A Scheme Bibliography

Ozan S. Yigit

York University
Computing and Communication Services

oz@nexus.yorku.ca

[11791] IEEE Std 1178-1990, *IEEE Standard for the Scheme Programming Language*, Institute of Electrical and Electronic Engineers, Inc., New York, NY, 1991.

[ASS85] Harold Abelson, Gerald Jay Sussman and Julie Sussman, *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge, Mass., 1985.

[AbS88] Harold Abelson and Gerald Jay Sussman, Lisp: A Language for Stratified Design, *BYTE*, February 1988, 207-218.

[AdR88] Norman Adams and Jonathan Rees, Object-Oriented Programming in Scheme, *Conference Record of the 1988 ACM Conference on Lisp and Functional Programming*, August 1988, 277-288.

[AsS89] J. Michael Ashley and Richard M. Salter, A Revised State Space Model for a Logic Programming Embedding in Scheme, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[Bar89] Joel F. Bartlett, SCHEME->C a Portable Scheme-to-C Compiler, Research Report 89/1, Dec. Western Research Laboratory, Palo Alto, California, January 1989.

[BaJ86] David H. Bartley and John C. Jensen, The Implementation of PC Scheme, *Proceedings of the 1986 ACM Conference on Lisp and Functional Programming*, 1986, 86-93.

[BGH82] John Batali, Edmund Goodhue, Chris Hanson, Howie Shrobe, Richard M. Stallman and Gerald Jay Sussman, The Scheme-81 Architecture - System and Chip, *Proceedings, Conference on Advanced Research in VLSI*, Artech House, Dedham MA., 1982, 69-77.

[BaR88] Alan Bawden and Jonathan Rees, Syntactic Closures, *Proceedings of the 1988 ACM Symposium on LISP and Functional Programming*, Salt Lake City, Utah., July 1988.

[Bec91] Brian Beckman, A Scheme for Little Languages in Interactive Graphics, *Software-Practice and Experience 21*, 2 (Feb 1991), 187-207, John Wiley & Sons, Ltd.

[Bon92] Anders Bondorf, Improving Binding Times without Explicit CPS-Conversion, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 1-10.

[Bon90] Pierre Bonzon, A Metacircular Evaluator for a Logical Extension of Scheme, *Lisp and Symbolic Computation: An International Journal 3*, 2 (March 1990), 113-133, Kluwer Academic Publishers.

[ChF83] Pee Hong Chen and Daniel P. Friedman, Prototyping Data Flow by Translation Into Scheme, Computer Science Technical Report #147, Indiana University, Bloomington, Indiana, August 1983.

[CCO83] Pee Hong Chen , W.Y. Chi, E.M. Ost, L.D. Sabbagh and G. Springer, Scheme Graphics Reference Manual, Computer Science Technical Report No. 145, Indiana University , Bloomington, Indiana, August 1983.

[ChS85] Pee Hong Chen and David Sabbagh, Scheme as an Interactive Graphics Programming Environment, Computer Science Technical Report No. 166, Indiana University, Bloomington, Indiana, March 1985.

[Cli84] William Clinger, The Scheme 311 compiler: An Exercise in Denotational Semantics, *Conference Record of the 1984 ACM Symposium on Lisp and Functional Programming*, 1984, 356-364.

[Cli85] The Revised Revised Report on Scheme, or An Uncommon Lisp, MIT AI Memo 848, Massachusetts Institute of Technology, Cambridge, Mass., August 1985. Also published as Computer Science Department Technical Report 174, Indiana University, June 1985.

[CFW85] William Clinger, Daniel P. Friedman and Mitchell Wand, A Scheme for a Higher-Level Semantic Algebra, in *Algebraic Methods in Semantics*, J. Reynolds, M. Nivat (ed.), Cambridge University Press, London, 1985, 237-250.

[Cli87] William Clinger, The Scheme of things: Streams versus Generators, Technical Report, Tektronix, Inc., 1987.

[CHO88] William D. Clinger, Anne H. Hartheimer and Eric M. Ost, Implementation Strategies for Continuations, *Conference Record of the 1988 ACM Conference on Lisp and Functional Programming*, August 1988, 124 131.

[Cli88] William Clinger, Semantics of Scheme, *BYTE*, February 1988, 221-227.

[Cli91a] William Clinger, Hygenic Macros Through Explicit Renaming, *Lisp Pointers IV*, 4 (October-December 1991), 17-23, ACM.

[Cli91b] William Clinger, Macros In Scheme, *Lisp Pointers IV*, 4 (October-December 1991), 25-28, ACM.

[CuR90] Pavel Curtis and James Rauen, A Module System for Scheme, *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.

[Dan89a] Olivier Danvy, Programming with Tighter Control, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[Dan89b] Olivier Danvy, Combiner Logiquement en Scheme, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[DaL92] Olivier Danvy and Julia L. Lawall, Back to Direct Style II: First-Class Continuations, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 299-310.

[Del89] Vincent Delacour, Picolo Expresso, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[DDD84] *MIT Scheme Manual, Seventh Edition*, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass., September 1984.

[DDC89] Alain Deutsch, Renaud Dumeur, Charles Consel and Jean-Daniel Fekete, CSKIM: An Extended Dialect of Scheme, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[Dic92] Ken Dickey, The Scheme Programming Language, *Computer Language*, June 1992.

[DFF87] Bruce F. Duba, Matthias Felleisen and Daniel P. Friedman, Dynamic Identifiers can be Neat, Computer Science Technical Report No. 220, Indiana University, Bloomington, Indiana, April 1987.

[DwD81] Rex A. Dwyer and R. Kent Dybvig, A SCHEME for Distributed Processes, Computer Science Department Technical Report #107, Indiana University, Bloomington, Indiana, April 1981.

[Dyb83] R. Kent Dybvig, C-Scheme, Computer Science Department Technical Report #149 (MS Thesis), Indiana University, Bloomington, Indiana, 1983.

[DyS85] R. Kent Dybvig and Bruce T. Smith, *Chez Scheme Reference Manual Version 1.0*, Cadence Research Systems, Bloomington, Indiana, May 1985.

[DFH86] R. Kent Dybvig, Daniel P. Friedman and Christopher T. Haynes, Expansion-Passing style: Beyond Conventional Macros, *Conference Record of the 1986 ACM Conference on Lisp and Functional Programming*, 1986, 143-150.

[Dyb87a] R. Kent Dybvig, *The Scheme Programming Language*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[Dyb87b] R. Kent Dybvig, Three Implementation Models for Scheme, Department of Computer Science Technical Report #87-011 (Ph.D. Dissertation), University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, April 1987.

[DyH88] R. Kent Dybvig and Robert Hieb, A Variable-Arity Procedural Interface, *Proceedings of the 1988 ACM Symposium on LISP and Functional Programming*, Salt Lake City, Utah, July 1988, 106-115. Also Indiana University Computer Science Department Technical Report #247.

[DFH88] R. Kent Dybvig, Daniel P. Friedman and Christopher T. Haynes, Expansion-Passing Style: A General Macro Mechanism, *Lisp and Symbolic Computation: An International Journal 1*, 1 (June 1988), 53-76, Kluwer Academic Publishers.

[DyH89] R. Kent Dybvig and Robert Hieb, Engines from Continuations, *Journal of Computer Languages 14*, 2 (1989), 109-123. Also Indiana University Computer Science Department Technical Report #254.

[DyH90a] R. Kent Dybvig and Robert Hieb, Continuations and Concurrency, *Proceedings of the Second ACM SIGPLAN Notices Symposium on Principles and Practice of Parallel Programming*, Seattle, Washington, March 1990, 128-136. Also Indiana University Computer Science Department Technical Report #256.

[DyH90b] R. Kent Dybvig and Robert Hieb, A New Approach to Procedures with Variable Arity, *Lisp and Symbolic Computation: An International Journal 3*, 3 (September 1990), 229-244, Kluwer Academic Publishers.

[Dyb92] Kent Dybvig, Writing Hygenic Macros in Scheme with Syntax-Case, Computer Science Department Technical Report #356, Indiana University, Bloomington, Indiana, June 1992.

[Eis85] Michael A. Eisenberg, Bochser: An Integrated Scheme Programming System, MIT Computer Science Technical Report 349, Massachusetts Institute of Technology, Cambridge, Mass., October 1985.

[Eis88] Michael Eisenberg, *Programming In Scheme*, The Scientific Press, Redwood City, CA, 1988.

[ECH90] Michael Eisenberg, William Clinger and Anne Hartheimer, *Programming In MacScheme*, The Scientific Press, Redwood City, CA, 1990.

[Fee86] Marc Feeley, *Deux Approches a' L'implantation du Language Scheme*, M.Sc. Thesis, De'partement d'Informatique et de Recherche Ope'rationelle, University of Montreal, May

1986.

[FeL86] Marc Feeley and Guy LaPalme, Closure Generation based on viewing LAMBDA as EPSILON plus COMPILE, Submitted for Publication, 1986.

[FeL87] Marc Feeley and Guy LaPalme, Using Closures for Code Generation, *Journal of Computer Languages 12,* 1 (1987), 47-66, Pergamon Press.

[FeM90] Marc Feeley and James S. Miller, A Parallel Virtual Machine for Efficient Scheme Compilation, *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.

[Fel85] Matthias Felleisen, Transliterating Prolog into Scheme, Computer Science Technical Report #182, Indiana University, Bloomington, Indiana, October 1985.

[FeF86] Matthias Felleisen and Daniel P. Friedman, Control operators, the SECD-machine, and the lambda-calculus, *3rd Working Conference on the Formal Description of Programming Concepts*, Ebberup, Denmark, August 1986, 193-219.

[FFK86] Matthias Felleisen, Daniel P. Friedman, Eugene E. Kohlbecker and Bruce Duba, Reasoning with Continuations, *Proceedings of the Symposium on Logic in Computer Science*, Washington DC, June 1986, 131-141.

[FeF86] Matthias Felleisen and Daniel P. Friedman, A Closer Look At Export and Import Statements, *Journal of Computer Languages 11*, 1 (1986), 29-37, Pergamon Press.

[FeF87] Matthias Felleisen and Daniel P. Friedman, A Syntactic Theory of Sequential State, Computer Science Dept. Technical Report #230, Indiana University, Bloomington, Indiana, October 1987.

[Fel87] Matthias Felleisen, The Calculi of Lambda-v-CS Conversion: A Syntactic Theory of Control And State in Imperative Higher-Order Programming Languages, Computer Science Technical Report #226. (Ph.D. Dissertation), Indiana University, Bloomington, Indiana, August 1987.

[FFK87] Matthias Felleisen, Daniel P. Friedman, Eugene E. Kohlbecker and Bruce Duba, A Syntactic Theory of Sequential Control, *Theoretical Computer Science 52*(1987), 205-237.

[FeF87] Matthias Felleisen and Daniel P. Friedman, A Reduction Semantics for Imperative Higher-Order Languages, *Parallel Architectures and Languages Europe 259*(1987), 206-223, Springer-Verlag.

[Fel87] Matthias Felleisen, Reflections on Landin's J-Operator: A Partly Historical Note, *Journal of Computer Languages 12*, 3/4 (1987), 197-207, Pergamon Press.

[FFD87] Matthias Felleisen, Daniel P. Friedman, Bruce Duba and John Merrill, Beyond Continuations, Computer Science Dept. Technical Report #216, Indiana University, Bloomington, Indiana, February, 1987.

[FeF87] Matthias Felleisen and Daniel P. Friedman, A Calculus for Assignments in Higher-Order Languages, *Conference Record of the 14th Annual ACM Symposium on Principles of Programming Languages*, Munich, West Germany, January 1987, 314-345.

[FWF88] Matthias Felleisen, Mitchell Wand, Daniel P. Friedman and Bruce Duba, Abstract Continuations: A Mathematical Semantics for Handling Functional Jumps, *Proceedings of the 1988 ACM Symposium on LISP and Functional Programming*, Salt Lake City, Utah., July 1988.

[Fel88] Matthias Felleisen, λ-vs-CS: An Extended λ-Calculus for Scheme, *Conference Record of the 1988 ACM Conference on Lisp and Functional Programming*, August 1988, 72-85.

[FeH9.] Matthias Felleisen and Robert Hieb, The Revised Report on the Syntactic Theories of Sequential Control and State., Computer Science Technical Report No. 100, Rice University, June 1989..

[FCF83] Carol Fessenden, William Clinger, Daniel P. Friedman and Christopher T. Haynes, Scheme 311 version 4 Reference Manual, Computer Science Technical Report 137, Indiana University, Bloomington, Indiana, February 1983. Superseded by Computer Science Technical Report 153, 1985.

[For91] Alessandro Forin, Futures, in *Topics in Advanced Language Implementation*, Peter Lee (ed.), The MIT Press, Cambridge, MASS, 1991.

[FrF88] John Franco and Daniel P. Friedman, Creating Efficient Programs by Exchanging Data for Procedures, Computer Science Technical Report #245, Indiana University, Bloomington, Indiana, March 1988.

[FrF90] John Franco and Daniel P. Friedman, Towards A Facility for Lexically Scoped, Dynamic Mutual Recursion in Scheme, *Journal of Computer Languages 15*, 1 (1990), 55-64, Pergamon Press.

[FFJ90] John Franco, Daniel P. Friedman and Steven Johnson, Multi-way Streams in Scheme, *Journal of Computer Languages 15*, 2 (1990), 109-125.

[FrF92] Eric T. Freeman and Daniel P. Friedman, Characterizing the paralation model using dynamic assignment, Computer Science Department Technical Report #348, Indiana University, Bloomington, Indiana, March 1992.

[FHK84] Daniel P. Friedman, Christopher T. Haynes and Eugene E. Kohlbecker, Programming with Continuations, in *Program Transformation and Programming Environments*, P. Pepper (ed.), Springer-Verlag, 1984, 263-274.

[FrW84] Daniel P. Friedman and Mitchell Wand, Reification: Reflection Without Metaphysics, *Conference Record of the 1984 ACM Symposium on LISP and Functional Programming*, Austin, TX., August 1984, 348-355.

[FHK85] Daniel P. Friedman, Christopher T. Haynes, Eugene E. Kohlbecker and Mitchell Wand, Scheme 84 Interim Reference Manual, Computer Science Technical Report 153, Indiana University, Bloomington, Indiana, January 1985.

[FrH85] Daniel P. Friedman and Christopher T. Haynes, Constraining Control, *Proceedings of the Twelfth Annual Symposium on Principles of Programming Languages*, New Orleans, LA., January 1985, 245-254.

[FrF86] Daniel P. Friedman and Matthias Felleisen, *The Little LISPer: Second Edition*, Science Research Associates, Inc., Palo Alto, California, 1986.

[FrF87] Daniel P. Friedman and Matthias Felleisen, *The Little LISPer*, MIT Press, 1987. Trade Edition.

[FWH92] Daniel P. Friedman, Mitchell Wand and Christopher T. Haynes, *Essentials of Programming Languages*, MIT Press and McGraw-Hill, 1992.

[Han90] Chris Hanson, Efficient Stack Allocation for Tail-Recursive Languages, *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.

[Han91] Chris Hanson, A Syntactic Closures Macro Facility, *Lisp Pointers IV*, 4 (October-December 1991), 9-16, ACM.

[HaF84] Christopher T. Haynes and Daniel P. Friedman, Engines Build Process Abstractions, *Conference Record of the 1984 ACM Symposium on Lisp and Functional Programming*, Austin, TX., 1984, 18-24.

[HFW84] Christopher T. Haynes, Daniel P. Friedman and Mitchell Wand, Continuations and Coroutines, *Conference Record of the 1984 ACM Symposium on Lisp and Functional Programming*, Austin, TX., 1984, 293-298.

[Hay86] Christopher T. Haynes, Logic Continuations, *Proceedings of the Third International Conference on Logic Programming*, July 1986, 671-685.

[HFW86] Christopher T. Haynes, Daniel P. Friedman and Mitchell Wand, Obtaining Coroutines With Continuations, *Journal of Computer Languages 11*, 3/4 (1986), 143-153, Pergamon Press.

[HaF87a] Christopher T. Haynes and Daniel P. Friedman, Abstracting Timed Preemption with Engines, *Journal of Computer Languages 12*, 2 (1987), 109-121, Pergamon Press.

[HaF87b] Christopher T. Haynes and Daniel P. Friedman, Embedding Continuations in Procedural Objects, *ACM Transactions on Programming Languages and Systems 9*, 4 (October 1987), 582-598.

[Hen92] Fritz Henglein, Global Tagging Optimization by Type Inference, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 205-215.

[HDB90] Robert Hieb, R. Kent Dybvig and Carl Bruggeman, Representing Control in the Presence of First-Class Continuations, *Proceedings of the SIGPLAN Notices '90 Conference on Programming Language Design and Implementation*, White Plains, New York, June 1990, 66-77.

[HDB92] Robert Hieb, Kent Dybvig and Carl Bruggeman, Syntactic Abstraction in Scheme, Computer Science Department Technical Report #355, Indiana University, Bloomington, Indiana, June 1992.

[III89] Williams Ludwell Harrison III, The Interprocedural Analysis and Automatic Parallellization of Scheme Programs, *Lisp and Symbolic Computation: An International Journal 2*, 3/4 (October 1989), , Kluwer Academic Publishers.

[JaP92a] Suresh Jagannathan and Jim Philbin, A Foundation for an Efficient Multi-Threaded Scheme System, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 345-357.

[JaP92b] Suresh Jagannathan and Jim Philbin, A Customizable Substrate for Concurrent Languages, *Proceedings of the Sigplan 92 Conference on Programming Language Design and Implementation*, San Francisco, CA, July 1992.

[JrS76] Guy Lewis Steele Jr. and Gerald Jay Sussman, Lambda, the Ultimate Imperative, MIT AI Memo 353, Massachusetts Institute of Technology, Cambridge, Mass., March 1976.

[Jr.76] Guy Lewis Steele Jr., Lambda, the Ultimate Declarative, MIT AI Memo 379, Massachusetts Institute of Technology, Cambridge, Mass., November 1976.

[Jr.77a] Guy Lewis Steele Jr., Debunking the ‘‘Expensive Procedure Call’’ Myth, or Procedure Call Implementations Considered Harmful, or LAMBDA, the Ultimate GOTO, *ACM Conference Proceedings*, 1977, 153-162.

[Jr.77b] Guy Lewis Steele Jr., Macaroni is Better than Spaghetti, *Proceedings of the Symposium on Artificial Intelligence and Programming Languages*, August 1977, 60-66. Special joint issue of SIGPLAN Notices Notices 12(8) and SIGART Newsletter 64.

[JrS78a] Guy Lewis Steele Jr. and Gerald Jay Sussman, The Revised Report on Scheme, a Dialect of Lisp, MIT AI Memo 452, Massachusetts Institute of Technology, Cambridge, Mass., January 1978.

[JrS78b] Guy Lewis Steele Jr. and Gerald Jay Sussman, The Art of the Interpreter, or the Modularity Complex (parts zero, one, and two), MIT AI Memo 453, Massachusetts Institute of Technology, Cambridge, Mass., May 1978.

[Jr.78] Guy Lewis Steele Jr., Rabbit: a Compiler for Scheme, MIT AI Memo 474, Massachusetts Institute of Technology, Cambridge, Mass., May 1978.

[JrS79] Guy Lewis Steele Jr. and Gerald Jay Sussman, Design of LISP-Based Processors or, SCHEME: A Dielectric LISP or, Finite Memories Considered Harmful or, LAMBDA: The Ultimate Opcode, MIT-AI Memo 514, Massachusetts Institute of Technology, Cambridge, Mass., 1979.

[JrS80] Guy Lewis Steele Jr. and Gerald Jay Sussman, Design of a Lisp-based Processor, *Comm. ACM 23*, 11 (November 1980), 628-645.

[Jr.80] Guy Lewis Steele Jr., Compiler Optimization Based on Viewing LAMBDA as RENAME + GOTO, in *AI: An MIT Perspective*, Patrick Henry Winston and Richard Henry Brown (ed.), MIT Press, Cambridge, Mass., 1980.

[JrS80] Guy Lewis Steele Jr. and Gerald Jay Sussman, The Dream of a Lifetime: a Lazy Variable Extent Mechanism, *Conference Record of the 1980 Lisp Conference*, 1980, 163-172.

[Kam90] Samuel Kamin, *Programming Languages: An Interpreter-based Approach*, Addison-Wesley, Reading, Mass., 1990.

[KaL89] Simon M. Kaplan and Joseph P. Loyall, GARP/Scheme: Implementing a Concurrent, Object-Based Language, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[KaW90] Morry Katz and Daniel Weise, Continuing Into the Future: On the Interaction of Futures and First-Class Continuations, *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.

[KiZ89] Tan Gon Kim and Bernard P. Zeigler, The DEVS-Scheme Modelling and Simulation Environment, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[KFF86] Eugene E. Kohlbecker, Daniel P. Friedman, Matthias Felleisen and Bruce Duba, Hygienic Macro Expansion, *Symposium on LISP and Functional Programming*, August 1986, 151-161. To appear in Lisp and Symbolic Computation.

[Koh86] Eugene E. Kohlbecker, Syntactic Extensions in the Programming Language Lisp, Computer Science Technical Report #199 (Ph.D. Dissertation), Indiana University, Bloomington, Indiana, August 1986.

[KoW87] Eugene E. Kohlbecker and Mitchell Wand, Macro-by-Example: Deriving Syntactic Transformations from their Specifications, *Conference Record of the Fourteenth Annual ACM Symposium on Principles of Programming Languages*, Munich, West Germany, Jan 1987, 77-84.

[KKR86] David Kranz, Richard Kelsey, Jonathan A. Rees, Paul Hudak, James Philbin and Norman I. Adams, Orbit: An Optimizing Compiler for Scheme, *Proceedings of the SIGPLAN Notices '86 Symposium on Compiler Construction*, June 1986, 219-233. Published as SIGPLAN Notices Notices 21(7), July 1986.

[Kra88] David Kranz, Orbit: An Optimizing Compiler for Scheme, Computer Science Technical report #632 (Ph.D. Dissertation), Yale University, 1988.

[LaP86] Kevin J. Lang and Barak A. Pearlmutter, Oaklisp: an Object-Oriented Scheme with First Class Types, *ACM Conference on Object-Oriented Systems, Programming, Languages and Applications*, September 1986, 30-37.

[LaP88] Kevin J. Lang and Barak A. Pearlmutter, Oaklisp: an Object-Oriented Dialect of Scheme, *Lisp and Symbolic Computation: An International Journal 1*, 1 (May 1988), 39-51, Kluwer Academic Publishers.

[LaF89a] Guy Lapalme and Marc Feeley, Micro-Scheme, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[LaF89b] Julia L. Lawall and Daniel P. Friedman, Embedding the Self Language in Scheme, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[LaF92] Julia L. Lawall and Daniel P. Friedman, Toward leakage containment, Computer Science Department Technical Report #346, Indiana University, Bloomington, Indiana, February 1992.

[LeF92] Shinn-Der Lee and Daniel P. Friedman, First-Class Extents, Computer Science Department Technical Report #350, Indiana University, Bloomington, Indiana, March 1992.

[McD80] Drew McDermott, An Efficient Environment Allocation Scheme in an Interpreter for a Lexically-Scoped Lisp, *Conference Record of the 1980 Lisp Conference*, 1980, 154-162. Proceedings reprinted by ACM.

[Mil87] James S. Miller, A Parallel Processing System Based on MIT Scheme, MIT LCS Technical Report 402 (Ph.D. Dissertation), Massachusetts Institute of Technology, Cambridge, Mass., August 1987.

[MiR91] James Miller and Guillermo Rozas, Free Variables and First-Class Environments, *Lisp and Symbolic Computation: An International Journal 3*, 4 (1991), 107-141, Kluwer Academic Publishers.

[MuP80] Steven S. Muchnick and Uwe F. Pleban, A Semantic Comparison of Lisp and Scheme, *Conference Record of the 1980 Lisp Conference*, 1980, 56-65.

[Nor90] Kurt Normark, Simulation of Object-Oriented Concepts and Mechanisms in Scheme, Institute for Electronic Systems Technical Report 90-01, Aalborg University, Aalborg, Denmark, January 1990.

[PeL91] Barak A. Pearlmutter and Kevin J. Lang, The Implementation of Oaklisp, in *Topics in Advanced Language Implementation*, Peter Lee (ed.), The MIT Press, Cambridge, MASS, 1991.

[PiB89] André Pic and Michel Briand, Visual Programming with Generators, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[Ple79] Uwe F. Pleban, The Standard Semantics of a Subset of SCHEME, a Dialect of LISP, Computer Science Technical Report Tech. Rep.-79-3, University of Kansas, Lawrence, Kansas, July 1979.

[Ple80] Uwe F. Pleban, A Denotational Approach to Flow Analysis and Optimization of SCHEME, A Dialect of LISP, Ph.D. Dissertation, University of Kansas, Lawrence, Kansas, 1980.

[Ple91] Uwe F. Pleban, Compilation Issues in the Screme Implementation for the 88000, in *Topics in Advanced Language Implementation*, Peter Lee (ed.), The MIT Press, Cambridge, MASS, 1991.

[Que89] Christian Queinnec, Validation Suite Generation, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[Ram92] John D. Ramsdell, An Operational Semantics for Scheme, *Lisp Pointers V*, 2 (April-June 1992), , ACM.

[ReA82] Jonathan A. Rees and Norman I. Adams, T: A Dialect of Lisp or, LAMBDA: The Ultimate Software Tool, *Conference Record of the 1982 ACM Symposium on Lisp and Functional Programming*, 1982, 114-122.

[RAM84] Jonathan A. Rees, Norman I. Adams and James R. Meehan, *The T manual, fourth edition*, Yale University Computer Science Department, January 1984.

[ReC86] Jonathan A. Rees and William Clinger, eds., Revisedˆ3 Report on the Algorithmic Language Scheme, *ACM Sigplan Notices 21*, 12 (December 1986), .

[Ree89] Jonathan Rees, Modular Macros, Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1989.

[ReD92] Jonathan Rees and Bruce Donald, Program Mobile Robots in Scheme, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, 2681-2688.

[Rey72] John Reynolds, Definitional Interpreters for Higher Order Programming Languages, *ACM Conference Proceedings*, 1972, 717-740.

[RoM92] John H. Rose and Hans Muller, Integrating the Scheme and C Languages, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 247-259.

[RBC89] J. C. Royer, J. P. Braquelaire, P. Casteran, M. Desainte-Catherine and J. G. Penaud, Le modèle OBJScheme: principes et applications, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[Roz84] Guillermo J. Rozas, Liar, an Algol-like Compiler for Scheme, S. B. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, January 1984.

[Roz92] Guillermo Rozas, Taming the Y Operator, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 226-234.

[Sé89] Nitsan Séniak, Compilation de Scheme par spécialisation explicite, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[SaF92] Amr Sabry and Matthias Felleisen, Reasoning about Programs in Continuation-Passing Style, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 288-298.

[ScS84] Richard Schooler and James W. Stamos, Proposal For a Small Scheme Implementation, MIT LCS Memo TM-267, Massachusetts Institute of Technology, Cambridge, Mass., October 1984.

[SSS85] *MacScheme Reference Manual*, Semantic Microsystems, Sausalito, California, 1985.

[Shi88a] Olin Shivers, The Semantics of Scheme Control Flow Analysis (Preliminary)., Technical Report ERGO-90-090, CMU School of Computer Science, Pittsburgh, Penn., November 1988.

[Shi88b] Olin Shivers, Control Flow Analysis in Scheme, *Proceedings of the Sigplan 1988 Conference on Programming Language Design and Implementation*, Atlanta, Georgia, June 1988, 164-174.

[Shi90] Olin Shivers, Data-Flow Analysis and Type Recovery in Scheme. , Technical Report CMU-CS-90-115, CMU School of Computer Science, Pittsburgh, Penn., March 1990. Also to appear in Topics in Advanced Language Implementation, Ed. Peter Lee, MIT Press..

[Shi91] Olin Shivers, Data-Flow analysis and Type Recovery in Scheme, in *Topics in Advanced Language Implementation*, Peter Lee (ed.), The MIT Press, Cambridge, MASS, 1991.

[SiF90] Dorai Sitaram and Matthias Felleisen, Control Delimiters and Their Hierarchies, *Lisp and Symbolic Computation: An International Journal 3*, 1 (January 1990), 67-99, Kluwer Academic Publishers.

[SiF91] Dorai Sitaram and Matthias Felleisen, Modeling Continuations Without Continuations, *Proceedings of the Eighteenth ACM Symposium on Principles of Programming Languages*, 1991, 185-196.

[Sla87] Stephen Slade, *The T programming Language*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1987.

[SpF89] George Springer and Daniel P. Friedman, *Scheme and the Art of Programming*, MIT Press and McGraw-Hill, 1989.

[SOS85] Amitabh Srivastava, Don Oxley and Aditya Srivastava, An (other) Integration of Logic and Functional Programming, *Proceedings of the Symposium on Logic Programming*, 1985, 254-260.

[Str89] Robert Strandh, OOOZ, A multi-User Programming Environment Based on Scheme, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[SuJ75] Gerald Jay Sussman and Guy Lewis Steele Jr., Scheme: an Interpreter for Extended Lambda Calculus, MIT AI Memo 349, Massachusetts Institute of Technology, Cambridge, Mass., December 1975.

[SHJ81] Gerald Jay Sussman, Jack Holloway, Guy Lewis Steele Jr. and Alan Bell, Scheme-79 - Lisp on a Chip, *IEEE Computer 14*, 7 (July 1981), 10-21, IEEE.

[Sus82] Gerald Jay Sussman, LISP, Programming and Implementation, in *Functional Programming and its Applications*, Darlington, Henderson, Turner (ed.), Cambridge University Press, London, 1982.

[Teo90] Dan Teodosiu, HARE: A Compiler for Scheme, Master's Thesis, Bucharest Polytechnic Institute, June 1990.

[Teo91] Dan Teodosiu, HARE: An Optimizing Portable Compiler for Scheme, *ACM Sigplan Notices 26*, 1 (Jan 1991), .

[TTT85] *TI Scheme Language Reference Manual*, Texas Instruments, Inc., November 1985. Preliminary version 1.0.

[Tun92a] Sho-Huan Simon Tung, Merging interactive, modular and object-oriented programming, Computer Science Department Technical Report #349, Indiana University, Bloomington, Indiana, March 1992.

[Tun92b] Sho-Huan Simon Tung, Interactive Modular Programming in Scheme, *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, San Francisco, USA, June 1992, 86-95.

[Ulr89] John Wade Ulrich, Enumeration Algorithms and Non-deterministic Programming in Scheme, in *BIGRE Bulletin*, André Pic, Michel Briand, Jean Bézivin (ed.), July 1989. Putting Scheme to Work.

[VeP89] Steven R. Vegdahl and Uwe F. Pleban, The Runtime Environment for Screme, a Scheme Implementation on the 88000, *Proceedings of the Third International Conference on Architectural Support for Programming Languages and Operating Systems*, Boston, Mass., April 1989, 172-182.

[Wan78] Mitchell Wand, Continuation-Based Program Transformation Strategies, *Journal of the ACM 27*, 1 (1978), 174-180.

[WaF78] Mitchell Wand and Daniel P. Friedman, Compiling Lambda Expressions Using Continuations and Factorizations, *Journal of Computer Languages 3*(1978), 241-263.

[Wan80a] Mitchell Wand, Continuation-Based Multiprocessing, *Conference Record of the 1980 Lisp Conference*, 1980, 19-28.

[Wan80b] Mitchell Wand, SCHEME Version 3.1 Reference Manual, Computer Science Technical Report 93, Indiana University, Bloomington, Indiana, June 1980.

[Wan86a] Mitchell Wand, From Interpreter to Compiler: A Representational Derivation, in *Programs as Data Objects*, Springer-Verlag Lecture Notes, 1986.

[Wan86b] Mitchell Wand, The Mystery of the Tower Revealed: a Non-Reflective Description of the Reflective Tower, *Proceedings of the 1986 ACM Symposium on LISP and Functional Programming*, August 1986, 298-307.

[Wan86c] Mitchell Wand, Finding the Source of Type Errors, *Conference Record of the Thirteenth Annual Symposium on Principles of Programming Languages*, St. Petersburg, Fla., 1986, 38-43.

[WaF88] Mitchell Wand and Daniel P. Friedman, The Mystery of the Tower Revealed: A Non-Reflective Description of the Reflective Tower, in *Meta-Level Architectures and Reflection*, P. Maes and D. Nardi (ed.), Elsevier Sci. Publishers B.V. (North Holland), 1988, 111-134. Also to appear in Lisp and Symbolic Computation.