

# Bibliography

- [85M85] *Macscheme Reference Manual*. Semantic Microsystems, Sausalito, California, 1985.
- [All78] John Allen. *Anatomy of Lisp*. Computer Science Series. McGraw-Hill, 1978.
- [ALQ95] Sophie Anglade, Jean-Jacques Lacrampe, and Christian Queinnec. Semantics of combinations in scheme. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 7(4):15–20, October–December 1995.
- [App87] Andrew Appel. Garbage Collection can be faster than Stack Allocation. *Information Processing Letters*, 25(4):275–279, June 1987.
- [App92a] Andrew Appel. *Compiling with continuations*. Cambridge Press, 1992.
- [App92b] Apple Computer, Eastern Research and Technology. *Dylan, An object-oriented dynamic language*. Apple Computer, Inc., April 1992.
- [AR88] Norman Adams and Jonathan Rees. Object-oriented programming in Scheme. In *Conference Record of the 1988 ACM Conference on Lisp and Functional Programming*, pages 277–288, August 1988.
- [AS85] Harold Abelson and Gerald Jay with Julie Sussman Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, Mass., 1985.
- [AS94] Andrew W Appel and Zhong Shao. An empirical and analytic study of stack vs heap cost for languages with closures. Technical Report CS-TR-450-94, Princeton University, Department of Computer Science, Princeton (New-Jersey USA), March 1994.
- [Att95] Giuseppe Attardi. The embeddable common Lisp. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 8(1):30–41, January–April 1995. LUV-94, Fourth International LISP Users and Vendors Conference.
- [Bak] Henry G Baker. Cons should not cons its arguments, part ii: Cheney on the m.t.a. <ftp://ftp.netcom.com/pub/hb/hbaker/CheneyMTA.ps>.
- [Bak78] Henry G Baker. List processing in real time on a serial computer. *Communications of the ACM*, 21(4):280–294, April 1978.
- [Bak92a] Henry G Baker. The buried binding and dead binding problems of Lisp 1.5. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 5(2):11–19, April–June 1992.
- [Bak92b] Henry G Baker. Inlining semantics for subroutines which are recursive. *SIGPLAN Notices*, 27(12):39–46, December 1992.
- [Bak92c] Henry G Baker. Metacircular semantics for common Lisp special forms. *Lisp Pointers*, 5(4):11–20, 1992.

- [Bak93] Henry G Baker. Equal rights for functional objects or, the more things change, the more they are the same. *OOPS Messenger*, 4(4):2-27, October 1993.
- [Bar84] H P Barendregt. *The Lambda Calculus*. Number 103 in Studies in Logic. North Holland, Amsterdam, 1984.
- [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.
- [Baw88] Alan Bawden. Reification without evaluation. In *Conference Record of the 1988 ACM Symposium on LISP and Functional Programming*, pages 342-351, Snowbird, Utah, July 1988. ACM Press.
- [BC87] Jean-Pierre Briot and Pierre Cointe. A uniform model for object-oriented languages using the class abstraction. In *IJCAI '87*, pages 40-43, 1987.
- [BCSJ86] Jean-Pierre Briot, Pierre Cointe, and Emmanuel Saint-James. Réécriture et récursion dans une fermeture, étude dans un Lisp à liaison superficielle et application aux objets. In Pierre Cointe and Jean Bézivin, editors, 3<sup>èmes</sup> journées LOO/AF CET, number 48 in Revue Bigre+Globule, pages 90-100, IRCAM, Paris (France), January 1986.
- [BDG<sup>+</sup>88] Daniel G. Bobrow, Linda G. DeMichiel, Richard P. Gabriel, Sonya E. Keene, Gregor Kiczales, and David A. Moon. Common Lisp object system specification. *SIGPLAN Notices*, 23, September 1988. special issue.
- [Bel73] James R Bell. Threaded code. *Communications of the ACM*, 16(6):370-372, June 1973.
- [Bet91] David Michael Betz. *XSCHEME: An Object-oriented Scheme*. P.O. Box 144, Peterborough, NH 03458 (USA), July 1991. version 0.28.
- [BG94] Henri E Bal and Dick Grune. *Programming Language Essentials*. Addison Wesley, 1994.
- [BHY87] Adrienne Bloss, Paul Hudak, and Jonathan Young. Code optimizations for lazy evaluation. *International journal on Lisp and Symbolic Computation*, 1(2):147-164, 1987.
- [BJ86] David H Bartley and John C Jensen. The implementation of pc Scheme. In *Conference Record of the 1986 ACM Symposium on LISP and Functional Programming*, pages 86-93, Cambridge, Massachusetts, August 1986. ACM Press.
- [BKK<sup>+</sup>86] D G Bobrow, K Kahn, G Kiczales, L Masinter, M Stefik, and F Zdybel. Commonloops: Merging Lisp and object oriented programming. In *OOPSLA '86 — Object-Oriented Programming Systems and Languages*, pages 17-29, Portland (Oregon, USA), 1986.
- [BM82] Robert S. Boyer and J. Strother Moore. A mechanical proof of the unsolvability of the halting problem. Technical Report ICSCA-CMP-28, July 1982.
- [BR88] Alan Bawden and Jonathan Rees. Syntactic closures. In *Proceedings of the 1988 ACM Symposium on LISP and Functional Programming*, Salt Lake City, Utah., July 1988.
- [BW88] Hans J Boehm and M Weiser. Garbage collection in an uncooperative environment. *Software — Practice and Experience*, 18(9), September 1988.
- [Car93] Luca Cardelli. An implementation of F<sub>λ</sub>. Research Report 97, DEC-SRC, February 1993.
- [Car94] George J. Carrette. *Siod, Scheme In One Defun*. PARADIGM Associates Incorporated, 29 Putnam Ave, Suite 6, Cambridge, MA 02138, USA, 1994.

- [Cay83] Michel Cayrol. *Le Langage Lisp*. Cepadues Editions, Toulouse (France), 1983.
- [CC77] P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL'77 — 4th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, CA, January 1977. ACM Press.
- [CDD<sup>+</sup>91] J. Chailloux, M. Devin, F. Dupont, J.-M. Hullot, B. Serpette, and J. Vuillemin. *Le-Lisp version 15.24, le manuel de référence*. INRIA, May 1991.
- [CEK<sup>+</sup>89] L W Cannon, R A Elliott, L W Kirchhoff, J H Miller, J M Milner, R W Mitze, E P Schan, N O Whittington, H Spencer, and D Keppel. *Recommended C Style and Coding Standards*, November 1989.
- [CG77] Douglas W. Clark and C. Cordell Green. An empirical study of list structure in LISP. *Communications of the ACM*, 20(2):78–87, February 1977.
- [CH94] William D Clinger and Lars Thomas Hansen. Lambda, the ultimate label, or a simple optimizing compiler for Scheme. In *Proceedings of the 1994 ACM Conference on Lisp and Functional Programming [lfp94]*, pages 128–139.
- [Cha80] Jérôme Chailloux. *Le modèle VLISP : description, implémentation et évaluation*. Thèse de troisième cycle, Université de Vincennes, April 1980. Rapport LITP 80-20, Paris (France).
- [Cha94] Gregory J Chaitin. The limits of mathematics. IBM PO Box 704, Yorktown Heights, NY 10598 (USA), July 1994.
- [CHO88] William D. Clinger, Anne H. Hartheimer, and Eric M. Ost. Implementation strategies for continuations. In *Conference Record of the 1988 ACM Conference on Lisp and Functional Programming*, pages 124–131, August 1988.
- [Chr95] Christophe. *La Famille Fenouillard*. Armand Colin, 1895.
- [Cla79] Douglas W. Clark. Measurements of dynamic list structure use in Lisp. *IEEE Transactions on Software Engineering*, 5(1):51–59, January 1979.
- [Cli84] William Clinger. The Scheme 311 compiler: an exercise in denotational semantics. In *Conference Record of the 1984 ACM Symposium on Lisp and Functional Programming*, pages 356–364, 1984.
- [Coi87] Pierre Cointe. The ObjVlisp kernel: a reflexive architecture to define a uniform object oriented system. In P. Maes and D. Nardi, editors, *Workshop on MetaLevel Architectures and Reflection*, Alghiero, Sardinia (Italy), October 1987. North Holland.
- [Com84] Douglas Comer. *Operating System Design: The XINU Approach*. Prentice-Hall, 1984.
- [CR91a] William Clinger and Jonathan Rees. Macros that work. In *POPL '91 – Eighteenth Annual ACM symposium on Principles of Programming Languages*, pages 155–162, Orlando, (Florida USA), January 1991.
- [CR91b] William Clinger and Jonathan A Rees. The revised<sup>4</sup> report on the algorithmic language Scheme. *Lisp Pointer*, 4(3), 1991.
- [Cur89] Pavel Curtis. (algorithms). *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 3(1):48–61, July 1989.
- [Dan87] Olivier Danvy. Memory allocation and higher-order functions. In *Proceedings of the ACM SIGPLAN'87 Symposium on Interpreters and Interpretive Techniques*, SIGPLAN Notices, Vol. 22, No 7, pages 241–252, Saint-Paul, Minnesota, June 1987. ACM, ACM Press.

- [Del89] Vincent Delacour. *Piccolo espresso*. *Revue Bigre+Globule*, (65):30-42, July 1989.
- [Deu80] L Peter Deutsch. *Bytelisp and its alto implementation*. In *Conference Record of the 1980 LISP Conference* [lfp80], pages 231-242.
- [Deu89] Alain Deutsch. *Génération automatique d'interpréteurs et compilation à partir de spécifications dénotationnelles*. Rapport de DEA-LAP 1988 LITP-RXF 89-17, LITP, January 1989.
- [Dev85] Matthieu Devin. *Le portage du système Le-Lisp*. Rapport technique 50, INRIA-Rocquencourt, May 1985.
- [DF90] Olivier Danvy and Andrzej Filinski. *Abstracting control*. In *LFP '90 - ACM Symposium on Lisp and Functional Programming*, pages 151-160, Nice (France), June 1990.
- [DFH86] R. Kent Dybvig, Daniel P. Friedman, and Christopher T. Haynes. *Expansion-passing style: Beyond conventional macros*. In *Conference Record of the 1986 ACM Conference on Lisp and Functional Programming*, pages 143-150, 1986.
- [DFH88] R. Kent Dybvig, Daniel P. Friedman, and Christopher T. Haynes. *Expansion-passing style: a general macro mechanism*. *International journal on Lisp and Symbolic Computation*, 1(1):53-76, June 1988.
- [DH92] Olivier Danvy and John Hatcliff. *Thunks (continued)*. In *Proceedings of the Workshop on Static Analysis WSA '92*, volume 81-82 of *Bigre Journal*, pages 3-11, Bordeaux, France, September 1992. IRISA, Rennes, France. Extended version available as Technical Report CIS-92-28, Kansas State University.
- [DHB93] R. Kent Dybvig, Robert Hieb, and Carl Bruggeman. *Syntactic abstraction in Scheme*. *International journal on Lisp and Symbolic Computation*, 5(4):295-326, 1993.
- [DHHM92] R Ducournau, M Habib, M Huchard, and M L Mugnier. *Monotonic conflict resolution mechanisms for inheritance*. In *OOPSLA '92 - Object-Oriented Programming Systems and Languages*, pages 16-4, 1992.
- [Dil88] Antoni Diller. *Compiling Functional Languages*. John Wiley and sons, 1988.
- [DM88] Olivier Danvy and Karoline Malmkjær. *A Blond primer*. DIKU report 88/21, University of Copenhagen, Copenhagen, Denmark, 1988.
- [dM95] Antoine Dumesnil de Maricourt. *Macro-expansion en Lisp, sémantique et réalisation*. Thèse d'université, Université Paris 7, Paris (France), June 1995.
- [DPS94a] Harley Davis, Pierre Parquier, and Nitsan Séniak. *Sweet harmony: the talk/c++ connection*. In *Proceedings of the 1994 ACM Conference on Lisp and Functional Programming* [lfp94], pages 121-127.
- [DPS94b] Harley Davis, Pierre Parquier, and Nitsan Séniak. *Talking about modules and delivery*. In *Proceedings of the 1994 ACM Conference on Lisp and Functional Programming* [lfp94], pages 113-120.
- [dR87] Jim des Rivières. *Control-related meta-level facilities in Lisp*. In P. Maes and D. Nardi, editors, *Workshop on Meta-Level Architecture and Reflection*, Alghiero, Sardinia (Italy), October 1987. North Holland.
- [dRS84] Jim des Rivières and Brian Cantwell Smith. *The implementation of procedurally reflective languages*. In *Conference Record of the 1984 ACM Symposium on LISP and Functional Programming*, pages 331-347, Austin, Texas, August 1984. ACM Press.

- [Dyb87] R. Kent Dybvig. *The Scheme Programming Language*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.
- [ES70] Jay Earley and Howard Sturgis. A formalism for translator interactions. *Communications of the ACM*, 13(10):607-617, October 1970.
- [Fel88] Matthias Felleisen. The theory and practice of first-class prompts. In *POPL '88 - Fifteenth Annual ACM symposium on Principles of Programming Languages*, pages 180-190, San Diego (California USA), January 1988.
- [Fel90] Matthias Felleisen. On the expressive power of programming languages. In Neil Jones, editor, *ESOP '90 - European Symposium on Programming*, volume 432 of *Lecture Notes in Computer Science*, pages 134-151, Copenhagen (Denmark), 1990. Springer-Verlag.
- [FF87] Matthias Felleisen and Daniel P. Friedman. A reduction semantics for imperative higher-order languages. *Parallel Architectures and Languages Europe*, 259:206-223, 1987.
- [FF89] Matthias Felleisen and Daniel P. Friedman. A syntactic theory of sequential state. *Theoretical Computer Science*, 69(3):243-287, 1989. Preliminary version in: *Proc. 14th ACM Symposium on Principles of Programming Languages*, 1987, 314-325.
- [FFDM87] Matthias Felleisen, Daniel P. Friedman, Bruce Duba, and John Merrill. Beyond continuations. Computer Science Dept. Technical Report 216, Indiana University, Bloomington, Indiana, February 1987.
- [FH89] 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.
- [FL87] Marc Feeley and Guy LaPalme. Using closures for code generation. *Journal of Computer Languages*, 12(1):47-66, 1987.
- [FM90] Marc Feeley and James S. Miller. A parallel virtual machine for efficient Scheme compilation. In *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.
- [FW76] D.P. Friedman and D.S. Wise. Cons should not evaluate its arguments. In Michaelson and Milner, editors, *Proceedings of the 3rd International Colloquium on "Automata, Languages and Programming"*, pages 257-284. Edinburgh University Press, July 1976.
- [FW84] Daniel P. Friedman and Mitchell Wand. Reification: Reflection without metaphysics. In *Conference Record of the 1984 ACM Symposium on LISP and Functional Programming*, pages 348-355, Austin, TX., August 1984.
- [FWFD88] Matthias Felleisen, Mitchell Wand, Daniel P. Friedman, and Bruce Duba. Abstract continuations: a mathematical semantics for handling functional jumps. In *Proceedings of the 1988 ACM Symposium on LISP and Functional Programming*, Salt Lake City, Utah., July 1988.
- [FWH92] Daniel P. Friedman, Mitchell Wand, and Christopher Haynes. *Essentials of Programming Languages*. MIT Press, Cambridge MA and McGraw-Hill, 1992.
- [Gab88] Richard P. Gabriel. The why of y. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 2(2):15-25, 1988.
- [GBM82] Martin L. Griss, Eric Benson, and Gerald Q. Maguire. Psl: A portable LISP system. In *Conference Record of the 1982 ACM Symposium on LISP and Functional Programming*, pages 88-97, Pittsburgh, Pennsylvania, August 1982. ACM Press.

- [Gor75] Michael J C Gordon. Operational reasoning and denotational semantics. In G. Huet and G. Kahn, editors, *Actes du Colloque IRIA "Constructions et Justifications de Programmes"*, pages 83–98, Arc et Senans, July 1975.
- [Gor88] Michael J C Gordon. *Programming Language Theory and its Implementation*. International Series in Computer Science. Prentice-Hall, 1988.
- [GP88] Richard P Gabriel and Kent M Pitman. Technical issues of separation in function cells and value cells. *International journal on Lisp and Symbolic Computation*, 1(1):81–101, June 1988.
- [GR83] Adèle Goldberg and David Robson. *Smalltalk-80, The Language and its Implementation*. Addison Wesley, 1983.
- [Gra93] Paul Graham. *On Lisp, Advanced Techniques for Common Lisp*. Prentice-Hall, 1993.
- [Gre77] Patrick Greussay. *Contribution à la définition interprétative et à l'implémentation des Lambda-langages*. Thèse d'état, Université Paris VI, November 1977. Rapport LITP 78-2.
- [Gud93] David Gudeman. Representing type information in dynamically typed languages. Technical Report 93-27, Department of Computer Science, University of Arizona, Tucson, AZ 85721, USA, October 1993.
- [Han90] Chris Hanson. Efficient stack allocation for tail-recursive languages. In *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.
- [HD90] Robert Hieb and R. Kent Dybvig. Continuations and concurrency. In *PPOPP '90 - ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming*, pages 128–136, Seattle (Washington US), March 1990.
- [HDB90] Robert Hieb, R. Kent Dybvig, and Carl Bruggeman. Representing control in the presence of first-class continuations. In *Proceedings of the SIGPLAN '90 Conference on Programming Language Design and Implementation*, pages 66–77, White Plains, New York, June 1990.
- [Hen80] Peter Henderson. *Functional Programming, Application and Implementation*. International Series in Computer Science. Prentice-Hall, 1980.
- [Hen92a] Fritz Henglein. Global tagging optimization by type inference. In *Proceedings of the 1992 ACM Conference on Lisp and Functional Programming*, pages 205–215, San Francisco, USA, June 1992.
- [Hen92b] Wade Hennessey. Wcl: Delivering efficient COMMON LISP applications under UN\*X. In *Conference Record of the 1992 ACM Symposium on LISP and Functional Programming*, pages 260–269, San Francisco, California, June 1992. ACM Press.
- [HFW84] Christopher T. Haynes, Daniel P. Friedman, and Mitchell Wand. Continuations and coroutines. In *Conference Record of the 1984 ACM Symposium on Lisp and Functional Programming*, pages 293–298, Austin, TX., 1984.
- [Hof93] Ulrich Hoffmann. Using c as target code for translating highlevel programming languages. Technical Report APPLY/CAU/IV.4/2, Christian-Albrechts-University of Kiel, 1993.
- [Hon93] P. Joseph Hong. Threaded code designs for forth interpreters. *SIG FORTH, Newsletter of the ACM's Special Interest Group on FORTH*, 4(2):11–18, fall 1993.

- [HS75] Carl Hewitt and Brian Smith. Towards a programming apprentice. *IEEE Transactions on Software Engineering*, 1(1):26–45, March 1975.
- [HS91] Samuel P Harbison and Guy L Steele, Jr. *C: A Reference Manual*. Prentice-Hall, 1991.
- [IEE91] IEEE Std 1178-1990. *IEEE Standard for the Scheme Programming Language*. Institute of Electrical and Electronic Engineers, Inc., New York, NY, 1991.
- [ILO94] ILOG, 2 Avenue Galliéni, BP 85, 94253 Gentilly, France. *Ilog-Talk Reference Manual*, 1994.
- [IM89] Takayasu Ito and Manabu Matsui. A parallel lisp language PaiLisp and its kernel specification. In Takayasu Ito and Robert H Halstead, Jr., editors, *Proceedings of the US/Japan Workshop on Parallel Lisp*, volume Lecture Notes in Computer Science 441, pages 58–100, Sendai (Japan), June 1989. Springer-Verlag.
- [IMY92] Yuuji Ichisugi, Satoshi Matsuoka, and Akinori Yonezawa. RbCL: A reflective object-oriented concurrent language without a run-time kernel. In Yonezawa and Smith [YS92], pages 24–35.
- [ISO90] ISO/IEC 9899:1990. Programming language — c. Technical report, Information Technology, 1990.
- [ISO94] ISO-IEC/JTC1/SC22/WG16. Programming language ISLISP, CD 13816. Technical report, ISO-IEC/JTC1/SC22/WG16, 1994.
- [ITW86] Takayasu Ito, Takashi Tamura, and Shin-ichi Wada. Theoretical comparisons of interpreted/compiled executions of Lisp on sequential and parallel machine models. In H J Kugler, editor, *Proceedings of the IFIP 10th World Computer Congress*, Dublin (Ireland), September 1986. North Holland.
- [Jaf94] Aubrey Jaffer. *Reference Manual for scm*, 1994.
- [JF92] Stanley Jefferson and Daniel P Friedman. A simple reflective interpreter. In Yonezawa and Smith [YS92], pages 48–58.
- [JGS93] Neil D Jones, C Karsten Gomard, and Peter Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall International, 1993. With chapters by L.O. Andersen and T. Mogensen.
- [Kah87] Gilles Kahn. Natural semantics. In *Proc. of STACS 1987, Lecture Notes in Computer Science*, chapter 247. Springer-Verlag, March 1987.
- [Kam90] Samuel Kamin. *Programming Languages: an Interpreter-Based Approach*. Addison-Wesley, Reading, Mass., 1990.
- [KdRB92] Gregor Kiczales, Jim des Rivières, and Daniel G Bobrow. *The Art of the Metaobject Protocol*. MIT Press, Cambridge MA, 1992.
- [Kes88] Robert R. Kessler. *Lisp, Objects, and Symbolic Programming*. Scott, Foreman/Little, Brown College Division, Glenview, Illinois, 1988.
- [KFFD86] Eugene E. Kohlbecker, Daniel P. Friedman, Matthias Felleisen, and Bruce Duba. Hygienic macro expansion. *Symposium on LISP and Functional Programming*, pages 151–161, August 1986.
- [KH89] Richard Kelsey and Paul Hudak. Realistic compilation by program transformation. In *POPL'89 — 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 281–292, Austin (Texas, USA), January 1989. ACM Press.

- [Knu84] Donald Ervin Knuth. *The T<sub>E</sub>X Book*. Addison Wesley, 1984.
- [KR78] Brian W Kernighan and Dennis Ritchie. *The C Programming Language*. Prentice-Hall, Englewood Cliffs, New Jersey, 1978.
- [KR90] Gregor Kiczales and Luis Rodriguez. Efficient method dispatch in pcl. In *LFP '90 - ACM Symposium on Lisp and Functional Programming*, pages 99-1052, Nice (France), June 1990.
- [KW90] Morry Katz and Daniel Weise. Continuing into the future: on the interaction of futures and first-class continuations. In *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming*, Nice, France, June 1990.
- [Lak80] Fred H. Lakin. Computing with text-graphic forms. In *Conference Record of the 1980 Lisp Conference*, pages 100-105, 1980.
- [Lan65] P J Landin. A Correspondence between algol 60 and Church's Lambda-notation. *Communications of the ACM*, 8:89-101 and 158-165, February 1965.
- [Leb05] Maurice Leblanc. *813*. Éditions Pierre Lafitte, 1905.
- [LF88] Henry Lieberman and Christopher Fry. Common eval. *Lisp Pointers*, 2(1):23-33, July 1988.
- [LF93] Shinn-Der Lee and Daniel P Friedman. Quasi-static scoping: Sharing variable bindings across multiple lexical scopes. In *POPL '93 - Twentieth Annual ACM symposium on Principles of Programming Languages*, pages 479-492, Charleston (South Carolina, USA), January 1993. ACM Press.
- [lfp80] *Conference Record of the 1980 LISP Conference*, Stanford (California USA), August 1980. The LISP Conference.
- [lfp94] *Proceedings of the 1994 ACM Conference on Lisp and Functional Programming*, Orlando (Florida USA), June 1994. ACM Press.
- [Lie87] Henry Lieberman. Reversible object-oriented interpreters. In *ECOOP '87 - European Conference on Object-Oriented Programming*, volume Special issue of Bigre 54, pages 13-22, Paris (France), June 1987. AFCET.
- [LLSt93] Bil Lewis, Dan LaLiberte, Richard Stallman, and the GNU Manual Group. Gnu emacs Lisp reference manual. Technical report, Free Software Foundation, 675 Massachusetts Avenue, Cambridge MA 02139 USA, Edition 2.0 1993.
- [LP86] Kevin J. Lang and Barak A. Pearlmutter. Oaklisp: an object-oriented Scheme with first class types. In *ACM Conference on Object-Oriented Systems, Programming, Languages and Applications*, pages 30-37, September 1986.
- [LP88] Kevin J. Lang and Barak A. Pearlmutter. Oaklisp: an object-oriented dialect of Scheme. *International journal on Lisp and Symbolic Computation*, 1(1):39-51, May 1988.
- [LW93] Xavier Leroy and Pierre Weis. *Manuel de référence du langage Caml*. InterÉditions, 1993.
- [MAE<sup>+</sup>62] John McCarthy, Paul W. Abrahams, Daniel J. Edwards, Timothy P. Hart, and Michael I. Levin. Lisp 1.5 programmer's manual. Technical report, MIT Press, Cambridge, MA (USA), 1962.
- [Man74] Zohar Manna. *Mathematical Theory of Computation*. Computer Science Series. McGraw-Hill, 1974.
- [Mas86] Ian A Mason. *The Semantics of Destructive Lisp*. CSLI Lecture Notes, Leland Stanford Junior University, California USA, 1986.



- [Mat92] Luis Mateu. Efficient implementation of coroutines. In Yves Bekkers and Jacques Cohen, editors, *International Workshop on Memory Management*, number 637 in Lecture Notes in Computer Science, pages 230–247, Saint-Malo (France), September 1992. Springer-Verlag.
- [MB93] Luis Mateu-Brule. *Stratégies avancées de gestion de blocs de contrôle*. Thèse de doctorat d'université, Université Pierre et Marie Curie (Paris 6), Paris (France), February 1993.
- [McC60] John McCarthy. Recursive functions of symbolic expressions and their computation by machine – part i. *Communications of the ACM*, 3(1):184–195, 1960.
- [McC78a] John McCarthy. Lisp history. In *Proc. SIGPLAN History of Programming Languages Conference*, pages 217–223, 1978. Also Sigplan Notices 13(8).
- [McC78b] John McCarthy. A micro-manual for Lisp - not the whole truth. In *Proc. SIGPLAN History of Programming Languages Conference*, pages 215–216, 1978. Also Sigplan Notices 13(8).
- [McD93] Raymond C McDowell. The relatedness and comparative utility of various approaches to operational semantics. Technical Report MS-CIS-93-16, LINC LAB 246, University of Pennsylvania, February 1993.
- [MNC<sup>+</sup>89] Gérald Masini, Amedeo Napoli, Dominique Colnet, Daniel Léonard, and Karl Tombre. *Les langages à objets*. InterÉditions, 1989.
- [Mor92] Luc Moreau. An operational semantics for a parallel functional language with continuations. In D. Etiemble and J-C. Syre, editors, *PARLE '92 – Parallel Architectures and Languages Europe*, pages 415–430, Paris (France), June 1992. Lecture Notes in Computer Science 605, Springer-Verlag.
- [Mor94] Luc Moreau. *Sound Evaluation of Parallel Functional Programs with First-Class Continuations*. Thèse de docteur en sciences appliquées, Université de Liège (Belgique), avril 1994.
- [Mos70] Joel Moses. The function of FUNCTION in LISP. *SIGSAM Bulletin*, 15:13–27, July 1970.
- [Moz87] Wolfgang A Mozart. Don Giovanni, 1787.
- [MP80] Steven S. Muchnick and Uwe F. Pleban. A semantic comparison of lisp and Scheme. In *Conference Record of the 1980 Lisp Conference*, pages 56–65. The Lisp Conference, P.O. Box 487, Redwood Estates CA., 1980.
- [MQ94] Luc Moreau and Christian Queindec. Partial continuations as the difference of continuations, a duumvirate of control operators. In Manuel Hermenegildo and Jaan Penjam, editors, *Lecture Notes in Computer Science 844*, pages 182–197, Madrid (Spain), September 1994. International Workshop PLILP '94 – Programming Language Implementation and Logic Programming, Springer-Verlag.
- [MR91] James S Miller and Guillermo J Rozas. Free variables and first-class environments. *International journal on Lisp and Symbolic Computation*, 4(2):107–141, 1991.
- [MS80] F Lockwood Morris and Jerald S Schwarz. Computing cyclic list structures. In *Conference Record of the 1980 Lisp Conference*, pages 144–153. The Lisp Conference, 1980.

- [Mul92] Robert Muller. M-lisp: A representation-independent dialect of lisp with reduction semantics. *ACM Transactions on Programming Languages and Systems*, 14(4):589-615, October 1992.
- [Nei84] Eugen Neidl. *Étude des relations avec l'interprète dans la compilation de Lisp*. Thèse de troisième cycle, Université Paris 6, Paris (France), 1984.
- [Nor72] Eric Norman. 1100 Lisp reference manual. Technical report, University of Wisconsin, 1972.
- [NQ89] Greg Nuyens and Christian Queinnec. Identifier Semantics: a Matter of References. Technical Report LIX RR 89 02, 67-80, Laboratoire d'Informatique de l'École Polytechnique, May 1989.
- [PE92] Julian Padget and Greg Nuyens (Editors). The eulisp definition. Technical report, University of Bath, 1992.
- [Per79] Jean-François Perrot. Lisp et  $\lambda$ -calcul. In Bernard Robinet (ed), editor,  *$\lambda$ -calcul et sémantique formelle des langages de programmation*, La Châtre (France), 1979. AFCET-GROPLAN, LITP-ENSTA, Paris.
- [Pit80] Kent Pitman. Special forms in Lisp. In *Conference Record of the 1980 LISP Conference* [lfp80], pages 179-187.
- [PJ87] Simon L. Peyton-Jones. *The Implementation of Functional Programming Languages*. International Series in Computer Science. Prentice-Hall, 1987.
- [PNB93] Padget, J.A., Nuyens, G., and Bretthauer, H. An overview of EuLisp. *Lisp and Symbolic Computation*, 6(1/2):9-98, 1993.
- [QC88] Christian Queinnec and Pierre Cointe. An open-ended Data Representation Model for Eu-Lisp. In *LFP '88 - ACM Symposium on Lisp and Functional Programming*, pages 298-308, Snowbird (Utah, USA), 1988.
- [QD93] Christian Queinnec and David De Roure. Design of a concurrent and distributed language. In Robert H Halstead Jr and Takayasu Ito, editors, *Parallel Symbolic Computing: Languages, Systems, and Applications, (US/Japan Workshop Proceedings)*, volume Lecture Notes in Computer Science 748, pages 234-259, Boston (Massachusetts USA), October 1993.
- [QD96] Christian Queinnec and David De Roure. Sharing code through first-class environments. In *Proceedings of ICFP'96 - ACM SIGPLAN International Conference on Functional Programming*, pages ??-??, Philadelphia (Pennsylvania, USA), May 1996.
- [QG92] Christian Queinnec and Jean-Marie Geffroy. Partial evaluation applied to symbolic pattern matching with intelligent backtrack. In M Billaud, P Castéran, MM Corsini, K Musumbu, and A Rauzy, editors, *WSA '92-Workshop on Static Analysis*, number 81-82 in Revue Bigre+Globule, pages 109-117, Bordeaux (France), September 1992.
- [QP90] Christian Queinnec and Julian Padget. A deterministic model for modules and macros. Bath Computing Group Technical Report 90-36, University of Bath, Bath (UK), 1990.
- [QP91a] Christian Queinnec and Julian Padget. A proposal for a modular Lisp with macros and dynamic evaluation. In *Journées de Travail sur l'Analyse Statique en Programmation Équationnelle, Fonctionnelle et Logique*, pages 1-8, Bordeaux (France), October 1991. Revue Bigre+Globule 74.

- [QP91b] Christian Queinnec and Julian Padget. Modules, macros and Lisp. In *Eleventh International Conference of the Chilean Computer Science Society*, pages 111–123, Santiago (Chile), October 1991. Plenum Publishing Corporation, New York NY (USA).
- [QS91] Christian Queinnec and Bernard Serpette. A Dynamic Extent Control Operator for Partial Continuations. In *POPL '91 – Eighteenth Annual ACM symposium on Principles of Programming Languages*, pages 174–184, Orlando (Florida USA), January 1991.
- [Que82] Christian Queinnec. *Langage d'un autre type : Lisp*. Eyrolles, Paris (France), 1982.
- [Que89] Christian Queinnec. Lisp – Almost a whole Truth. Technical Report LIX/RR/89/03, 79–106, Laboratoire d'Informatique de l'École Polytechnique, December 1989.
- [Que90a] Christian Queinnec. A Framework for Data Aggregates. In Pierre Cointe, Philippe Gautron, and Christian Queinnec, editors, *Actes des JFLA 90 – Journées Francophones des Langages Applicatifs*, pages 21–32, La Rochelle (France), January 1990. Revue Bigre+Globule 69.
- [Que90b] Christian Queinnec. *Le filtrage : une application de (et pour) Lisp*. InterÉditions, Paris (France), 1990. ISBN 2-7296-0332-8.
- [Que90c] Christian Queinnec. PolyScheme : A Semantics for a Concurrent Scheme. In *Workshop on High Performance and Parallel Computing in Lisp*, Twickenham (UK), November 1990. European Conference on Lisp and its Practical Applications.
- [Que92a] Christian Queinnec. Compiling syntactically recursive programs. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 5(4):2–10, October–December 1992.
- [Que92b] Christian Queinnec. A concurrent and distributed extension to scheme. In D. Etiemble and J-C. Syre, editors, *PARLE '92 – Parallel Architectures and Languages Europe*, pages 431–446, Paris (France), June 1992. Lecture Notes in Computer Science 605, Springer-Verlag.
- [Que93a] Christian Queinnec. Continuation conscious compilation. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 6(1):2–14, January 1993.
- [Que93b] Christian Queinnec. Designing MEROON v3. In Christian Rathke, Jürgen Kopp, Hubertus Hohl, and Harry Bretthauer, editors, *Object-Oriented Programming in Lisp: Languages and Applications. A Report on the ECOOP'93 Workshop*, number 788, Sankt Augustin (Germany), September 1993.
- [Que93c] Christian Queinnec. A library of high-level control operators. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 6(4):11–26, October 1993.
- [Que93d] Christian Queinnec. Literate programming from scheme to T<sub>E</sub>X. Research Report LIX RR 93.05, Laboratoire d'Informatique de l'École Polytechnique, 91128 Palaiseau Cedex, France, November 1993.
- [Que94] Christian Queinnec. Locality, causality and continuations. In *LFP '94 – ACM Symposium on Lisp and Functional Programming*, pages 91–102, Orlando (Florida, USA), June 1994. ACM Press.
- [Que95] Christian Queinnec. Dmeroon overview of a distributed class-based causally-coherent data model. In T. Ito, R. Halstead, and C. Queinnec, editors, *PSLS 95 – Parallel Symbolic Languages and Systems*, Beaune (France), October 1995.

- [R3R86] Revised<sup>3</sup> report on the algorithmic language scheme. *ACM Sigplan Notices*, 21(12), December 1986.
- [RA82] Jonathan A. Rees and Norman I. Adams. T: a dialect of lisp or, lambda: the ultimate software tool. In *Conference Record of the 1982 ACM Symposium on Lisp and Functional Programming*, pages 114–122, 1982.
- [RAM84] Jonathan A. Rees, Norman I. Adams, and James R. Meehan. *The T Manual, Fourth Edition*. Yale University Computer Science Department, January 1984.
- [Ray91] Eric Raymond. *The New Hacker's Dictionary*. MIT Press, Cambridge MA, 1991. With assistance and illustrations by Guy L. Steele Jr.
- [Rey72] John Reynolds. Definitional interpreters for higher order programming languages. In *ACM Conference Proceedings*, pages 717–740. ACM, 1972.
- [Rib69] Daniel Ribbens. *Programmation non numérique : Lisp 1.5*. Monographies d'Informatique, AFCET, Dunod, Paris, 1969.
- [RM92] John R Rose and Hans Muller. Integrating the Scheme and c languages. In *LFP '92 - ACM Symposium on Lisp and Functional Programming*, pages 247–259, San Francisco (California USA), June 1992. ACM Press. Lisp Pointers V(1).
- [Row90] William Rowan. A Lisp compiler producing compact code. In *Conference Record of the 1980 Lisp Conference [lfp80]*, pages 216–222.
- [Roz92] Guillermo Juan Rozas. Taming the y operator. In *LFP '92 - ACM Symposium on Lisp and Functional Programming*, pages 226–234, San Francisco (California USA), June 1992. ACM Press. Lisp Pointers V(1).
- [Sam79] Hanan Sammet. Deep and shallow binding: The assignment operation. *Computer Languages*, 4:187–198, 1979.
- [Sch86] David A Schmidt. *Denotational Semantics, a Methodology for Language Development*. Allyn and Bacon, 1986.
- [Sco76] Dana Scott. Data types as lattices. *Siam J. Computing*, 5(3):522–587, 1976.
- [Sén89] Nitsan Séniak. Compilation de Scheme par spécialisation explicite. *Revue Bigre+Globule*, (65):160–170, July 1989.
- [Sén91] Nitsan Séniak. *Théorie et pratique de Sqil, un langage intermédiaire pour la compilation des langages fonctionnels*. Thèse de doctorat d'université, Université Pierre et Marie Curie (Paris 6), Paris (France), October 1991.
- [Ser93] Manuel Serrano. De l'utilisation des analyses de flot de contrôle dans la compilation des langages fonctionnels. In Pierre Lescanne, editor, *Actes des journées du GDR de Programmation*, 1993.
- [Ser94] Manuel Serrano. *Bigloo User's Manual*, 1994. available on <ftp://ftp.inria.fr/INRIA/Projets/icsla/Implementations>.
- [SF89] George Springer and Daniel P. Friedman. *Scheme and the Art of Programming*. MIT Press and McGraw-Hill, 1989.
- [SF92] Amr Sabry and Matthias Felleisen. Reasoning about continuation-passing style programs. In *LFP '92 - ACM Symposium on Lisp and Functional Programming*, pages 288–298, San Francisco (California USA), June 1992. ACM Press. Lisp Pointers V(1).
- [SG93] Guy L. Steele, Jr. and Richard P Gabriel. The evolution of Lisp. In *The Second ACM SIGPLAN History of Programming Languages Conference (HOPL-II)*, pages 231–270, Cambridge (Massachusetts, USA), April 1993. ACM SIGPLAN Notices 8, 3.

- [Shi91] Olin Shivers. Data-flow analysis and type recovery in Scheme. In Peter Lee, editor, *Topics in Advanced Language Implementation*. The MIT Press, Cambridge, MASS, 1991.
- [SJ87] Emmanuel Saint-James. *De la Méta-Récurtivité comme Outil d'Implémentation*. Thèse d'état, Université Paris VI, December 1987.
- [SJ93] Emmanuel Saint-James. *La programmation applicative (de LISP à la machine en passant par le lambda-calcul)*. Hermès, 1993.
- [Sla61] J R Slagle. *A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculees, Symbolic Automatic Integration (SAINT)*. PhD thesis, MIT, Lincoln Lab, 1961.
- [Spi90] Éric Spir. *Gestion dynamique de la mémoire dans les langages de programmation, application à Lisp*. Science informatique. InterÉditions, Paris (France), 1990.
- [SS75] 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.
- [SS78a] 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.
- [SS78b] 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.
- [SS80] Guy Lewis Steele Jr. and Gerald Jay Sussman. The dream of a lifetime: a lazy variable extent mechanism. In *Conference Record of the 1980 Lisp Conference*, pages 163–172. The Lisp Conference, 1980.
- [Ste78] Guy Lewis Steele Jr. Rabbit: a compiler for Scheme. MIT AI Memo 474, Massachusetts Institute of Technology, Cambridge, Mass., May 1978.
- [Ste84] Guy L. Steele, Jr. *Common Lisp, the Language*. Digital Press, Burlington MA (USA), 1984.
- [Ste90] Guy L. Steele, Jr. *Common Lisp, the Language*. Digital Press, Burlington MA (USA), 2nd edition, 1990.
- [Sto77] Joseph E Stoy. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*. MIT Press, Cambridge Massachusetts USA, 1977.
- [Str86] Bjarne Stroustrup. *The C++ Programming Language*. Addison Wesley, 1986. ou "Le langage C++", InterÉditions 1989.
- [SW94] Manuel Serrano and Pierre Weis. 1+1=1: An optimizing caml compiler. In *Record of the 1994 ACM SIGPLAN Workshop on ML and its Applications*, pages 101–111, Orlando (Florida, USA), June 1994. INRIA RR 2265.
- [Tak88] M. Takeichi. Lambda-hoisting: A transformation technique for fully lazy evaluation of functional programs. *New Generation Computing*, (5):377–391, 1988.
- [Tei74] Warren Teitelman. *InterLISP Reference Manual*. Xerox Palo Alto Research Center, Palo Alto (California, USA), 1974.
- [Tei76] Warren Teitelman. Clisp: Conversational LISP. *IEEE Trans. on Computers*, C-25(4):354–357, April 1976.

- [VH94] Jan Vitek and R Nigel Horspool. Taming message passing: Efficient method look-up for dynamically typed languages. In *ECOOP '94 — 8th European Conference on Object-Oriented Programming*, Bologna (Italy), 1994.
- [Wad88] Philip Wadler. Deforestation: Transforming programs to eliminate trees. In H Ganzinger, editor, *ESOP '88 — European Symposium on Programming*, volume 300 of *Lecture Notes in Computer Science*, pages 344–358, 1988.
- [Wan80a] Mitchell Wand. Continuation-based multiprocessing. In *Conference Record of the 1980 Lisp Conference*, pages 19–28. The Lisp Conference, 1980.
- [Wan80b] Mitchell Wand. Continuation-based program transformation strategies. *Journal of the ACM*, 27(1):164–180, 1980.
- [Wan84] Mitchell Wand. A semantic prototyping system. In *Proceedings ACM SIGPLAN '84 Compiler Construction Conference*, pages 213–221, 1984.
- [Wan86] Mitchell Wand. The mystery of the tower revealed: a non-reflective description of the reflective tower. In *Proceedings of the 1986 ACM Symposium on LISP and Functional Programming*, pages 298–307, August 1986.
- [Wat93] Richard C Waters. Macroexpand-all: An example of a simple Lisp code walker. *Lisp Pointers, ACM SIGPLAN Special Interest Publication on Lisp*, 6(1):25–32, January 1993.
- [WC94] Andrew K Wright and Robert Cartwright. A practical soft typing system for Scheme. In *Proceedings of the 1994 ACM Conference on Lisp and Functional Programming [lfp94]*, pages 250–262.
- [WH88] Patrick H Winston and Berthold K Horn. *Lisp*. Addison Wesley, third edition, 1988.
- [Wil92] Paul R. Wilson. Uniprocessor garbage collection techniques. In Yves Bekkers and Jacques Cohen, editors, *International Workshop on Memory Management*, number 637 in *Lecture Notes in Computer Science*, pages 1–42, St. Malo, France, September 1992. Springer-Verlag.
- [WL93] Pierre Weis and Xavier Leroy. *Le langage Caml*. InterÉditions, 1993.
- [WS90] Larry Wall and Randal L Schwartz. *Programming perl*. O'Reilly & Associates, Inc., 1990.
- [WS94] Mitchell Wand and Paul Steckler. Selective and lightweight closure conversion. In *POPL'94 — 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 435–445, Portland (Oregon, USA), January 1994. ACM Press.
- [YH85] Taiichi Yuasa and Masami Hagiya. Kyoto common Lisp report. Technical report, Kyoto University, Japan, 1985.
- [YS92] Akinori Yonezawa and Brian C. Smith, editors. *Reflection and Meta-Level Architecture: Proceedings of the International Workshop on New Models for Software Architecture '92*, Tokyo, Japan, November 1992. Research Institute of Software Engineering (RISE) and Information-Technology Promotion Agency, Japan (IPA), in cooperation with ACM SIGPLAN, JSSST, IPSJ.