

## Readings In Scheme

Ozan S. Yigit

York University  
Computing Services  
4700 Keele St. North York  
Ontario, M3J 1P3  
Canada



(416) 736-5257

### Intro Blurb

This is the first edition of *Readings in Scheme*, an irregular column featuring a comprehensive bibliography on Scheme. The core of this bibliography was shamelessly snarfed from [56], and has been growing ever since, thanks to further additions by Dan Friedman, Ken Dickey and Mark MacLennan.

I would appreciate your contributions and corrections to keep this bibliography current and complete. I found that it is very difficult for one person to keep track of all of the additions to the literature on Scheme.

The machine readable copy of this bibliography is available in *bib [refer]* format, and if necessary, I can supply a converted database suitable for *BibTeX*.

If you have any corrections and/or additions or a request for the machine readable copy, e-mail to: [oz@yulibra.BITNET](mailto:oz@yulibra.BITNET) or [oz@yunexus.UUCP](mailto:oz@yunexus.UUCP). [[yunexus!oz@uunet.UU.NET](mailto:yunexus!oz@uunet.UU.NET) also appears to work.]

August 1988

### Bibliography

- [1] John Reynolds, Definitional Interpreters for Higher Order Programming Languages, *ACM Conference Proceedings*, 1972, 717-740.
- [2] Gerald Jay Sussman and Guy Lewis Steele, Jr., Scheme: an Interpreter for Extended Lambda Calculus, MIT Artificial Intelligence Memo 349, Cambridge, Mass., December 1975.
- [3] Guy Lewis Steele, Jr. and Gerald Jay Sussman, Lambda, the Ultimate Imperative, MIT Artificial Intelligence Memo 353, Cambridge, Mass., March 1976.
- [4] Guy Lewis Steele, Jr., Lambda, the Ultimate Declarative, MIT Artificial Intelligence Memo 379, Cambridge, Mass., November 1976.
- [5] 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.
- [6] 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.

- [7] Mitchell Wand, Continuation-Based Program Transformation Strategies, *Journal of the ACM* 27, 1 (1978), 174-180.
- [8] Mitchell Wand and Daniel P. Friedman, Compiling lambda expressions using continuations and factorizations, *Computer Languages* 3, 1978, 241-263.
- [9] Guy Lewis Steele, Jr. and Gerald Jay Sussman, The Revised Report on Scheme, a Dialect of Lisp, MIT Artificial Intelligence Memo 452, Cambridge, Mass., January 1978.
- [10] Guy Lewis Steele, Jr., Rabbit: a Compiler for Scheme, MIT Artificial Intelligence Laboratory Technical Report 474, Cambridge, Mass., May 1978.
- [11] Guy Lewis Steele, Jr. and Gerald Jay Sussman, The Art of the Interpreter, or the Modularity Complex (parts zero, one, and two), MIT Artificial Intelligence Memo 453, Cambridge, Mass., May 1978.
- [12] 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.
- [13] Steven S. Muchnick and Uwe F. Pleban, A Semantic Comparison of Lisp and Scheme, *Conference Record of the 1980 Lisp Conference*, 1980, 56-65.
- [14] 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 (editor), MIT Press, Cambridge, Mass., 1980.
- [15] 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.
- [16] Mitchell Wand, Continuation-Based Multiprocessing, *Conference Record of the 1980 Lisp Conference*, 1980, 19-28.
- [17] Guy Lewis Steele, Jr. and Gerald Jay Sussman, Design of a Lisp-based Processor, *Comm. of the ACM* 23, 11 (November 1980), 628-645.
- [18] 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.
- [19] 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.
- [20] Peter Henderson, Functional Geometry, *Conference Record of the 1982 ACM Symposium on Lisp and Functional Programming*, 1982, 179-187.
- [21] 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.
- [22] Gerald Jay Sussman, LISP, Programming and Implementation, in *Functional Programming and its Applications*, Darlington, Henderson, Turner (editor), Cambridge University Press, London, 1982.

- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] Daniel P. Friedman, Christopher T. Haynes and Eugene E. Kohlbecker, Programming with Continuations, in *Program Transformation and Programming Environments*, P. Pepper (editor), Springer-Verlag, 1984, 263-274.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] Jonathan A. Rees, Norman I. Adams and James R. Meehan, *The T manual, fourth edition*, Yale University Computer Science Department, January 1984.
- [32] Guillermo J. Rozas, *Liar, an Algol-like Compiler for Scheme*, S. B. Thesis, MIT Department of Electrical Engineering and Computer Science, January 1984.
- [33] *MIT Scheme Manual, Seventh Edition*, Department of Electrical Engineering and Computer Science, MIT, Cambridge, Mass., September 1984.
- [34] *MacScheme Reference Manual*, Semantic Microsystems, Sausalito, Calif., 1985.
- [35] Harold Abelson, Gerald Jay Sussman and Julie Sussman, *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge, Mass., 1985.
- [36] 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 (editor), Cambridge University Press, London, 1985, 237-250.
- [37] 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.
- [38] William Clinger, editor. The Revised Revised Report on Scheme, or An Uncommon Lisp, MIT Artificial Intelligence Memo 848, Cambridge, Mass., August 1985. Also published as Computer Science Department Technical Report

- 174, Indiana University, June 1985.
- [39] 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.
  - [40] 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.
  - [41] Peehong Chen and L. David Sabbagh, Scheme as an Interactive Graphics Programming Environment, Computer Science Technical Report No. 166, Indiana University, Bloomington, Indiana, March 1985.
  - [42] *TI Scheme Language Reference Manual*, Texas Instruments, Inc., November 1985. Preliminary version 1.0.
  - [43] Michael A. Eisenberg, Bochser: An Integrated Scheme Programming System, MIT Computer Science Technical Report 349, Cambridge, Mass., October 1985.
  - [44] Matthias Felleisen, Transliterating Prolog into Scheme, Computer Science Technical Report #182, Indiana University, Bloomington, Indiana, October 1985.
  - [45] 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.
  - [46] R. Kent Dybvig, Daniel P. Friedman and Christopher T. Haynes, Expansion-Passing style: Beyond Conventional Macros, *Proceedings of the 1986 ACM Conference on Lisp and Functional Programming*, 1986, 143-150.
  - [47] Marc Feeley and Guy LaPalme, Closure Generation based on viewing LAMBDA as EPSILON plus COMPILE, Submitted for Publication, 1986.
  - [48] 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.
  - [49] Daniel P. Friedman and Matthias Felleisen, *The Little LISPer: Second Edition*, Science Research Associates, Inc., Palo Alto, California, 1986.
  - [50] 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.
  - [51] Mitchell Wand, Finding the Source of Type Errors, *Conference Record of the Thirteenth Annual Symposium on Principles of Programming Languages*, St. Peterburg, Fla., 1986, 38-43.
  - [52] Mitchell Wand, From Interpreter to Compiler: A Representational Derivation, in *Programs as Data Objects*, Springer-Verlag Lecture Notes, 1986.
  - [53] 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.
  - [54] Eugene E. Kohlbecker, Syntactic Extensions in the Programming Language Lisp, Computer Science Technical Report #199 (PhD Dissertation), Indiana University,

- Bloomington, Indiana, August 1986.
- [55] 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*.
  - [56] 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.
  - [57] Revised<sup>3</sup> Report on the Algorithmic Language Scheme, *ACM Sigplan Notices* 21, 12 (December 1986).
  - [58] Christopher T. Haynes, Logic Continuations, *Proceedings of the Third International Conference on Logic Programming*, Jul 1986, 671-685.
  - [59] Matthias Felleisen, Daniel P. Friedman, Eugene E. Kohlbecker and Bruce Duba, Reasoning with Continuations, *Proceedings of the Symposium on Logic in Computer Science*, Washigton DC, June 1986, 131-141.
  - [60] 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.
  - [61] Marc Feeley, *Deux Approches a' L'implantation du Language Scheme*, M.Sc. Thesis, D'epartement d'Informatique et de Recherche Op'rationelle, University of Montreal, May 1986.
  - [62] R. Kent Dybvig, *The Scheme Programming Language*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.
  - [63] Marc Feeley and Guy LaPalme, Using Cloures for Code Generation, *Journal of Computer Languages* 12, 1 (1987), 47-66, Pergamon Press.
  - [64] Matthias Felleisen, Reflections on Landin's J-Operator: A Partly Historical Note, *Journal of Computer Languages* 12, 3/4 (1987), 197-207, Pergamon Press.
  - [65] 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.
  - [66] Matthias Felleisen, Daniel P. Friedman, Eugene E. Kohlbecker and Bruce Duba, A syntactic theory of sequential control, *Theoretical Computer Science* 52 (1987), 205-237.
  - [67] Daniel P. Friedman and Matthias Felleisen, *The Little LISPer*, MIT Press, 1987. Trade Edition.
  - [68] Christopher T. Haynes and Daniel P. Friedman, Abstracting Timed Preemption with Engines, *Journal of Computer Languages* 12, 2 (1987), 109-121, Pergamon Press.
  - [69] 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. (PhD Dissertation), Indiana University, Bloomington, Indiana, August 1987.

- [70] Matthias Felleisen, Daniel P. Friedman, Bruce Duba and John Merrill, Beyond Continuations, Computer Science Dept. Technical Report #216, Indiana University, Bloomington, Indiana, February, 1987.
- [71] 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.
- [72] Matthias Felleisen and Daniel P. Friedman, A Syntactic Theory of Sequential State, Computer Science Dept. Technical Report #230, Indiana University, Bloomington, Indiana, October 1987.
- [73] 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.
- [74] Michael Eisenberg, *Programming In Scheme*, Scientific Press, Redwood City, CA, 1988.
- [75] 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 (editor), Elsevier Sci. Publishers B.V. (North Holland), 1988, 111-134. Also to appear in *Lisp and Symbolic Computation*.
- [76] Daniel P. Friedman, Mitchell Wand, Christopher T. Haynes and Eugene E. Kohlbecker, *Programming Languages: Their Abstractions, Representations, and Implementations*, M.I.T. Press and McGraw Hill, 1988-1989. in progress.
- [77] George Springer and Daniel P. Friedman, An Introduction to the Art of Programming in Scheme, in progress, 1988-1989.
- [78] Harold Abelson and Gerald Jay Sussman, Lisp: A Language for Stratified Design, *BYTE*, February 1988, 207-218.
- [79] William Clinger, Semantics of Scheme, *BYTE*, February 1988, 221-227.
- [80] Alan Bawden and Jonathan Rees, Syntactic Closures, *Proceedings of the 1988 ACM Symposium on LISP and Functional Programming*, Salt Lake City, Utah., July 1988.
- [81] 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.
- [82] 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.