesis, University of Edinburgh, 1994.

[Polya, 1945] G. Polya. *How to solve it*. Princeton University Press, Princeton, New Jersey, 1945.

[Prawitz, 1960] Dag Prawitz. An improved proof procedure. *Theoria*, 26:102–139, 1960.

[Prawitz, 1965] Dag Prawitz. *Natural Deduction: A Proof-Theoretical study*. Almquist & Wiksell, 1965.

[Quine, 1963] Willard Van Orman Quine. *Set Theory and its logic*. Belknap Press, Cambridge, MA, 1963.

[Reps & Teitelbaum, 1984] Thomas Reps and Tim Teitelbaum. The synthesizer generator. In *Proceedings ACM SIGPLAN SIGOA Symposium on Practical Software Development Environments*, April 1984.

[Reynolds, 1981] John C. Reynolds. *The craft of programming*. Prentice Hall, Englewood Cliffs, NJ, Englewood Cliffs, N.J., 1981.

[Richter, 1978] Michel M. Richter. *Logikkalküle*, volume 43 of *Teubner Studienbücher Informatik*. B.G.Teubner, Stuttgart, 1978.

[Robinson & Wos, 1969] G. Robinson and L. Wos. Paramodulation and theorem proving in first order theories with equality. *Machine Intelligence*, 4:135–150, 1969.

[Robinson, 1965] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, 12(1):23–41, January 1965.

[Rogers, 1967] Hartley jr. Rogers. *Theory of recursive functions and effective computability*. MIT Press, Cambridge, MA, 1967.

[Russel, 1908] Bertrand Russel. Mathematical logic as based on a theory of types. *Am. J. of Math.*, 30:222–262, 1908.

[Salvesen & Smith, 1988] Anne Salvesen and Jan M. Smith. The strength of the subset type in Martin-Löf's Type Theory. In Yuri Gurevich, editor, *LICS-88 — Proceedings of the Third Annual Symposium on Logic in Computer Science, Edinburgh, Scotland, July 1988*, pages 384–391. IEEE Computer Society Press, 1988.

[Sannella & Tarlecki, 1987] Donald Sannella and Andrzej Tarlecki. On observational equivalence and algebraic specification. *Journal of Computer and System Sciences*, 34:150–178, 1987.

[Sannella & Tarlecki, 1988] Donald Sannella and Andrzej Tarlecki. Toward formal development of programs from algebraic specifications: implementations revisited. *Acta Informatica*, 25(3):233–281, 1988.

[Sasaki, 1986] James T. Sasaki. *Extracting Efficient Code From Constructive Proofs*. PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, June 1986.

[Sato & Tamaki, 1989] T. Sato and H. Tamaki. First order compiler: A deterministic logic program synthesis algorithm. *Journal of Symbolic Computation*, 8:605–627, 1989.

[Scherlis, 1981] William Scherlis. Program improvement by internal specialization. In *Proceedings of the 8^{th} ACM Symposium on Principles of Programming Languages*, pages 41–49. Association for Computing Machinery, 1981.

[Schonberg *et.al.*, 1981] Edward Schonberg, Jacob Schwartz, and M. Sharir. An automatic technique for the selection of data representations in SETL programs. *ACM Transactions on Programming Languages and Systems*, 3(2):126–143, April 1981.

[Schütte, 1977] Kurt Schütte. *Proof Theory*. Springer Verlag, Berlin, Heidelberg, New-York, 1977.

[Schwartzbach, 1986] Michael Ignatieff Schwartzbach. *A category theoretic analysis of predicative Type Theory*. PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, December 1986. TR 86-793.

[Scott, 1972] Dana Scott. Lattice theory, data types, and semantics. In R. Rustin, editor, *Formal semantics of Programming Languages*, pages 65–106. Prentice Hall, Englewood Cliffs, NJ, 1972.

[Scott, 1976] Dana Scott. Data types as lattices. *SIAM Journal on Computing*, 5:522–287, 1976.

[Silverman *et.al.*, 1988] J. Silverman, V. Vickers, and J. Mooney. On the number of Costas arrays as a function of array size. In *Proceedings of the IEEE*, volume 76, pages 851–853, 1988.

[Smith & Lowry, 1990] Douglas R. Smith and Michael R. Lowry. Algorithm theories and design tactics. *Science of Computer Programming*, 14(2-3):305–321, October 1990. Report KES.U.89.3, Kestrel Institute.

[Smith & Parra, 1993] Douglas R. Smith and Eduardo A. Parra. Transformational approach to transportation scheduling. In *Proceedings of the 8th Knowledge-Based Software Engineering Conference, Chicago, Il, September 1993*, pages 60–68, 1993.

[Smith, 1982] Douglas R. Smith. Top-down synthesis of simple divide-and-conquer algorithms. Technical Report NPS52-82-011, Naval Postgraduate School, Monterey, California, 1982.

[Smith, 1983a] Douglas R. Smith. The structure of divide-and-conquer algorithms. Technical Report NPS52-83-002, Naval Postgraduate School, Monterey, California, March 1983.

[Smith, 1983b] Jan M. Smith. The identification of propositions and types in Martin-Löf's Type Theory: A programming example. In *International Conference on Foundations of Computation Theory, Borgholm, Sweden, August 1983*, pages 445–456. Springer Verlag, Berlin, Heidelberg, New-York, 1983.

[Smith, 1984] Jan M. Smith. An interpretation of Martin-Löf's Type Theory in a type-free theory of propositions. *Journal of Symbolic Logic*, 49(3):730–753, 1984.

[Smith, 1985a] Douglas R. Smith. The design of divide and conquer algorithms. *Science of Computer Programming*, 5:37–58, 1985.

[Smith, 1985b] Douglas R. Smith. Top-down synthesis of divide-and-conquer algorithms. *Artificial Intelligence*, 27(1):43–96, September 1985. (Reprinted in *Readings in Artificial Intelligence and Software Engineering*, C. Rich and R. Waters, Eds., Los Altos, CA, Morgan Kaufmann, 1986.).

[Smith, 1987a] Douglas R. Smith. Application of a strategy for designing divide-and-conquer algorithms. *Science of Computer Programming*, 8(3):213–229, June 1987. (also Technical Report KES.U.85.2, Kestrel Institute, March 1985).

[Smith, 1987b] Douglas R. Smith. Structure and design of global search algorithms. Technical Report KES.U.87.12, Kestrel Institute, November 1987. Revised Version, July 1988.

[Smith, 1988] Scott Fraser Smith. *Partial objects in Type Theory.* PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, August 1988. TR 88-938.

[Smith, 1990a] Douglas R. Smith. KIDS: A semi-automatic program development system. *IEEE Transactions on Software Engineering*, 16(9):1024–1043, September 1990. Special Issue on Formal Methods extended version: [Smith, 1990b].

[Smith, 1990b] Douglas R. Smith. KIDS: A semi-automatic program development system. Technical Report KES.U.90.1, Kestrel Institute, April 1990.

[Smith, 1991a] Douglas R. Smith. KIDS — a knowledge-based software development system. In Michael R. Lowry and Robert D. McCartney, editors, *Automating Software Design*, pages 483–514, Menlo Park, CA, 1991. AAAI Press / The MIT Press.

[Smith, 1991b] Douglas R. Smith. Structure and design of dynamic programming algorithms. Technical report, Kestrel Institute, 1991.

[Smith, 1991c] Douglas R. Smith. Structure and design of problem reduction generators. In B. Möller, editor, *Proceedings of the IFIP TC2 Working Conference on Constructing Programs from Specifications*, pages 91–124. Elsevier Science Publishers B.V. (North–Holland) Amsterdam, 1991.

[Smith, 1992] Douglas R. Smith. Constructing specification morphisms. Technical Report KES.U.92.1, Kestrel Institute, April 1992.

[Smith, 1993] Douglas R. Smith. Classification approach to design. Technical Report KES.U.93.4, Kestrel Institute, 1993.

[Stansifer, 1985] Ryan Stansifer. *Representing Constructive theories in High-level Programming languages.* PhD thesis, Cornell University. Department of Computer Science, Ithaca, NY 14853-7501, March 1985. TR 85-664.

[Stenlund, 1972] Sören Stenlund. *Combinators, λ-terms and Proof Theory.* D. Reidel, Dordrecht, The Netherlands, 1972.

[Suppes, 1972] Patrick Suppes. *Axiomatic Set Theory.* Dover Publications, 1972.

[Szabo, 1969] M. E. Szabo, editor. *The collected papers of Gerhard Gentzen.* North–Holland, Amsterdam, 1969.

[Tait, 1967] William W. Tait. Intensional interpretation of functionals of finite type. *Journal of Symbolic Logic*, 32(2):187–199, 1967.

[Takeuti, 1975] Gaisi Takeuti. *Proof Theory.* North–Holland, Amsterdam, 1975.

[Teitelbaum & Reps, 1981] Tim Teitelbaum and Thomas Reps. The Cornell Program Synthesizer: A syntax-directed environment. *Communications of the Association for Computing Machinery*, 24(9), September 1981.

[Troelsta, 1969] Ann S. Troelsta. *Principles of Intuitionism.* Springer Verlag, Berlin, Heidelberg, New-York, 1969.

[Troelsta, 1977] Ann S. Troelsta. Aspects of constructive mathematics. In Jon Barwise, editor, *Handbook of mathematical logic*, chapter D.5, pages 974–1053. North–Holland, Amsterdam, 1977.

[Turner, 1984] R. Turner. *Logics for Artificial Intelligence*. Ellis Horwood Ltd., 1984.

[van Benthem Jutting, 1977] L. S. van Benthem Jutting. *Checking Landau's Grundlagen in the AUTOMATH System*. PhD thesis, Eindhoven University of Technology, Eindhoven, Netherlands, March 1977. also: Mathematical Centre Tracts 83, Math. Centre, Amsterdam, 1979.

[van Dalen, 1986] Dirk van Dalen. Intuitionistic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, chapter III.4, pages 224–339. D. Reidel, 1986.

[van Heijenoort, 1967] J. van Heijenoort. *From Frege to Gödel: A Sourcebook of Mathematical Logic*. Harvard University Press, Cambridge, Massachusetts, 1967.

[Wallen, 1990] Lincoln Wallen. *Automated deduction in nonclassical logic*. MIT Press, Cambridge, Mass, 1990.

[Weber, 1990] Matthias Weber. Formalization of the Bird-Meertens algorithmic calculus in the DEVA meta-calculus. In *IFIP Working Conference on Programming Concepts and Methods*, 1990.

[Weis *et.al.*, 1990] Pierre Weis, Maria-Virginia Aponte, Alain Laville, Michel Mauny, and Acsander Suarez. The CAML reference manual. Rapports Techniques 121, Institut National de Recherche en Informatique et en Automatique, September 1990.

[Whitehead & Russell, 1925] Alan N. Whitehead and Bertrand Russell. *Principia Mathematicae*, volume 1. Cambridge University Press, Cambridge, MA, 1925.

[Wirth, 1971] N. Wirth. Program development by stepwise refinement. *Communications of the Association for Computing Machinery*, 14(4):221–227, April 1971.

[Wos *et.al.*, 1984] Larry Wos, Russ Overbeek, Ewing Lusk, and J. Boyle. *Automated Reasoning*. Prentice Hall, Englewood Cliffs, NJ, 1984.

[Wos *et.al.*, 1990] L. Wos, S. Winker, W. McCune, R. Overbeek, E. Lusk, R. Stevens, and R. Butler. Automated reasoning contributes to mathematics and logic. In M. E. Stickel, editor, *Proceedings of the 10^{th} Conference on Automated Deduction, Kaiserslautern, Germany, July 1990*, volume 449 of *Lecture Notes in Computer Science*, pages 485–499. Springer Verlag, Berlin, Heidelberg, New-York, 1990.

[Wu, 1986] Wen Tsün Wu. Basic principles of mechanical theorem proving in geometries. *Journal of Automated Reasoning*, 2(4):221–251, 1986.