

## Readings In Scheme

*Ozan S. Yigit*

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



oz@nexus.yorku.ca

### Intro Blurb

In this edition of the *Readings in Scheme*, I present the entire Scheme bibliography (done once a year) as it currently stands, containing 97 entries. Unfortunately, no abstracts are included at this time, as they have not reached a critical mass.

As usual, I would appreciate your contributions and corrections to keep this bibliography current and complete. If you know of any old or new technical reports, articles, books on Scheme, do let me know.

The machine readable copy of this bibliography is available in *bib* [*refer*] format, and there is also a *BibTeX* version floating around.

If you have any corrections and/or additions or a request for the machine readable copy, e-mail to [oz@nexus.yorku.ca](mailto:oz@nexus.yorku.ca) or [oz@yulibra.BITNET](mailto:oz@yulibra.BITNET).

Happy scheming... January 1990.

### 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, *Journal of 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] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] Steven S. Muchnick and Uwe F. Pleban, A Semantic Comparison of Lisp and Scheme, *Conference Record of the 1980 Lisp Conference*, 1980, 56-65.
- [17] 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.
- [18] Mitchell Wand, Continuation-Based Multiprocessing, *Conference Record of the 1980 Lisp Conference*, 1980, 19-28.
- [19] Guy Lewis Steele Jr. and Gerald Jay Sussman, Design of a Lisp-based Processor, *Comm. of the ACM* 23, 11 (November 1980), 628-645.
- [20] Rex A. Dwyer and R. Kent Dybvig, A SCHEME for Distributed Processes, Computer Science Department Technical Report #107, Indiana University, Bloomington, Indiana, April 1981.
- [21] 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.
- [22] 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.
- [23] Peter Henderson, Functional Geometry, *Conference Record of the 1982 ACM Symposium on Lisp and Functional Programming*, 1982, 179-187.
- [24] 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.

- [25] Gerald Jay Sussman, LISP, Programming and Implementation, in *Functional Programming and its Applications*, Darlington, Henderson, Turner (editor), Cambridge University Press, London, 1982.
- [26] R. Kent Dybvig, C-Scheme, Computer Science Department Technical Report #149 (MS Thesis), Indiana University, Bloomington, Indiana, 1983.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] Jonathan A. Rees, Norman I. Adams and James R. Meehan, *The T manual, fourth edition*, Yale University Computer Science Department, January 1984.
- [36] Guillermo J. Rozas, *Liar, an Algol-like Compiler for Scheme*, S. B. Thesis, MIT Department of Electrical Engineering and Computer Science, January 1984.
- [37] Richard Schooler and James W. Stamos, Proposal For a Small Scheme Implementation, MIT Lab for Computer Science Memo TM-267, Cambridge, Mass., October 1984.
- [38] *MIT Scheme Manual, Seventh Edition*, Department of Electrical Engineering and Computer Science, MIT, Cambridge, Mass., September 1984.
- [39] *MacScheme Reference Manual*, Semantic Microsystems, Sausalito, Calif., 1985.
- [40] Harold Abelson, Gerald Jay Sussman and Julie Sussman, *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge, Mass., 1985.
- [41] 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.
- [42] 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.
  - [43] 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.
  - [44] 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.
  - [45] 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.
  - [46] 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.
  - [47] R. Kent Dybvig and Bruce T. Smith, *Chez Scheme Reference Manual Version 1.0*, Cadence Research Systems, Bloomington, Indiana, May 1985.
  - [48] *TI Scheme Language Reference Manual*, Texas Instruments, Inc., November 1985. Preliminary version 1.0.
  - [49] Michael A. Eisenberg, Bochser: An Integrated Scheme Programming System, MIT Computer Science Technical Report 349, Cambridge, Mass., October 1985.
  - [50] Matthias Felleisen, Transliterating Prolog into Scheme, Computer Science Technical Report #182, Indiana University, Bloomington, Indiana, October 1985.
  - [51] 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.
  - [52] 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.
  - [53] Marc Feeley and Guy LaPalme, Closure Generation based on viewing LAMBDA as EPSILON plus COMPILE, Submitted for Publication, 1986.
  - [54] 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.
  - [55] Daniel P. Friedman and Matthias Felleisen, *The Little LISPer: Second Edition*, Science Research Associates, Inc., Palo Alto, California, 1986.
  - [56] 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.
  - [57] 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.

- [58] Mitchell Wand, From Interpreter to Compiler: A Representational Derivation, in *Programs as Data Objects*, Springer-Verlag Lecture Notes, 1986.
- [59] 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.
- [60] Eugene E. Kohlbecker, Syntactic Extensions in the Programming Language Lisp, Computer Science Technical Report #199 (Ph.D. Dissertation), Indiana University, Bloomington, Indiana, August 1986.
- [61] 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*.
- [62] 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.
- [63] Revised Report on the Algorithmic Language Scheme, *ACM Sigplan Notices* 21, 12 (December 1986).
- [64] Christopher T. Haynes, Logic Continuations, *Proceedings of the Third International Conference on Logic Programming*, Jul 1986, 671-685.
- [65] 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.
- [66] 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.
- [67] 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.
- [68] R. Kent Dybvig, *The Scheme Programming Language*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.
- [69] Marc Feeley and Guy LaPalme, Using Cloures for Code Generation, *Journal of Computer Languages* 12, 1 (1987), 47-66, Pergamon Press.
- [70] Matthias Felleisen, Reflections on Landin's J-Operator: A Partly Historical Note, *Journal of Computer Languages* 12, 3/4 (1987), 197-207, Pergamon Press.
- [71] 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.
- [72] Matthias Felleisen, Daniel P. Friedman, Eugene E. Kohlbecker and Bruce Duba, A syntactic theory of sequential control, *Theoretical Computer Science* 52 (1987), 205-237.
- [73] Daniel P. Friedman and Matthias Felleisen, *The Little LISPer*, MIT Press, 1987. Trade Edition.
- [74] Christopher T. Haynes and Daniel P. Friedman, Abstracting Timed Preemption with Engines, *Journal of Computer Languages* 12, 2 (1987), 109-121, Pergamon Press.

- [75] 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.
- [76] 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.
- [77] Matthias Felleisen, Daniel P. Friedman, Bruce Duba and John Merrill, Beyond Continuations, Computer Science Dept. Technical Report #216, Indiana University, Bloomington, Indiana, February, 1987.
- [78] 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.
- [79] Matthias Felleisen and Daniel P. Friedman, A Syntactic Theory of Sequential State, Computer Science Dept. Technical Report #230, Indiana University, Bloomington, Indiana, October 1987.
- [80] 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.
- [81] Michael Eisenberg, *Programming In Scheme*, Scientific Press, Redwood City, CA, 1988.
- [82] David Kranz, Orbit: An optimizing compiler for Scheme, Computer Science Technical report #632 (Ph.D. Dissertation), Yale University, 1988.
- [83] 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*.
- [84] Daniel P. Friedman, Mitchell Wand, Christopher T. Haynes and Eugene E. Kohlbecker, in *Programming Languages: Their Abstractions, Representations, and Implementations*, MIT Press and McGraw-Hill, 1988-1989. in progress.
- [85] Harold Abelson and Gerald Jay Sussman, Lisp: A Language for Stratified Design, *BYTE*, February 1988, 207-218.
- [86] William Clinger, Semantics of Scheme, *BYTE*, February 1988, 221-227.
- [87] Alan Bawden and Jonathan Rees, Syntactic Closures, *Proceedings of the 1988 ACM Symposium on LISP and Functional Programming*, Salt Lake City, Utah., July 1988.
- [88] 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.
- [89] R. Kent Dybvig and Robert Hieb, Engines from Continuations, Computer Science Department Technical Report #254, Indiana University, Bloomington, Indiana, July 1988. Also to appear in *Journal of Computer Languages*.
- [90] R. Kent Dybvig and Robert Hieb, Continuations and Concurrency, Computer Science Department Technical Report #256, Indiana University, Bloomington, Indiana, July

1988.

- [91] 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.
- [92] 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.
- [93] 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.
- [94] 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.
- [95] George Springer and Daniel P. Friedman, in *Scheme and the Art of Programming*, MIT Press and McGraw-Hill, 1989.
- [96] Steven R. Vegdahl and Uwe F. Pleban, The Runtime Environment for Scream, 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.
- [97] Joel F. Bartlett, SCHEME->C a Portable Scheme-to-C Compiler, Research Report 89/1, Dec. Western Research Laboratory, Palo Alto, California, January 1989.

[ FROM A GUY WHO ONLY HAS SECOND-HAND KNOWLEDGE OF THE AN-C COMPILER. ]

TOP TEN THINGS OVERHEARD AT THE ANSI C DRAFT COMMITTEE MEETINGS:

- 10. Sorry, but that's too useful.
- 9. But little-endian systems *\*are\** more consistent!
- 8: I'm on the committee and I *\*still\** don't know what the hell #pragma is for.
- 7. Well, it's an excellent idea, but it would make the compilers too hard to write.
- 6. Them bats is smart; they use radar.
- 5. All right, who's the wiseguy who stuck this trigraph stuff in here?
- 4. How many times do we have to tell you, "No prior art!"
- 3. Ha, ha, they're actually going to adopt this sucker.
- 2. Thank you for your generous donation, Mr. Wirth.

AND NUMBER ONE, THE TOP THING HEARD AT THE ANSI C DRAFT COMMITTEE,

- 1. Gee, I wish we hadn't backed down on 'noalias'.

from Barry Margolin

## EDITORIAL POLICY

All submissions to Lisp Pointers, with the exception of technical articles, should be made in camera-ready text and sent to the appropriate department head. Technical articles may be submitted to the Technical Articles Editor in either hard copy or in TEX source files by Arpanet link, tar format cartridge tape, or tar format reel-to-reel. All submissions should be single-spaced with no page numbers. Without a special waiver from the appropriate department head, submissions will be limited to ten pages. This can be achieved by printing longer articles two-up. Camera-ready text is defined to be no more than 7 1/2 x 10 inches or 19 x 25 centimeters, centered on an 8 1/2 x 11 inch page. Articles that contain too much blank space will be rejected. It is the author's responsibility to retain a working copy of the submission, as contributions will not be returned to authors. Authors not fluent in writing English are requested to have their work reviewed and corrected for style and syntax prior to submission.

Although Lisp Pointers is not refereed, acceptance is subject to the discretion of the appropriate department head. The scope of topics for Lisp Pointers includes all dialects of Lisp and Scheme. We encourage research articles, tutorials, and summarizations of discussions in other forums. Lisp Pointers is not a forum for detailed discussions on proposed changes to the Common Lisp standard. Lisp Pointers is a Special Interest Publication of the Special Interest Group on Programming Languages (SIGPLAN). A subscription to LISP Pointers does not include membership in any group.

Note: ACM Members who expect to renew their membership within the next six months should send no LISP Pointers subscription payment now as they will be billed on their renewal notice. Those expecting to renew in seven or more months should send on half the annual LISP Pointers subscription payment now.

\_\_\_\_\_  
Name (Please print or type)

\_\_\_\_\_  
Mailing Address

\_\_\_\_\_  
City State Zip

\_\_\_\_\_  
Signature

Annual subscription rates are  
\$8 for ACM Members,  
\$4 for ACM Student Members,  
\$18 for Non-ACM Members.

- ACM MEMBER**  
ACM Member No. \_\_\_\_\_  
(see note regarding dues payment)
- ACM STUDENT MEMBER**  
ACM Student Member No. \_\_\_\_  
(see note regarding dues payment)
- NON-ACM MEMBER**  
Enclosed is annual subscription  
payment of \$18
- Please send information on ACM  
Membership.

New address. Please change my ACM record.

Please make checks payable to ACM Inc. and mail to: ACM, Inc., P.O. Box 12115, Church Street Station, New York, N.Y. 10249.