

References

1. M. Abadi, L. Cardelli, B. Pierce, and G. Plotkin. Dynamic Typing in a Statically Typed Language. *ACM Transactions on Programming Languages and Systems*, 13(2):237–268, 1991.
2. A. E. Abdallah. Derivation of Parallel Algorithms from Functional Specifications to CSP Processes. In B. Moller, editor, *Mathematics of Program Construction*, volume 947 of *LNCS*, pages 67–96. Springer Verlag, 1995.
3. A. E. Abdallah. Synthesis of Massively Pipelined Algorithms for List Manipulation. In A. M. L. Bouge, P. Fraigniaud and Y. Robert, editors, *Proceedings of the European Conference on Parallel Processing, EuroPar'96, LNCS 1024*, pages 911–920. Springer Verlag, 1996.
4. A. E. Abdallah. An Algebraic Approach for Refining Functions into CSP Processes. (*Submitted to*) *Formal Aspects of Computing*, 1999.
5. S. Abramsky. The Lazy Lambda Calculus. In D. A. Turner, editor, *Research Topics in Functional Programming*. Addison Wesley, 1990.
6. S. Abramsky and C. Hankin. *Abstract Interpretation of Declarative Languages*. Ellis Horwood, Chichester, West Sussex, 1987.
7. P. Achten. Annotations for Load Distribution. In H. Glaser and P. Hartel, editors, *Proceedings of the Workshop on Parallel Implementation of Functional Languages, CSTR 91-07*, pages 247–264, University of Southampton, U.K., 1991.
8. G. Akerholt, K. Hammond, S. L. Peyton Jones, and P. Trinder. Processing Transactions on GRIP. In *PARLE'93 — Parallel Languages and Architectures Europe*, Lecture Notes in Computer Science, pages 634–647, Munich, Germany, June 14–18, 1993. Springer-Verlag.
9. A. G. Alexandrakis, A. V. Gerbessiotis, D. S. Lecomber, and C. J. Siniolakis. Bandwidth, Space and Programming: The BSP Approach. In *Proceedings of SUPEUR'96*, Krakow, May 1996.
10. M. Amamiya. Data Flow Computing and Parallel Reduction Machine. *Future Generation Computer Systems*, 4:53–67, 1988.
11. M. Amamiya and R. Hasegawa. Dataflow Computing and Eager and Lazy Evaluation. *New Generation Computing*, 2:105–129, 1984.
12. T. Amtoft, F. Nielson, and H. R. Nielson. *Type and Effect Systems: Behaviours for Concurrency*. Imperial College Press, 1999.
13. T. Amtoft, H. R. Nielson, and F. Nielson. Behaviour Analysis for Validating Communication Patterns. *Software Tools for Technology Transfer*, 2(1):13–28, 1998.
14. J. Armstrong. Erlang — a Survey of the Language and its Industrial Applications. In *INAP'96 — The 9th Exhibitions and Symposium on Industrial Applications of Prolog*, pages 16–18, Hino, Tokyo, Japan, October 1996.
15. J. Armstrong, R. Virding, C. Wikström, and M. Williams. *Concurrent Programming in Erlang, Second Edition*. Prentice-Hall, 1996.

16. Arvind and D. Culler. Dataflow Architectures. *Annual Review of Computer Science*, 1:225–254, 1986.
17. Arvind, V. Kathail, and K. K. Pingali. A Dataflow Architecture with Tagged Tokens. Technical Report LCS Memo TM-174, MIT, 1980.
18. Arvind and R. S. Nikhil. Can Dataflow Subsume von Neumann Computing? Technical Report CSG Memo 292, MIT, November 1988.
19. Arvind and R. S. Nikhil. Executing a Program on the MIT Tagged-Token Dataflow Architecture. *IEEE Transactions on Computers*, 39(3), 1990. Also: CSG Memo 271.
20. Arvind, N. R. S., and K. K. Pingali. I-structures: Data Structures for Parallel Computing. *ACM Transactions on Programming Languages and Systems*, 11(4):589–632, October 1989.
21. Arvind and R. Thomas. I-Structures : An Efficient Data Type for Functional Languages. Technical Report 178, MIT Laboratory of Computer Science, 1980.
22. L. Augustsson. A Parallel G-Machine. Technical Report PMG53, Department of Computer Science, Chalmers University of Technology, Gteborg, SE, 1987.
23. L. Augustsson. *Compiling Lazy Functional Languages, Part II*. Phd thesis, Chalmers University of Technology, Gteborg, SE, 1987.
24. L. Augustsson and T. Johnsson. Parallel Graph Reduction with the $< \nu, G >$ -Machine. In *Proceedings of the Conference on Functional Programming Languages and Computer Architecture '89, Imperial College, London*, pages 202–213, New York, NY, 1989. ACM.
25. B. Bacci, B. Cantalupo, M. Danelutto, S. Orlando, D. Pasetto, S. Pelagatti, and M. Vanneschi. An Environment for Structured Parallel Programming. In *Advances in High Performance Computing*, pages 219–234. Kluwier, 1997.
26. B. Bacci, M. Danelutto, S. Orlando, S. Pelagatti, and M. Vanneschi. P³L: A Structured High Level Programming Language and its Structured Support. *Concurrency — Practice and Experience*, 7(3):225–255, May 1995.
27. J. Backus. Can Programming Be Liberated From the Von Neumann Style? *Commuinactions of the ACM*, 21(8), 1978.
28. D. Bailey and et. al. The NAS Parallel Benchmarks. Technical Report Report RNR-91-002 revision 2, NASA Ames Research Center, 1991.
29. P. Bailey. Algorithmic Skeletons in paraML. TRACS Research Report, Edinburgh Parallel Computing Centre, 1994.
30. P. Bailey and M. Newey. An Extension of ML for Distributed Memory Computers. In *Proceedings of 16th Australian Computer Science Conference, Brisbane, Australia*, pages 387 – 396, 1993.
31. P. Bailey and M. Newey. Implementing ML on Distributed Memory Multi-computers. *ACM SIGPLAN Notices*, 28(1):59 – 63, 1993.
32. P. Bailey, M. Newey, D. Sitsky, and R. Stanton. Supporting Coarse and Fine Grain Parallelism in an Extension of ML. In *CONPAR'94 — Conference on Algorithms and Hardware for Parallel Processing*, volume 854 of *Lecture Notes in Computer Science*, pages 693–704, Linz, Austria, September 1994. Springer-Verlag.
33. M. Bamha and G. Hains. A Self-Balancing Join Algorithm for SN Machines. In *PDCS'98 10th International Conference on Parallel and Distributed Computing Systems*, Las Vegas, 1998.
34. H. Barendregt, M. van Eekelen, P. Hartel, L. Hertzberger, M. Plasmeijer, and W. Vree. The Dutch Parallel Reduction Machine Project. *Future Generation Computer Systems*, 3(4):261–270, December 1987.
35. H. P. Barendregt. *The Lambda Calculus: its Syntax and Semantics*. North-Holland, 1981.

36. E. Barendsen and J. E. W. Smetsers. Extending Graph Rewriting with Copying. In *Graph Transformations in Computer Science*, volume 776 of *LNCS*, pages 51–70, Dagstuhl, Wadern, 1993. Springer-Verlag.
37. G. H. Barnes, R. M. Brown, M. Kato, D. Kuck, D. Slotnick, and R. Stokes. The ILLIAC IV Computer. In C. G. Bell and A. Newell, editors, *Computer Structures: Readings and Examples*, chapter 27. McGraw-Hill, 1971.
38. M. Barnett, D. Payne, R. van de Geijn, and J. Watts. Broadcasting on Meshes with Wormhole Routing. *Journal of Parallel and Distributed Computing*, 35(2):111–122, 1996.
39. P. S. Barth, R. S. Nikhil, and Arvind. M-Structures: Extending a Parallel, Non-Strict, Functional Language with State. In *FPCA'91 — Conference on Functional Programming Languages and Computer Architectures*, volume 523 of *Lecture Notes in Computer Science*, pages 538–568, Harvard, MA, Aug. 1991. Springer-Verlag. Also: CSG Memo 327.
40. C. Batini and G. Battista. A Methodology for Conceptual Documentation and Maintenance. *Information Systems*, 13(3):297–318, 1988.
41. A. Baumker and W. Dittrich. Fully Dynamic Search Trees for an Extension of the BSP Model. In *SPAA'96: ACM Symposium on Parallel Algorithms and Architectures*, pages 233–242, 1996.
42. A. Baumker, W. Dittrich, F. Meyer auf der Heide, and I. Rieping. Realistic Parallel Algorithms: Priority Queue Operations and Selection for the BSP* Model. In *EuroPar'96*, number 1124 in *Lecture Notes in Computer Science*, pages 369–376, Lyon, France, Aug. 1996. Springer-Verlag.
43. J. L. Bentley. Multi-Dimensional Divide-and-Conquer. *CACM*, 23:214–229, 1980.
44. K. Berkling. Reduction Languages for Reduction Machines. In *Proceedings of the 2nd Annual Symposium on Computer Architecture*, pages 133–140. ACM/IEEE, 1975.
45. R. S. Bird. Introduction to the Theory of Lists. In M. Broy, editor, *Logic of Programming and Calculi of Discrete Design*. Springer-Verlag, 1987.
46. R. S. Bird. Algebraic Identities for Program Calculation. *The Computer Journal*, 32(2):122–126, 1989.
47. R. S. Bird. A Calculus of Functions for Program Derivation. In D. A. Turner, editor, *Research Topics in Functional Programming*. Addison Wesley, 1990.
48. R. S. Bird. Functional Algorithm Design. *Mathematics of Program Construction, LNCS 947*, pages 2–17, 1995.
49. R. S. Bird. *Introduction to Functional Programming Using Haskell*. Prentice-Hall, 2nd edition, 1998.
50. R. S. Bird and O. de Moor. List Partitions. *Formal Aspects of Computing*, 5(1), 1993.
51. R. S. Bird and O. de Moor. *Algebra of Programming*. Prentice-Hall, 1997.
52. R. S. Bird and P. Wadler. *Introduction to Functional Programming*. Prentice-Hall, 1988.
53. R. H. Bisseling. Sparse Matrix Computations on Bulk Synchronous Parallel Computers. In G. Alefeld, O. Mahrenholtz, and R. Mennicken, editors, *Proceedings ICIAM'95. Issue 1. Numerical Analysis, Scientific Computing, Computer Science*, pages 127–130. Akademie Verlag, Berlin, 1996.
54. R. H. Bisseling. Basic Techniques for Numerical Linear Algebra on Bulk Synchronous Parallel Computers. In L. Vulkov, J. Waśniewski, and P. Yalamov, editors, *Proceedings First Workshop on Numerical Analysis and Applications, Rousse, Bulgaria, 1996*, volume 1196 of *Lecture Notes in Computer Science*, pages 46–57. Springer-Verlag, Berlin, 1997.

55. R. H. Bisseling and W. F. McColl. Scientific Computing on Bulk Synchronous Parallel Architectures. In B. Pehrson and I. Simon, editors, *Technology and Foundations: Information Processing '94, Vol. 1*, volume 51 of *IFIP Transactions A*, pages 509–514. Elsevier Science Publishers, Amsterdam, 1994.
56. D. Bitton, D. DeWitt, and C. Turbyfill. Benchmarking Database Systems: A Systematic Approach. In *Very Large Database Conference*, 1983.
57. G. E. Blelloch. *Vector Models for Data-Parallel Computing*. MIT Press, 1990.
58. G. E. Blelloch. Programming Parallel Algorithms. *Communications of the ACM March 1996*, 39(3):85–97, 1996.
59. G. E. Blelloch, S. Chatterje, J. Hardwick, J. Sipelstein, and M. Zagha. Implementation of a Portable Nested Data-Parallel Language. *Journal of Parallel and Distributed Computing*, 21(1):4 – 14, 1994.
60. G. E. Blelloch and J. Hardwick. A Library of Parallel Algorithms. Web Page, February 1999.
61. G. E. Blelloch, G. L. Miller, and D. Talmor. Developing a Practical Projection-Based Parallel Delaunay Algorithm. In *12th Annual Symposium on Computational Geometry*. ACM, May 1996.
62. G. E. Blelloch and G. Narlikar. A Practical Comparison of N -Body Algorithms. In *Parallel Algorithms*, volume 30 of *Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1997.
63. G. E. Blelloch and G. Sabot. Compiling Collection-Oriented Languages onto Massively-Parallel Computers. *Journal of Parallel and Distributed Computing*, pages 119–134, 1990.
64. C. Bodei, P. Degano, F. Nielson, and H. R. Nielson. Control Flow Analysis for the π -Calculus. In *Proceedings of CONCUR'98*, number 1466 in Lecture Notes in Computer Science, pages 84–98. Springer, 1998.
65. C. Bodei, P. Degano, F. Nielson, and H. R. Nielson. Static Analysis of Processes for No Read-Up No Write-Down. In *Proceedings of FOSSACS'99*, number 1578 in Lecture Notes in Computer Science, pages 120–134. Springer, 1999.
66. C. Boeres, V. E. F. Rebello, and D. B. Skillicorn. Static Scheduling Using Task Replication for LogP and BSP Models. In *EuroPar'98*, volume 1470 of *LNCS*, pages 698–708. Springer-Verlag, September 1998.
67. A. Böhm and R. Hiromoto. The Dataflow Complexity of Fast Fourier Transforms. In L. Bic, G. R. Gao, and J. L. Gaudiot, editors, *Advanced Topics in Dataflow Computing and Multithreading*. IEEE CS Press, 1995.
68. A. Böhm and J. Sargeant. Code Optimisation for Tagged Token Dataflow Machines. *IEEE Transactions on Computers*, 38(1):4–14, January 1989.
69. A. P. W. Böhm and R. E. Hiromoto. Dataflow Time and Space Complexity of FFTs. *Journal of Parallel and Distributed Computing*, 18, 1993.
70. A. P. W. Böhm and R. E. Hiromoto. Functional Implementations of the Jacobi Eigen-Solver. *Journal of Scientific Programming*, 5:111–120, 1996.
71. O. Bonorden, B. Juulink, I. von Otto, and I. Rieping. The Paderborn University BSP (PUB) Library—Design, Implementation and Performance. In *13th International Parallel Processing Symposium & 10th Symposium on Parallel and Distributed Processing*, April 1999.
72. G. Botorog. *High-Level Parallel Programming and the Efficient Implementation of Numerical Algorithms*. PhD thesis, RWTH-Aachen, Jan. 1998. Also: Technical Report 97-15.
73. G. Botorog and H. Kuchen. Algorithmic Skeletons for Adaptive Multigrid Methods. In *Irregular '95*, volume 980 of *Lecture Notes in Computer Science*, pages 27–41, 1995.

74. G. Botorog and H. Kuchen. Skil: An Imperative Language with Algorithmic Skeletons for Efficient Distributed Programming. In *HPDC'96 — International Symposium on High Performance Distributed Computing*, pages 243–252. IEEE Computer Society Press, 1996.
75. G. Botorog and H. Kuchen. Efficient High-Level Parallel Programming. *Theoretical Computer Science*, 196:71–107, 1998.
76. G. H. Botorog and H. Kuchen. Efficient Parallel Programming with Algorithmic Skeletons. In L. Bouge, P. Fraigniaud, A. Mignotte, and Y. Robert, editors, *Proceedings of EuroPar '96*, volume 1123 of *LNCS*, pages 718–731. Springer, 1996.
77. D. Boucher and M. Feeley. Construction Parallèle de l'Automate LR(0): Une Application de MultiLisp à la Compilation. In *6ième Rencontres Francophones du Parallélisme*. Département d'Informatique et R.O., Université de Montréal, June 1994.
78. L. Bougé. The Data-Parallel Programming Model: A Semantic Perspective. In *The Data-Parallel Programming Model*, volume 1132 of *LNCS Tutorial*, pages 4–26. Springer-Verlag, June 1996.
79. T. A. Bratvold. Parallelising a Functional Program Using a List-Homomorphism Skeleton. In H. Hong, editor, *PASCO'94: First International Symposium on Parallel Symbolic Computation*, pages 44–53. World Scientific Publishing Company, Sept. 1994.
80. T. A. Bratvold. *Skeleton-Based Parallelisation of Functional Programs*. PhD thesis, Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh, November 1994.
81. S. Breitinger, U. Klusik, and R. Loogen. From (sequential) Haskell to (parallel) Eden: An Implementation Point of View. In *International Symposium on Programming Languages: Implementations, Logics, Programs (PLILP)*, LNCS 1490. Springer, 1998.
82. S. Breitinger, U. Klusik, R. Loogen, Y. Ortega-Mallen, and R. Pena. DREAM: The Distributed Eden Abstract Machine. In C. Clack, K. Hammond, and T. Davie, editors, *Implementation of Functional Languages: Proceedings of the 9th International Workshop, IFL'97, St. Andrews, Scotland*, volume 1467 of *LNCS*, pages 250–269. Springer-Verlag, 1997.
83. S. Breitinger, R. Loogen, Y. Ortega-Mallén, and R. Peña. Eden — Language Definition and Operational Semantics. Technical Report 10, Philipps-Universität Marburg, 1996.
84. S. Breitinger, R. Loogen, Y. Ortega-Mallén, and R. Peña. The Eden Coordination Model for Distributed Memory Systems. In *High-Level Parallel Programming Models and Supportive Environments (HIPS)*. IEEE Press, 1997.
85. R. Brent. The Parallel Evaluation of General Arithmetic Expressions. *Journal of the ACM*, 21, No.2:201–206, April 1974.
86. J. F. Briesmeister. MCNP—A General Monte Carlo N-Particle Transport Code, Version 4A. Technical Report Report LA-12625-M, Los Alamos National Laboratory, 1993.
87. P. Brinch-Hansen. *Studies in Computational Science*. Prentice-Hall, 1995.
88. P. Brinch-Hansen. *The Search for Simplicity*. IEEE Computer Press, 1996.
89. S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A Theory of Communicating Sequential Processes. *Journal of the ACM*, 31(3):560–599, 1984.
90. R. Bru and J. Marin. BSP Cost of the Preconditioned Conjugate Gradient Method. In *Proceedings of the VII Jornadas de Paralelismo*, Santiago de Compostela, Spain, September 1996. Universidad de Santiago de Compostela.
91. T. Brus, M. van Eekelen, M. van Leer, and M. Plasmeijer. Clean: A Language for Functional Graph Rewriting. In *Third International Conference on*

- Functional Programming Languages and Computer Architecture*, volume 274 of LNCS, pages 364–384, Portland, Oregon, USA, 1987. Springer-Verlag.
92. T. Buelck, A. Held, W. Kluge, S. Pantke, C. Rath sack, S. Scholz, and R. Schroeder. Experience with the Implementation of a Concurrent Graph Reduction System on an nCUBE/2 Platform. In *Joint International Conference on Parallel and Vector Processing*. Springer, LNCS 854, 1994.
 93. Bulloch. Scans as Primitive Parallel Operations. *IEEE Transaction on Computers*, 38(11):1526–1538, 1989.
 94. W. Burge. *Recursive Programming Techniques*. Addison-Wesley, Reading, MA, 1975.
 95. D. A. Burgess, P. I. Crumpton, and M. B. Giles. A Parallel Framework for Unstructured Grid Solvers. In S. Wagner, E. Hirschel, J. Périaux, and R. Piva, editors, *Computational Fluid Dynamics '94. Proceedings of the Second European Computational Fluid Dynamics Conference 5-8 September 1994 Stuttgart, Germany*, pages 391–396. John Wiley & Sons, 1994.
 96. D. A. Burgess and M. B. Giles. Renumbering Unstructured Grids to Improve the Performance of Codes on Hierarchical Memory Machines. Technical Report 95/06, Numerical Analysis Group, Oxford University Computing Laboratory, Oxford OX1 3QD, England, May 1995.
 97. G. Burn. Evaluation Transformers - A Model for the Parallel Evaluation of Functional Languages. In G. Kahn, editor, *Functional Programming Languages and Computer Architecture*, pages 446–470. Springer-Verlag, Berlin, DE, 1987.
 98. G. Burn. A Shared Memory Parallel G-machine Based on the Evaluation Transformer Model of Computation. In T. Johnsson et al., editors, *Aspens Workshop on Implementation of Functional Languages*, pages 301–330. Programming Methodology Group, University of Gteborg and Chalmers University of Technology, 1988.
 99. G. Burn. The Evaluation Transformer Model of Reduction and its Correctness. In *TAPSOFT '91*, pages 458–482. Springer-Verlag, New York, NY, 91.
 100. G. Burn, C. L. Hankin, and S. Abramsky. Strictness Analysis of Higher Order Functions. *Science of Computer Programming*, 7, 1985.
 101. G. Burn, S. L. Peyton Jones, and J. Robson. The Spineless G-machine. In *Proceedings of the 1988 ACM Conference on LISP and Functional Programming, Snowbird, UT*, pages 244–258, New York, NY, 1988. ACM.
 102. F. W. Burton. Functional Programming for Concurrent and Distributed Computing. *Computer Journal*, 30(5):437–450, 1987.
 103. F. W. Burton and M. R. Sleep. Executing Functional Programs on a Virtual Tree of Processors. In *Proceedings of the ACM Conference on Functional Programming Languages and Computer Architecture, Portsmouth, NH*, pages 187–194, New York, 1981. ACM.
 104. R. Butler and E. Lusk. Users's Guide to the p4 Parallel Programming System. Technical Report ANL-92/17, Mathematics and Computer Science Division, Argonne National Laboratory, Oct 92.
 105. E. Caceres, F. Dehne, A. Ferreira, P. Flocchini, I. Rieping, A. Roncato, N. Santoro, and S. W. Song. Efficient Parallel Graph Algorithms for Coarse Grained Multicomputers and BSP. In *24th International Colloquium on Automata, Languages and Programming (ICALP'97)*, volume 1256 of LNCS, pages 390–400, Bologna, Italy, 1997. Springer-Verlag.
 106. R. Calinescu. Bulk Synchronous Parallel Algorithms for Conservative Discrete Event Simulation. *Parallel Algorithms and Applications*, 9:15–38, 1996. A preliminary version of this paper appeared as Technical Report PRG-TR-16-95, Programming Research Group, Oxford University Computing Laboratory,

- April 1995.
107. R. Calinescu. A BSP Approach to the Scheduling of Tightly-Nested Loops. In *International Parallel Processing Symposium (IPPS'97)*, pages 549–553, Geneva, Switzerland, April 1997. IEEE Computer Society Press.
 108. R. Calinescu and D. Evans. Bulk-Synchronous Parallel Algorithms for QR and QZ Matrix Factorisation. *Parallel Algorithms and Applications*, 11:97–112, 1997.
 109. D. K. G. Campbell. Towards the Classification of Algorithmic Skeletons. Technical Report YCS 276, Department of Computer Science, University of York, 1996.
 110. D. Cann. Compilation Techniques for High Performance Applicative Computation. Technical Report CS-89-108, Lawrence Livermore National Laboratory, LLNL, Livermore, California, 1989.
 111. D. Cann. Retire Fortran? A Debate Rekindled. *Communications of the ACM*, 35(8):81–89, August 1992.
 112. D. Cann and P. Evripidou. Advanced Array Optimizations for High Performance Functional Languages. *IEEE Transactions on Parallel and Distributed Systems*, 6(3):229–239, 1995.
 113. L. Cardelli. ML Under UNIX. *Polymorphism: The ML/HCF/Hope Newsletter*, I(3), December 1983.
 114. L. Cardelli and D. McQueen. The Functional Abstract Machine. *The ML/LCF/HOPE Newsletter, AT&T, Bell Labs, Murray Hill, NJ*, 1983.
 115. M. Castan, G. Durrieu, B. Lecussan, M. Lematre, A. Contessa, E. Cousin, and P. Ng. Toward the Design of a Parallel Graph Reduction Machine: The MaRS Project. In J. Fasel and R. Keller, editors, *Graph Reduction: Proceedings of a Workshop at Santa F, New Mexico*, pages 160–180, New York, NY, 1987. Springer-Verlag. Lecture Notes in Computer Science 279.
 116. M. M. Chakravarty. Integrating Multithreading into the Spineless Tagless G-machine. In D. N. Turner, editor, *Glasgow Workshop on Functional Programming*, Workshops in Computing. Springer Verlag, 1995.
 117. N. Charles. *New Tools for Parallelism in Lazy Functional Programs*. D.Phil Thesis, Department of Computer Science, University of York, 1999. Forthcoming.
 118. N. Charles and C. Runciman. An Interactive Approach to Profiling Parallel Functional Programs. In *Draft Proceedings of 10th International workshop on the Implementation of Functional Languages (IFL '98), London, England*, pages 447–463, 1998.
 119. J. Chattratchat, J. Darlington, M. Ghanem, Y. Guo, H. Hüning, M. Köhler, J. Sutiwaraphun, H. To, and D. Yang. Large Scale Data Mining: The Challenges and The Solutions. In *KDD97 — International Conference on Knowledge Discovery and Data Mining*. AAAI Press, August 1997.
 120. I. Checkland. *Speculative Concurrent Evaluation in a Lazy Functional Language*. PhD thesis, Department of Computer Science, University of York, 1994.
 121. I. Chern and I. Foster. Design and Parallel Implementation of Two Methods for Solving PDEs on the Sphere. In *Conference on Parallel Computational Fluid Dynamics*, pages 83–96, Stuttgart, Germany, 1991. Elsevier Science Publishers.
 122. A. Church. *The Calculi of Lambda Conversion*. Princeton University Press, 1941.
 123. CIP Language Group. *The Munich Project CIP, LNCS 1*. Springer-Verlag, 1984.

124. C. Clack. GRIP Status Update — 1989. In T. Fountain and M. Shute, editors, *Multiprocessor Computer Architecture*, pages 119–120. North Holland, Amsterdam, 1990.
125. C. Clack. The Implementation of Sum and Product Domain Constructors for the Four-Stroke Reduction Engine. Research Note RN/92/54, Department of Computer Science, University College London, London, UK, 1992.
126. C. Clack. GRIP: the GRIP Multiprocessor. In T. Fountain, editor, *Parallel Computing Principles and Practice*, pages 266–275. Cambridge University Press, 1994.
127. C. Clack. The DIGRESS Project: Final Report. Internal technical report, Athena Systems Design Ltd., London, UK, 1994.
128. C. Clack, S. Clayman, and D. Parrott. Lexical Profiling — Theory and Practice. *Journal of Functional Programming*, 5(2):225–277, 1995.
129. C. Clack and S. Courtenage. An Overview of the UCL DIGRESS Project. Research Note RN/92/55, Department of Computer Science, University College London, London, UK, 1992.
130. C. Clack and S. L. Peyton Jones. Strictness Analysis - A Practical Approach. In J.-P. Jouannaud, editor, *Functional Programming Languages and Computer Architecture*, pages 35–49. Springer-Verlag, Berlin, DE, 1985.
131. C. Clack and S. L. Peyton Jones. The Four-Stroke Reduction Engine. In *Proceedings ACM Conference on Lisp and Functional Programming*, pages 220–232. ACM, 1986.
132. C. Clack, S. L. Peyton Jones, and J. Salkild. Efficient Parallel Graph Reduction on GRIP. Research Note RN/88/29, Department of Computer Science, University College London, London, UK, 1988.
133. T. Clarke, P. Gladstone, C. MacLean, and A. Norman. SKIM - The S,K,I Reduction Machine. In *Conference Record of the 1980 LISP Conference, Stanford University*, 1980.
134. M. I. Cole. Algorithmic Skeletons: A Structured Approach to the Management of Parallel Computation. PhD thesis CST-56-88, University of Edinburgh, Computer Science Department, 1988.
135. M. I. Cole. *Algorithmic Skeletons: Structured Management of Parallel Computation*. Research Monographs in Parallel and Distributed Computing. Pitman, 1989.
136. E. Cooper and J. Morrisett. Adding Threads to Standard ML. Technical Report CMU-CS-90-186, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213, 1990.
137. C. Corral, I. Gimenez, J. Marin, and J. Mas. An M-Step Block Diagonal Multisplitting Preconditioned Conjugate Gradient Method. In *EAMA-97 (International Meeting on Matrix Analysis and Applications)*, September 1997.
138. G. Cousineau, P. Curien, and M. Mauny. The Categorical Abstract Machine. *Science of Computer Programming*, 8:173–202, 1987.
139. G. Cousineau and M. Mauny. *The Functional Approach to Programming*. Cambridge University Press, 98.
140. P. Cousot. Methods and Logics for Proving Programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*. MIT Press/Elsevier, 1990.
141. S. Cox, S.-Y. Huang, P. H. J. Kelly, J. Liu, and F. Taylor. An Implementation of Static Functional Process Networks. In D. Etiemble and J.-C. Syre, editors, *PARLE '92: Proceedings of the 4th International Conference on Parallel Architectures and Languages Europe, Paris*, pages 497–514, Berlin, DE, 1992. Springer-Verlag.

142. M. D. Cripps, J. Darlington, A. J. Field, P. G. Harrison, and M. J. Reeve. The Design and Implementation of ALICE: A Parallel Graph Reduction Machine. In *Proceedings of the Workshop on Graph Reduction*, New York, NY, 1987. Springer-Verlag.
143. P. I. Crumpton and M. B. Giles. Multigrid Aircraft Computations Using the OPlus Parallel Library. In *Parallel Computational Fluid Dynamics: Implementation and Results Using Parallel Computers. Proceedings Parallel CFD'95*, pages 339–346, Pasadena, CA, USA, June 1995. Elsevier/North-Holland.
144. D. Culler. *Managing Parallelism and Resources in Scientific Dataflow Programs*. PhD thesis, Laboratory for Computer Science, M.I.T., June 1989.
145. D. Culler and Arvind. Resource Requirements of Dataflow Programs. In *15th Annual ACM Symposium on Computer Architecture*, 1988.
146. D. Culler, S. Goldstein, K. Schauser, and T. von Eicken. TAM — A Compiler Controlled Threaded Abstract Machine. *Journal of Parallel and Distributed Computing*, 18:347–370, June 1993.
147. D. E. Culler and J. P. Singh. *Parallel Computer Architecture*. Pitman/MIT Press, 1989.
148. P. L. Curien. *Categorical Combinators, Sequential Algorithms and Functional Programming*. Research Notes in Theoretical Computer Science. Wiley, 1986.
149. H. Curry, W. Craig, and R. Feys. *Combinatory Logic, Volume 1*. North-Holland, Amsterdam, NL, 1958.
150. O. Danvy, R. Glück, and P. Thiemann, editors. *Partial Evaluation*, volume 1110 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.
151. J. Darlington. A Synthesis of Several Sorting Algorithms. *Acta Informatica*, 11(1), 1978.
152. J. Darlington, A. J. Field, P. G. Harrison, P. H. J. Kelly, D. W. N. Sharp, and Q. Wu. Parallel Programming Using Skeleton Functions. In A. Bode, M. Reeve, and G. Wolf, editors, *Proceedings of PARLE '93*, pages 146–160, 1993.
153. J. Darlington, M. Ghanem, Y. Guo, and H. To. Guided Resource Organisation in Heterogeneous Parallel Computing. *Journal of High Performance Computing*, 4(1):13–23, Dec. 1997.
154. J. Darlington, M. Ghanem, and H. To. Structured Parallel Programming. In *Massively Parallel Programming Models Conference*, pages 160–169, Berlin, Sept 1993. IEEE Computer Society Press.
155. J. Darlington, Y. Guo, H. To, and J. Yang. Functional Skeletons for Parallel Coordination. In *EuroPar'95 — European Conference on Parallel Processing*, volume 966 of *Lecture Notes in Computer Science*, pages 55–69, Stockholm, Sweden, August 29–31, Aug. 1995. Springer-Verlag.
156. J. Darlington, Y. Guo, H. To, and J. Yang. Parallel Skeletons for Structured Composition. In *Proceedings of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 19–28. ACM Press, 1995.
157. J. Darlington, Y. Guo, H. To, and J. Yang. SPF: Structured Parallel Fortran. In *PCW'96 — International Parallel Computing Workshop*, Kawasaki, Japan, November 1996.
158. J. Darlington, Y. Guo, H. W. To, Q. Wu, J. Yang, and M. Kohler. Fortran-S: A Uniform Functional Interface to Parallel Imperative Languages. In *Proceedings of the Third Parallel Computing Workshop*, Fujitsu Laboratories Ltd, Kawasaki Japan, 1994.
159. J. Darlington and M. Reeve. ALICE - A Multi-Processor Reduction Machine for the Parallel Evaluation of Applicative Languages. In *Proceedings of the ACM Conference on Functional Languages and Computer Architecture, Portsmouth, NH*, pages 65–76, 1981.

160. J. Darlington and H. To. Supporting Irregular Applications in SPF. In *PCW'97 — International Parallel Computing Workshop*, Canberra, Australia, September 1997.
161. J. Darlington and H. W. To. Building Parallel Applications Without Programming. In *Abstract Machine Models*, Leeds, 1993.
162. D. Das and D. Batory. Prairie: A Rule Specification Framework for Query Optimizers. In *International Conference on Database Engineering*, pages 201–210. iee, 1995.
163. A. Davie and D. McNally. CASE - A Lazy Version of an SECD Machine with a Flat Environment. In *Proceedings of the IEEE TENCON '89, Bombay*, New York, NY, 1989. IEEE.
164. J. R. Davy, H. Deldari, and P. M. Dew. Constructive Solid Geometry Using Algorithmic Skeletons. In *Eurographics Workshop on Programming Paradigms for Graphics*. Springer-Verlag, 1995.
165. J. R. Davy, H. Deldarie, and P. M. Dew. *Constructive Solid Geometry Using Algorithmic Skeletons*. Research Report 96.1, School of Computer Studies, University of Leeds, 1996.
166. N. G. de Bruijn. Lambda-Calculus Notation with Nameless Dummies. A Tool for Automatic Formula Manipulation with Application to the Church-Rosser Theorem. *Indagationes Mathematicae*, 34:484–492, 1972.
167. J. B. Dennis. Dataflow Computation.
168. J. B. Dennis. First Version of a Data-Flow Procedure Language. In B. Robinet, editor, *Proceedings of the Colloque sur la Programmation*, volume 19 of *LNCS*, pages 362–376, Berlin, DE, 1974. Springer-Verlag.
169. F. Desprez, S. Domas, J. Dongarra, A. Petitet, C. Randriamaro, and Y. Robert. More on Scheduling Block-Cyclic Array Redistribution. In *Proceedings of 4th Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers (LCR98)*, volume 1511 of *Lecture Notes in Computer Science*, pages 275–287, Pittsburgh, PA, 1998. Springer Verlag.
170. I. Dimov, A. Karaivanova, H. Kuchen, and H. Stoltze. Monte Carlo Algorithms for Elliptic Differential Equations — a Data Parallel Functional Approach. *Journal of Parallel Algorithms and Applications*, 9:39–65, 1996.
171. S. R. Donaldson, J. M. D. Hill, and D. B. Skillicorn. BSP clusters: High performance, reliable and very low cost. Technical Report PRG-TR-5-98, Programming Research Group, Oxford University Computing Laboratory, September 1998.
172. S. R. Donaldson, J. M. D. Hill, and D. B. Skillicorn. Predictable Communication on Unpredictable Networks: Implementing BSP over TCP/IP. In *EuroPar'98*, LNCS, Southampton, UK, September 1998. Springer-Verlag.
173. S. R. Donaldson, J. M. D. Hill, and D. B. Skillicorn. Communication Performance Optimisation Requires Minimising Variance. *Journal of Future Generation Computer Systems*, April 1999.
174. S. R. Donaldson, J. M. D. Hill, and D. B. Skillicorn. Exploiting Global Structure for Performance on Clusters. In *IPPS/SPDP*, San Juan, Puerto Rico, April 1999. Springer-Verlag.
175. S. R. Donaldson, J. M. D. Hill, and D. B. Skillicorn. Performance Results for a Reliable Low-latency Cluster Communication Protocol. In *PC-NOW '99: International Workshop on Personal Computer based Networks Of Workstations, held in conjunction with IPPS'99*, San Juan, Puerto Rico, April 1999. Springer-Verlag.
176. J. J. Dongarra and D. C. Sorensen. A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem. *SIAM Journal of Scientific and Statistical Computation*, 8:139–154, March 1987.

177. M. C. Dracopoulos, C. Glasgow, K. Parrott, and J. Simkin. Bulk Synchronous Parallelisation of Industrial Electromagnetic Software. *International Journal of Supercomputer Application of High Performance Computing*, 1996.
178. R. Dybvig. *The SCHEME Programming Language*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
179. G. Egan. Implementing the Kernel of the Australian Weather Prediction Model in Sisal. In *Sisal'93 — Sisal Users Conference*, number CONF 9310206 in LLNL Report, pages 11–17, San Diego, CA, Oct. 1993.
180. J. Eisenbiegler, W. Löwe, and W. Zimmermann. BSP, LogP, and Oblivious Programs. In *EuroPar'98*, volume 1470 of *LNCS*, pages 865–874. Springer-Verlag, September 1998.
181. H. El-Rewini and T. Lewis. Scheduling Parallel Programming Tasks onto Arbitrary Target Machines. *Journal of Parallel and Distributed Computing*, 9:138–153, 1989.
182. R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, second edition, 1994.
183. R. Elmasri and S. B. Navathe. *SQL: A Relational Database Language*, chapter 7.4, pages 215–219. In [182], second edition, 1994.
184. J. Fairbairn and S. Wray. Code Generation Techniques for Functional Languages. In *Proceedings of the 1986 ACM Conference on LISP and Functional Programming*, pages 94–104, New York, NY, 1986. ACM. Held at MIT, Cambridge, MA.
185. J. Fairbairn and S. Wray. TIM : A Simple Lazy Abstract Machine to Execute Supercombinators. In G. Kahn, editor, *Proceedings of Conference on Functional Programming and Computer Architecture*, pages 34–45. Springer, LNCS 274, 1987.
186. W. Farmer, J. Ramsdell, and R. Watro. A Correctness Proof for Combinator Reduction with Cycles. *ACM Transactions on Programming Languages and Systems*, 12(1):123–134, 1990.
187. M. Feeley, M. Turcotte, and G. LaPalme. Using MultiLisp for Solving Constraint Satisfaction Problems: an Application to Nucleic Acid 3D Structure Determination. *Lisp and Symbolic Computation*, 7:231–247, 1994.
188. M. Felleissen. On the Expressive Power of Programming Languages. *Science of Computer Programming*, 17:35–75, 1991.
189. J. T. Feo, P. J. Miller, S. K. Skedzielewski, S. M. Denton, and C. J. Solomon. Sisal 90. In ??? ???, 1994.
190. A. J. Field and P. G. Harrison. *Functional Programming*. Addison-Wesley, 1988.
191. R. A. Finkel. Large Grain Parallelism - Three Case Studies. In L. Jamieson, D. Gannon, and R. Douglass, editors, *Characteristics of Parallel Algorithms*, pages 21–64, 1987.
192. FISh web-site. <http://www-staff.socs.uts.edu.au/~cbj/FISh>.
193. C. Foisy and E. Chailloux. Caml Flight: A Portable SPMD Extension of ML for Distributed Memory Multiprocessors. In A. P. W. Bohm and J. T. Feo, editors, *High Performance Functional Computing*, pages 83–96, April 1995.
194. S. Fortune and J. Wyllie. Parallelism in Random Access Machines. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 114–118, 1978.
195. C. C. Foster. *Content Addressable Parallel Processors*. Van Nostrand Reinhold Co., New York, 1976.
196. I. Foster. Compositional Parallel Programming Languages. *ACM Transactions on Programming Languages and Systems*, 18(4):454–476, 1996.

197. I. Foster, R. Olson, and S. Tuecke. Productive Parallel Programming: The PCN Approach. *Journal of Scientific Programming*, 1(1):51–66, 1992.
198. I. Foster and S. Taylor. *Strand: New Concepts in Parallel Programming*. Prentice-Hall, 1989.
199. W. D. Frazer and A. C. McKellar. Samplesort: A Sampling Approach to Minimal Storage Tree Sorting. *Journal of the ACM*, 17(3):496–507, 1970.
200. D. P. Freidman and D. S. Wise. CONS Should Not Evaluate its Arguments. In S. Michaelson and R. Milner, editors, *Automata, Languages and Programming: 3rd International Colloquium*. Edinburgh University Press, 1976.
201. M. Frigo. A Fast Fourier Transform Compiler. In *PLDI'99 — Conference on Programming Language Design and Implementation*, Atlanta, GA, May 1999. SCM SIGPLAN. To appear.
202. M. Frigo and S. Johnson. The Fastest Fourier Transform in the West. Technical Report MIT-LCS-TR-728, MIT Laboratory for Computer Science, September 1997.
203. D. Gaertner and W. Kluge. π -RED⁺ – An Interactive Compiling Graph Reduction System for an Applied λ -Calculus. *Journal of Functional Programming*, 6(5):723–757, 1996.
204. E. Gansner and J. Reppy. A Multithreaded Higher-Order User Interface Toolkit. In *User Interface Software*, volume 1 of *Software Trends*, pages 61–80. John Wiley & Sons, 1993.
205. A. Geerling. Program Transformations and Skeletons: Formal Derivation of Parallel Programs. In N. Mirenkov, editor, *pAs'95 Parallel Algorithm/Architecture Synthesis*, pages 250–256. IEEE Computer Society Press, 1995.
206. N. Gehani. Broadcasting Sequential Processes (BSP). *IEEE Transactions on Software Engineering*, 10(4), July 1984.
207. G. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Menchek, and V. Sunderam. *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing*. MIT press, 1994.
208. G. A. Geist, J. A. Kohla, and P. M. Papadopoulos. PVM and MPI: A Comparison of Features. *Calculateurs Paralleles*, 8(2):137–150, 1996.
209. D. Gelernter and N. Carriero. Coordination Languages and Their Significance. *CACM*, 32(2):97–107, February 1992.
210. W. Gentleman. Some Complexity Results for Matrix Computations on Parallel Processors. *Journal of the ACM*, 25:112–115, 1978.
211. A. V. Gerbessiotis and C. Siniolakis. Selection on the Bulk-Synchronous Parallel Model with Applications to Priority Queues. In *Proceedings of the 1996 International Conference on Parallel and Distributed Processing Techniques and Applications*, August 1996.
212. A. V. Gerbessiotis and C. J. Siniolakis. Deterministic Sorting and Randomized Median Finding on the BSP Model. In *Proceedings of the 8th ACM Symposium on Parallel Algorithms and Architectures*, Padova, Italy, June 1996. ACM Press.
213. A. V. Gerbessiotis and C. J. Siniolakis. Primitive Operations on the BSP Model. Technical Report PRG-TR-23-96, Oxford University Computing Laboratory, October 1996.
214. A. V. Gerbessiotis and C. J. Siniolakis. An Experimental Study of BSP Sorting Algorithms: Theory, Practice and Experience. Technical Report PRG-TR-06-97, Oxford University Computing Laboratory, February 1997.
215. A. V. Gerbessiotis, C. J. Siniolakis, and A. Tiskin. Parallel Priority Queue and List Contraction: The BSP approach. In *Europar'97*, LNCS, pages 409–416, Passau, Germany, 1997. Springer-Verlag.

216. A. V. Gerbessiotis and L. G. Valiant. Direct Bulk-Synchronous Parallel Algorithms. *Journal of Parallel and Distributed Computing*, 22(2):251–267, Aug. 1994.
217. A. Giacolone, P. Mishra, and S. Prasad. Facile: A Symmetric Integration of Concurrent and Functional Programming. *International Journal of Parallel Programming*, 18(2), 1989.
218. D. Ginhac, J. Sérot, and J. Dértin. Fast Prototyping of Image Processing Applications Using functional Skeletons on MIMD-DM Architecture. In *IAPR Workshop on Machine Vision Applications*, Chiba, Japan, Nov. 1998.
219. B. Goldberg. *Multiprocessor Execution of Functional Programs*. Phd thesis, Yale University, Department of Computer Science, New Haven, CT, 1988.
220. R. G. Goldsmith, D. L. McBurney, and M. R. Sleep. Parallel Execution of Concurrent Clean on ZAPP. In M. R. Sleep, M. J. Plasmeijer, and M. C. J. van Eekelen, editors, *Term Graph Rewriting: Theory and Practice*, pages 283–302. Wiley, Toronto, 1993.
221. A. J. Gordon. *Functional Programming and Input/Output*. British Computer Society Distinguished Dissertations in Computer Science. Cambridge University Press, 1994.
222. S. Gorlatch. From Transformations to Methodology in Parallel Program Development: A Case Study. *Microprocessing and Microprogramming*, 41:571–588, 1996.
223. S. Gorlatch. Systematic Efficient Parallelization of Scan and Other List Homomorphisms. In L. Bouge, P. Fraigniaud, A. Mignotte, and Y. Robert, editors, *Proceedings of the European Conference on Parallel Processing, Euro-Par'96, LNCS 1124*, pages 401–408. Springer Verlag, 1996.
224. S. Gorlatch. Abstraction and Performance in the Design of Parallel Programs. Technical Report MIP-9803, Universitat Passau, January 1998.
225. S. Gorlatch. Programming with Divide-and-Conquer Skeletons: An Application to FFT. *Journal of Supercomputing*, 12(1–2):85–97, 1998.
226. S. Gorlatch and H. Bischof. A Generic MPI Implementation for a Data-Parallel Skeleton: Formal Derivation and Application to FFT. *Parallel Processing Letters*, 1998.
227. S. Gorlatch and C. Lengauer. Parallelization of Divide-and-Conquer in the Bird-Meertens Formalism. *Formal Aspects of Computing*, 7(6):663–682, 1995.
228. S. Gorlatch and C. Lengauer. (De)Composition Rules for Parallel Scan and Reduction. In *Proceedings of 3rd International Working Conference on Massively Parallel Programming Models (MPPM'97)*, 1998.
229. J. Gosling. Java Intermediate Bytecodes. In *Papers of the ACM SIGPLAN workshop on Intermediate representations*, New York, NY, 1995. ACM.
230. M. W. Goudreau, K. Lang, S. B. Rao, T. Suel, and T. Tsantilas. Towards Efficiency and Portability: Programming with the BSP Model. In *Proceedings of 8th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 1–12, June 1996.
231. C. Grellck. Shared-Memory Multiprocessor Support for SAC. In K. Hammond, T. Davie, and C. Clack, editors, *Proceedings of Implementing Functional Languages (IFL '98) University College, London, UK September 9-11, 1998 (selected papers)*, pages 38–54. Springer, LNCS Vol. 1595, 1999.
232. K. Gremban, G. Miller, and M. Zagha. Performance Evaluation of a New Parallel Preconditioner. Technical Report CMU-CS-94-205, School of Computer Science, Carnegie Mellon University, October 1994.
233. J. R. Gurd, C. C. Kirkham, and I. Watson. The Manchester Prototype Dataflow Computer. *Comm. ACM*, 28(1):34–52, January 1985.

234. M. Haines and A. P. W. Böhm. Task Management, Virtual Shared Memory, and Multithreading in a Distributed Implementation of SISAL. *Lecture Notes in Computer Science*, 694:12 – 23, 1993.
235. G. Hains and C. Foisy. The Data-Parallel Categorical Abstract Machine. In *Parallel Architectures & Languages Europe*. Springer LNCS, 1993.
236. R. Halstead Jr. Multilisp: A Language for Concurrent Symbolic Computation. *ACM Transactions on Programming Languages and Systems*, 7(4):501–538, 1985.
237. J. Hammes and A. P. W. Böhm. Towards a Time and Space Efficient Functional Implementation of a Monte Carlo Photon Transport Code. In *Proceedings of PACT'97*, pages 286–294, 1997.
238. J. Hammes, S. Sur, and A. P. W. Böhm. On the Effectiveness of Functional Language Features: NAS benchmark FT. *Journal of Functional Programming*, 7(1), January 1997.
239. J. P. Hammes, O. Lubeck, and A. P. W. Böhm. Comparing Id and Haskell in a Monte Carlo Photon Transport Code. *Journal of Functional Programming*, 5(3):283–316, July 1995.
240. K. Hammond. 1994.
241. K. Hammond. Parallel Functional Programming - an Introduction. In *Proceedings PASCOS Symposium on Parallel Symbolic Computation*, pages 181–193, 1994.
242. K. Hammond, J. S. M. Jr., and S. L. Peyton Jones. Automatic Spark Strategies and Granularity for a Parallel Graph Reducer. In *Proceedings CONPAR'94*, volume 854 of *LNCS*, pages 521–532. Springer-Verlag, 1994.
243. K. Hammond, H.-W. Loidl, and A. S. Partridge. Improving Granularity for Parallel Functional Programs: A Graphical Winnowing System for Haskell. In *Proceedings of the 1995 Conference on High Performance Functional Computing (HPFC'95), Denver, Colorado*, pages 208–221, 1995.
244. K. Hammond, H.-W. Loidl, and A. S. Partridge. Visualising Granularity in Parallel Programs: A Graphical Winnowing System for Haskell. In A. P. W. Böhm and J. T. Feo, editors, *HPFC'95 – High Performance Functional Computing*, pages 208–221, 1995.
245. K. Hammond, H.-W. Loidl, and P. Trinder. Parallel Cost Centre Profiling. In *Proceedings of 1997 Glasgow FP Workshop, Ullapool, Scotland*, 1997.
246. K. Hammond and S. L. Peyton Jones. Some Early Experiments on the GRIP Parallel Reducer. In *IFL'90 – International Workshop on the Parallel Implementation of Functional Languages*, pages 51–72, Nijmegen, The Netherlands, June 1990.
247. K. Hammond and S. L. Peyton Jones. Profiling Strictness Strategies on the Grip Parallel Reducer. Bericht Nr. 92-19, RWTH Aachen, Fachgruppe Informatik, Aachen, DE, 1992. Proceedings of the 4th International Workshop on the Parallel Implementation of Functional Languages.
248. C. Hankin. *Lambda Calculi*. OUP, 1994.
249. F. Hanna and W. Howells. Parallel Theorem-Proving. In C. Runciman and D. Wakeling, editors, *Applications of functional programming*. UCL Press, 1995.
250. M. R. Hansen and H. Rischel. *Introduction to Programming Using SML*. Addison Wesley Longman, 1999.
251. H. Harrar, H. Keller, D. Lin, and S. Taylor. Parallel Computation of Taylor-Vortex Flows. In *Conference on Parallel Computational Fluid Dynamics*, Stuttgart, Germany, 1991. Elsevier Science Publishers.
252. T. J. Harris. A Survey of PRAM Simulation Techniques. *ACM Computing Surveys*, 26(2):187–206, June 1994.

253. P. Harrison and H. Khoshnevisan. A New Approach to Recursion Removal. *Theoretical Computer Science*, 93:91–113, 1992.
254. P. Harrison and M. Reeve. The Parallel Graph Reduction Machine, Alice. In J. Fasel and R. Keller, editors, *Graph Reduction: Proceedings of a Workshop at Santa F, New Mexico*, pages 181–202, New York, NY, 1987. Springer-Verlag.
255. P. G. Harrison. A Higher-Order Approach to Parallel Algorithms. *Computer Journal*, 35(6):555–566, 1992.
256. P. Hartel, R. Hofman, K. Langendoen, H. Muller, W. Vree, and L. Hertzberger. A Toolkit for Parallel Functional Programming. *Concurrency — Practice and Experience*, 7(8):765–793, 1995.
257. P. Henderson. *Functional Programming: Application and Implementation*. Prentice-Hall, London, UK, 1980.
258. P. Henderson. Functional Geometry. In *Proceedings of the 1982 ACM Symposium on LISP and Functional Programming*. ACM Press, 1982.
259. P. Henderson, G. Jones, and S. Jones. The LispKit Manual. Technical monograph PRG-2, Oxford University Computing Laboratory, 1982. Two volumes.
260. P. Henderson and J. H. Morris. A Lazy Evaluator. In *Proceedings of 3rd ACM Symposium on Principles of Programming Languages*, 1976.
261. C. A. Herrmann and C. Lengauer. Transformation of Divide & Conquer to Nested Parallel Loops. In H. Glaser, P. Hartel, and H. Kuchen, editors, *Programming Languages: Implementation, Logics, and Programs (PLILP'97)*, Lecture Notes in Computer Science 1292, pages 95–109. Springer-Verlag, 1997.
262. M. Heytens and R. S. Nikhil. List Comprehensions in Agna, a Parallel, Persistent Object System. In *FPCA'91 — Conference on Functional Programming Languages and Computer Architectures*, volume 523 of *Lecture Notes in Computer Science*, pages 569–591, Harvard, MA, Aug. 1991. Springer-Verlag.
263. J. Hicks, D. Chiou, B. S. Ang, and Arvind. Performance studies of Id on the Monsoon Dataflow System. *Journal of Parallel and Distributed Computing*, 18(3):273–300, 1993.
264. High Performance Fortran Forum, Houston, Texas. *High Performance Fortran Language Specification*, November 1994. Version 1.1.
265. J. M. D. Hill. The aim is Laziness in a Data-Parallel Language. In K. Hammond and J. T. O'Donnell, editors, *Glasgow Functional Programming Workshop*, Workshops in Computing, pages 83–99. Springer-Verlag, 1993.
266. J. M. D. Hill. *Data-Parallel Lazy Functional Programming*. PhD thesis, Department of Computer Science, Queen Mary & Westfield College, University of London, Sept. 1994.
267. J. M. D. Hill, S. Donaldson, and D. B. Skillicorn. Stability of Communication Performance in Practice: from the Cray T3E to Networks of Workstations. Technical Report PRG-TR-33-97, Programming Research Group, Oxford University Computing Laboratory, October 1997.
268. J. M. D. Hill, S. R. Donaldson, and D. B. Skillicorn. Portability of Performance with the BSPLib communications library. In *Programming Models for Massively Parallel Computers, (MPPM'97)*, London, November 1997. IEEE Computer Society Press.
269. J. M. D. Hill, S. Jarvis, C. Siniolakis, and V. P. Vasilev. Portable and Architecture Independent Parallel Performance Tuning Using a Call-Graph Profiling Tool. In *6th EuroMicro Workshop on Parallel and Distributed Processing (PDP'98)*, pages 286–292. IEEE Computer Society Press, January 1998.
270. J. M. D. Hill, B. McColl, D. C. Stefanescu, M. W. Goudreau, K. Lang, S. B. Rao, T. Suel, T. Tsantilas, and R. Bisseling. BSPLib: The BSP Programming Library. *Parallel Computing*, to appear 1998. see www.bsp-worldwide.org for more details.

271. J. M. D. Hill and D. B. Skillicorn. Lessons Learned from Implementing BSP. *Journal of Future Generation Computer Systems*, March 1998.
272. J. M. D. Hill and D. B. Skillicorn. Practical Barrier Synchronisation. In *6th EuroMicro Workshop on Parallel and Distributed Processing (PDP'98)*, pages 438–444. IEEE Computer Society Press, January 1998.
273. W. D. Hillis. *The Connection Machine*. MIT Press, 1985.
274. W. D. Hillis and G. L. Steele, Jr. Data Parallel Algorithms. *Communications of the ACM*, 29(12):1170–1183, 1986.
275. C. A. R. Hoare. Quicksort. *Computer Journal*, 5(1):10–15, 1962.
276. C. A. R. Hoare. Communicating Sequential Processes. *Communications of the ACM*, 21:666–671, Aug. 1978.
277. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall International, 1985.
278. J. S. Huang and Y. C. Chow. Parallel Sorting and Data Partitioning by Sampling. In *IEEE Computer Society's Seventh International Computer Software & Applications Conference (COMPSAC'83)*, pages 627–631, Nov. 1983.
279. Y. Huang and W. F. McColl. Analytical Inversion of General Tridiagonal Matrices. *Journal of Physics, A: Mathematics and General.*, 30:7919–7933, November 1997.
280. Y. Huang and W. F. McColl. Two-Way BSP Algorithm for Tridiagonal Systems. *Future Generation Computer Systems*, 13:337–347, March 1998.
281. P. Hudak. Para-functional programming. *IEEE Computer*, 19:60–71, 1986.
282. P. Hudak et al. Report on the Programming Language Haskell: A Non-Strict, Purely, Functional Language. *Transactions on Programming Languages and Systems*, 27(5), March 1992.
283. P. Hudak and B. Goldberg. Serial Combinators: Optimal Grains for Parallelism. In *Proceedings of the Conference on Functional Programming and Computer Architecture*, volume 201 of *Lecture Notes in Computer Science*, pages 382–399. Springer, 1985.
284. R. J. M. Hughes. Graph Reduction with Super Combinators. Technical Monograph PRG-28, Oxford University Computer Laboratory, 1982.
285. R. J. M. Hughes. Super-Combinators - A New Implementation Technique for Applicative Languages. In *Proceedings of ACM Conference on LISP and Functional Programming, Pittsburgh, PA*, pages 1 – 19, 1982.
286. R. J. M. Hughes. Why Functional Programming Matters. *The Computer Journal*, 32(2):98–107, 1989.
287. S. Hwang and D. Rushall. The ν -STG Machine: A Parallelized Spineless Tagless Graph Reduction Machine in a Distributed Memory Architecture. Bericht Nr. 92-19, RWTH Aachen, Fachgruppe Informatik, Aachen, DE, 1992.
288. I. P. de Guzman and P.G. Harrison and E. Medina. Pipelines for Divide-and-Conquer Functions. *The Computer Journal*, 36(3), 1993.
289. In W. Kluge, editor, *Proceedings of 8th International Workshop on the Implementation of Functional Languages*, volume 1268 of *Lecture Notes in Computer Science*. Springer Verlag, 1996.
290. In K. Hammond, T. Davie, and C. Clack, editors, *Proceedings of 10th International Workshop on the Implementation of Functional Languages*, volume 1595 of *Lecture Notes in Computer Science*. Springer Verlag, 1998.
291. R. Irvin. *Performance Tool for High-Level Parallel Programming Languages*. PhD thesis, University of Wisconsin-Madison, November 1995.
292. J. Irwin, J.-M. Loingtier, J. R. Gilbert, and G. Kiczales. Aspect-Oriented Programming of Sparse Matrix Code. *Lecture Notes in Computer Science*, 1343:249–??, 1997.

293. ISO. *Information Processing Systems: Open Systems Interconnection: Estelle: A Formal Description Technique Based on an Extended State Transition Model*. ISO 9074:1989. International Organization for Standards, 1989.
294. ISO. *Information Processing Systems: Open Systems Interconnection: LOTOS: A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour*. ISO 8807:1989(E). International Organization for Standards, 1989.
295. R. H. E. Ito, T.; Halstead. *Parallel Lisp: Languages and Systems*.
296. T. Ito. Efficient Evaluation Strategies for Structured Concurrency Constructs in Parallel Scheme Systems. In *PSLS'95 — International Workshop on Parallel Symbolic Languages and Systems*, number 1068 in Lecture Notes in Computer Science, pages 22–52. Springer-Verlag, 1995.
297. J. Caesar. *De Bellum Gallicum*. Rome, 45 BP.
298. L. Jamieson, D. Gannon, and R. Douglass. *The Characteristics of Parallel Algorithms*. MIT Press, 1987.
299. P. Jansson and J. Jeuring. PolyP - a Polytypic Programming Language Extension. In *POPL '97: The 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 470–482. ACM Press, 1997.
300. C. B. Jay. Matrices, Monads and the Fast Fourier Transform. In *Proceedings of the Massey Functional Programming Workshop 1994*, pages 71–80, 1994.
301. C. B. Jay. A Semantics for Shape. *Science of Computer Programming*, 25:251–283, 1995.
302. C. B. Jay. Shape in Computing. *ACM Computing Surveys*, 28(2):355–357, 1996.
303. C. B. Jay. Poly-Dimensional Regular Arrays in FISH, April 1998. superceded.
304. C. B. Jay. The FISH language definition. <http://www-staff.socs.uts.edu.au/~cbj/Publications/fishdef.ps.gz>, 1998.
305. C. B. Jay. Partial Evaluation of Shaped Programs: Experience With FISH. In O. Danvey, editor, *ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation (PEPM '99) San Antonio, Texas, January 22-23, 1999: Proceedings*, pages 147–158. BRICS, 1999.
306. C. B. Jay, G. Bellè, and E. Moggi. Functorial ML. *Journal of Functional Programming*, 1999. to appear.
307. C. B. Jay, M. I. Cole, M. Sekanina, and P. A. Steckler. A Monadic Calculus for Parallel Costing of a Functional Language of Arrays. In C. Lengauer, M. Griehl, and S. Gorlatch, editors, *Euro-Par'97 Parallel Processing*, volume 1300 of *Lecture Notes in Computer Science*, pages 650–661. Springer-verlag, August 1997.
308. C. B. Jay and P. A. Steckler. The Functional Imperative: Shape! In C. Hankin, editor, *Programming languages and systems: 7th European Symposium on Programming, ESOP'98 Held as part of the joint european conferences on theory and practice of software, ETAPS'98 Lisbon, Portugal, March/April 1998*, volume 1381 of *Lecture Notes in Computer Science*, pages 139–53. Springer Verlag, 1998.
309. J. I. Jenkins, M. A.; Glasgow. A Logic Basis for Nested Array Data Structures. *Computer Languages Journal*, 14(1):35 – 51, 1989.
310. G. Johnson and J. Genetti. Volume Rendering of Large Datasets on the Cray T3D. In *Spring Proceedings (Cray User Group)*, 1996.
311. T. Johnsson. Efficient Compilation of Lazy Evaluation. In *Proceedings of ACM Conference on Compiler Construction*, pages 58–69, Montreal, 1984.
312. T. Johnsson. Lambda Lifting: Transforming Programs to Recursive Equations. In J.-P. Jouannaud, editor, *Functional Programming Languages and Computer Architecture*, pages 190–203. Springer-Verlag, Berlin, DE, 1985.

313. T. Johnsson. *Compiling Lazy Functional Languages*. Phd thesis, Department of Computer Science, Chalmers University of Technology, Göteborg, SE, 1987.
314. G. Jones and M. Sheeran. Circuit Design in Ruby. *Formal methods in VLSI Design*, pages 13–70, 1990.
315. M. Jones. A System of Constructor Classes: Overloading and Implicit Higher-Order Polymorphism. *Journal of Functional Programming*, 5(1):1–37, January 1995.
316. M. P. Jones. A Theory of Qualified Types. In *Proceedings of ESOP'92*, volume 582 of *Lecture Notes in Computer Science*, pages 287–306. Springer, 1992.
317. S. Junaidu. *A Parallel Functional Language Compiler for Message Passing Multicomputers*. PhD thesis, School of Mathematical and Computational Sciences, University of St. Andrews, March 1998.
318. S. Junaidu, A. Davie, and K. Hammond. Naira: A Parallel² Haskell Compiler. In *IFL'97 — International Workshop on the Implementation of Functional Languages*, volume 1467 of *Lecture Notes in Computer Science*, pages 215–231, University of St. Andrews, Scotland, September 10–12, 1997.
319. H. Jung, L. Kirousis, and P. Spirakis. Lower Bounds and Efficient Algorithms for Multiprocessor Scheduling of DAGs with Communication Delays. In *Proceedings of ACM Symposium on Parallel Architectures and Algorithms*, pages 254–264, 1989.
320. B. H. H. Juurlink and H. A. G. Wijshoff. Communication Primitives for BSP Computers. *Information Processing Letters*, 58:303–310, 1996.
321. G. Kahn and D. MacQueen. Coroutines and Networks of Parallel Processes. In *IFIP 77*. North Holland, 1977.
322. E. Karlsen. The UniForM Concurrency Toolkit and its Extensions to Concurrent Haskell. In *Glasgow Workshop on Functional Programming*, Ullapool, Scotland, September 1997. Department of Computing Science, University of Glasgow.
323. E. Karlsen. The UniForM User Interaction Manager. Technical report, University Bremen, Germany, March 1998.
324. E. Karlsen. The UniForM WorkBench — a Higher Order Tool Integration Framework. In *AFM'98 — International Workshop on Current Trends in Applied Formal Methods*, Boppard, Germany, October 7–9, 1998.
325. E. Karlsen and S. Westmeier. Using Concurrent Haskell to Develop Views over an Active Repository. In *IFL'97 — International Workshop on the Implementation of Functional Languages*, volume 1467 of *Lecture Notes in Computer Science*, pages 285–303, September 10–12, St. Andrews, Scotland, 1997. Springer-Verlag.
326. O. Kaser, C. Ramakrishnan, I. V. Ramakrishnan, and R. C. Sekar. EQUALS — A Fast Parallel Implementation of a Lazy Language. *Journal of Functional Programming*, 7(2):183 – 217, 1997.
327. G. Keller and M. Simons. A Calculational Approach to Flattening Nested Data Parallelism in Functional Languages. In J. Jaffar and R. H. C. Yap, editors, *Concurrency and Parallelism, Programming, Networking, and Security: Second Asian Computing Science Conference, ASIAN'96*, volume 1179 of *Lecture Notes in Computer Science*, pages 234–243. Springer-Verlag, 1996.
328. P. H. J. Kelly. *Functional Programming for Loosely-coupled Multiprocessors*. Research Monographs in Parallel and Distributed Computing. Pitman, London and MIT Press, Boston, 1989.
329. M. H. G. Kessler. Uniqueness and Lazy Graph Copying - Copyright for the Unique. In *Proceedings of the 6th International Workshop on the Implementation of Functional Languages*, Norwich, UK, 1994. University of East Anglia.

330. M. H. G. Kesseler. Constructing Skeletons in Clean: the Bare Bones. In *HPFC'95 — Conference on High Performance Functional Computing*, pages 182–192, Denver, CO, April 10–12, 1995.
331. M. H. G. Kesseler. *The Implementation of Functional Languages on Parallel Machines with Distributed Memory*. PhD thesis, Wiskunde en Informatica, Katholieke Universiteit van Nijmegen, The Netherlands, 1996.
332. R. Kieburtz. Performance Measurements of a G-Machine Implementation. Technical report, Oregon Graduate Center, University of Oregon, 1988.
333. D. King, J. Hall, and P. Trinder. A Strategic Profiler for Glasgow Parallel Haskell. In *Draft Proceedings of 10th International workshop on the Implementation of Functional Languages (IFL '98), London, England*, pages 465–475, 1998.
334. H. Kingdon, D. Lester, and G. Burn. The HDG-Machine: A Highly Distributed Graph-Reducer for a Transputer Network. *Computer Journal*, 34(4):290–301, 1991.
335. W. Kluge. Cooperating Reduction Machines. *IEEE Transactions on Computers*, C-32(11):1002 – 1012, 1983.
336. W. Kluge. A User's Guide for the Reduction System π -RED. Internal Report 9409, Department of Computer Science, University of Kiel, 1994.
337. W. E. Kluge. The Architecture of a Reduction Language Machine — Hardware Model. Internal Report ISF 79.03, Institut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, Germany, August 1979.
338. D. E. Knuth. *The Art of Computer Programming. Volume III: Sorting and Searching*. Addison-Wesley, 1973.
339. H. Kobayashi. *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*. Addison-Wesley, 1978.
340. P. W. M. Koopman, M. C. J. D. van Eekelen, E. G. J. M. H. Nöcker, J. E. W. Smetsers, and M. J. Plasmeijer. The ABC-machine: A Sequential Stack-based Abstract Machine for Graph Rewriting. Technical Report 90-22, Department of Computer Science, University of Nijmegen, 1990.
341. H. F. Korth and A. Silberschatz. *Database System Concepts*. McGraw-Hill, 1986.
342. J. Koster and R. H. Bisseling. An Improved Algorithm for Parallel Sparse LU Decomposition on a Distributed-Memory Multiprocessor. In J. G. Lewis, editor, *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, pages 397–401. SIAM, Philadelphia, PA, 1994.
343. H. Kuchen, R. Plasmeijer, and H. Stoltze. Distributed Implementation of a Data Parallel Functional Language. In *Parallel Architectures & Languages Europe*, pages 464–477. LNCS 817, Springer Verlag, 1994.
344. V. Kumar, A. Grama, A. Gupta, and G. Karypis. *Introduction to Parallel Computing, Design and Analysis of Algorithms*. The Benjamin/Cummings Publishing Company, Inc., California, 1994.
345. H. Kung. The Structure of Parallel Algorithms. In M. Yovits, editor, *Advances in Computers, Volume 19*, pages 65–112. Academic Press, 1980.
346. R. E. Ladner and M. J. Fischer. Parallel Prefix Computation. *JACM*, 27(4):831–838, Oct. 1980.
347. P. Landin. The Mechanical Evaluation of Expressions. *Computer Journal*, 6(4):308–320, January 1964.
348. K. Langendoen and P. H. Hartel. FCG: A Code Generator for Lazy Functional Languages. In U. Kastens and P. Pfahler, editors, *Compiler Construction; 4th International Conference CC'92 Proceedings*, pages 278–296, Berlin, Germany, 1992. Springer-Verlag.

349. Lawrence Livermore National Laboratories. Sisal Performance Data. <http://www.llnl.gov/sisal/PerformanceData.html>, June 1998.
350. D. S. Lecomber, K. R. Sujithan, and J. M. D. Hill. Architecture-Independent Locality Analysis and Efficient PRAM Simulations. In *High Performance Computing and Networking (HPCN'97)*, volume 1225. Springer-Verlag, April 1997.
351. D. S. Lecomber, K. R. Sujithan, and C. J. Siniolakis. PRAM Programming In Theory and In Practice. In *Concurrency: Practice and Experience*, 1998.
352. L. M. Leslie et al. A High Resolution Primitive Equations NWP Model for Operations and Research. *Australian Metereological Magazine*, 33:11–35, Mar. 1985.
353. D. Lester. The G-Machine as a Representation of Stack Semantics. In G. Kahn, editor, *Functional Programming Languages and Computer Architecture*, pages 47–59. Springer-Verlag, Berlin, DE, 1987. Lecture Notes in Computer Science 274; Proceedings of Conference held at Portland, OR.
354. D. Lester. Combinator Graph Reduction: A Congruence and its Applications. Technical Monograph PRG-73, Oxford University Computing Laboratory, Oxford, UK, 1989. D. Phil Thesis.
355. D. Lester and G. Burn. An Executable Specification of the HDG-Machine. *International Journal of High Speed Computing*, 5(3):327–378, 1993.
356. C. Lewerentz and T. Lindner. *Formal Development of Reactive Systems, Case Study "Production Cell"*, volume 891 of Lecture Notes in Computer Science. Springer, 1995.
357. H.-W. Loidl. *GranSim User's Guide*, 1996. <http://www.dcs.glasgow.ac.uk/fp/software/gransim/>.
358. H.-W. Loidl. LinSolv: A Case Study in Strategic Parallelism. In *Glasgow Workshop on Functional Programming*, Ullapool, Scotland, September 15–17, 1997.
359. H.-W. Loidl. *Granularity in Large-Scale Parallel Functional Programs*. Phd thesis, University of Glasgow, Department of Computing Science, Glasgow, Scotland, 1998.
360. H.-W. Loidl and K. Hammond. ?? 1996.
361. H.-W. Loidl, R. Morgan, P. Trinder, S. Poria, C. Cooper, S. L. Peyton Jones, and R. Garigliano. Parallelising a Large Functional Program; Or: Keeping LOLITA Busy. In *IFL'97 — International Workshop on the Implementation of Functional Languages*, volume 1467 of *Lecture Notes in Computer Science*, pages 199–214, University of St. Andrews, Scotland, September 10–12, 1997. Springer-Verlag.
362. H.-W. Loidl, P. Trinder, K. Hammond, S. Junaidu, R. Morgan, and S. L. Peyton Jones. Engineering Parallel Symbolic Programs in GPH. *Concurrency — Practice and Experience*, 1999. To appear.
363. R. Loogen, H. Kuchen, K. Indermark, and W. Damm. Distributed Implementation of Programmed Graph Reduction. In K. Odijk, M. Rem, and J.-C. Syre, editors, *PARLE '89, Parallel Architectures and Languages Europe, Volume I: Parallel Architectures, Eindhoven, NL*, LNCS 365, pages 136–157. Springer-Verlag, Berlin, DE, 1989.
364. F. Loulergue. BSML : Programmation BSP Purement Fonctionnelle. In *REN-PAR'10, Rencontres Francophones du Parallisme*. Université de Strasbourg, Juin 1998.
365. F. Loulergue and G. Hains. Parallel Functional Programming with Explicit Processes: Beyond SPMD. In C. Lengauer, M. Griebel, and S. Gorlatch, editors, *EuroPar'97 Parallel Processing*, number 1300 in LNCS, pages 530–537, Passau, Germany, August 1997. Springer.

366. F. Loulergue and G. Hains. An Introduction to BSL. Rapport de Recherche 98-09, Université d'Orléans, LIFO, Orléans, France, september 1998.
367. W. Luk and G. Jones. The Derivation of Regular Synchronous Circuits. In *Proceedings of the International Conference on Systolic Arrays, San Diego*, 1988.
368. C. Lüth, E. Karlsen, Kolyang, S. Westmeier, and B. Wolff. HOL-Z in the UniForM WorkBench — a Case Study in Tool Integration for Z. In *ZUM'98 — 11th International Conference of Z Users*, September 1998.
369. G. Mago. A Network of Multiprocessors to Execute Reduction Languages. *International Journal of Computer and Information Science*, 8(5):349 – 471, 1979.
370. M. Marin. Billiards and Related Systems On The Bulk Synchronous Parallel Model. In *ACM/IEEE/SCS 11th Workshop on Parallel and Distributed Simulation (PADS'97)*, Vienna, Austria, June 1997. IEEE-CS Press.
371. M. Marin. Priority Queue Operations on EREW-PRAM. In *EuroPar'97*, LNCS, pages 417–420, Passau, Germany, 1997. Springer-Verlag.
372. M. Marin. Asynchronous (Time-Warp) Versus Synchronous (Event-Horizon) Simulation Time Advance in BSP. In *EuroPar'98*, volume 1470 of LNCS, pages 897–905. Springer-Verlag, September 1998.
373. J. S. Mattson. *An Efficient Speculative Evaluation Technique for Parallel Supercombinator Graph Reduction*. PhD thesis, Department of Computer Science and Engineering, UC San Diego, 1993.
374. H. Mattsson, H. Nilsson, and C. Wikström. Mnesia — A Distributed Robust DBMS for Telecommunications Applications. In *International Switching Symposium*, Toronto, Canada, 1997.
375. D. McAllester and K. Arkondas. Walther Recursion. In M. Robbie and J. Slaney, editors, *CADE 13*. Springer, 1996.
376. D. McBurney and M. Sleep. Transputer-Based Experiments with the Zapp Architecture. In J. de Bakker, A. Nijman, and P. Treleaven, editors, *PARLE: Parallel Architectures and Languages Europe (Volume 1: Parallel Architectures)*, LNCS 258, pages 242–259. Springer-Verlag, Berlin, DE, 1987.
377. D. McBurney and M. Sleep. Experiments with the ZAPP Virtual Tree Architecture. In T. Fountain and M. Shute, editors, *Multiprocessor Computer Architecture*, pages 83–99. North Holland, Amsterdam, 1990.
378. D. McBurney and M. Sleep. Graph Rewriting as a Computational Model. In A. Yonezawa and T. Ito, editors, *Concurrency: Theory, Language and Architecture*. Springer-Verlag, New York, NY, 1991. Lecture Notes in Computer Science 491.
379. W. F. McColl. General Purpose Parallel Computing. In A. Gibbons and P. Spirakis, editors, *Lectures on Parallel Computation*, Cambridge International Series on Parallel Computation, pages 337–391. Cambridge University Press, Cambridge, 1993.
380. W. F. McColl. BSP Programming . In G. E. Blelloch, K. M. Chandy, and S. Jagannathan, editors, *Specification of Parallel Algorithms. Proceedings of DIMACS Workshop, Princeton, May 9-11, 1994*, volume 18 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 21–35. American Mathematical Society, 1994.
381. W. F. McColl. Scalable Parallel Computing: A Grand Unified Theory and its Practical Development. In *Proceedings of IFIP World Congress*, volume 1, pages 539–546, Hamburg, August 1994.
382. W. F. McColl. Scalable Computing. In J. van Leeuwen, editor, *Computer Science Today: Recent Trends and Developments*, volume 1000 of *Lecture Notes in Computer Science*, pages 46–61. Springer-Verlag, 1995.

383. W. F. McColl and Q. Miller. The GPL Language: Reference Manual . Technical report, ESPRIT GEPPCOM Project, Oxford university Computing Laboratory, Oct. 1995.
384. J. McGraw et al. SISAL: Streams and Iteration in a Single Assignment Language: Reference Manual Version 1.2. Technical Report Memo M-146, Rev. 1, Lawrence Livermore National Laboratory, 1985.
385. L. Meertens. Algorithmics – Towards Programming as a Mathematical Activity. In *Proceedings of CWI Symposium on Mathematics and Computer Science*, pages 289–334. North-Holland, 1986.
386. L. Meertens. Calculate Polytypically! In H. Kuchen and S. Swierstra, editors, *Programming Languages: Implementations, Logics, and Programs: 8th International Symposium, PLILP'96, Proceedings*, volume 1140 of *Lecture Notes in Computer Science*. Springer, 1996.
387. K. Mehlhorn and U. Vishkin. Randomized and Deterministic Simulations of PRAMs by Parallel Machines with Restricted Granularity of Parallel Memories. *Acta Informatica*, 21:339–374, 1984.
388. Message Passing Interface Forum. MPI: A Message Passing Interface. In *Proceedings of Supercomputing '93*, pages 878–883. IEEE Computer Society Press, 1993.
389. G. Michaelson, A. Ireland, and P. King. Towards a Skeleton Based Parallelising Compiler for SML. In C. Clack, K. Hammond, and T. Davie, editors, *Proceedings of 9th International Workshop on the Implementation of Functional Languages*, pages 539–546, 1997.
390. G. Michaelson and N. Scaife. Prototyping a Parallel Vision System in Standard ML. *Journal of Functional Programming*, 5(3):345–382, July 1995.
391. R. Miller. A Library for Bulk Synchronous Parallel Programming. In *Proceedings of the BCS Parallel Processing Specialist Group workshop on General Purpose Parallel Computing*, pages 100–108, December 1993.
392. R. Miller. *Two Approaches to Architecture-Independent Parallel Computation*. D.Phil thesis, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, Michaelmas Term 1994.
393. R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
394. R. Milner. Calculi for Interaction. *Acta Informatica*, 33, 1996.
395. R. Milner, J. Parrow, and D. Walker. A Calculus of Mobile Processes I & II. *Information and Computation*, 100(1):1–77, 1992.
396. R. Milner, M. Tofte, and B. Harper. *The Definition of Standard ML*. MIT Press, 1990.
397. R. Mirani and P. Hudak. First-Class Schedules and Virtual Maps. In *Proceedings of FPCA '95, La Jolla, CA*, pages 78–85, June 1995.
398. J. Misra. Powerlist: A Structure for Parallel Recursion. *ACM TOPLAS*, 16(6), 1994.
399. S. Mitrovic and M. Trobina. Computer Vision Algorithms in Sisal. In *Sisal'93 – Sisal Users Conference*, number CONF 9310206 in LLNL Report, pages 114–119, San Diego, CA, Oct. 1993.
400. E. Moggi. Notions of Computation and Monads. *Information and Computation*, 93(1):55–92, 1991.
401. E. Mohr, D. Kranz, and R. Halstead Jr. Lazy Task Creation: A Technique for Increasing the Granularity of Parallel Programs. *IEEE Transactions on Parallel and Distributed Systems*, 2(3):264–280, July 1991.
402. E. Mohr, D. Kranz, and R. H. H. Jr. A technique of increasing the granularity of parallel programs. In *Proceedings of the ACM Conference on Lisp and Functional Programming*, pages 185–197. ACM, 1990.

403. P. Moller-Nielsen and J. Staunstrup. Problem-Heap: A Paradigm for Multiprocessor Algorithms. *Parallel Computing*, 4:63–74, 1987. (Algorithms with a divide and conquer feel.)
404. P. B. Monk, A. K. Parrott, and P. J. Wesson. A Parallel Finite Element Method for Electro-Magnetic Scattering. In *COMPEL*, volume 13, pages Supp.A:237–242, 1994.
405. D. Moon. The Architecture of the Symbolics 3600. In *Proceedings of 12th International Symposium on Computer Architecture, Boston*, pages 76 – 83, 1985.
406. D. Morais. Id World: An Environment for the Development of Dataflow Programs Written in Id. Technical Report MIT-LCS-TR-365, Laboratory of Computer Science, M.I.T., May 1986.
407. Z. G. Mou and P. Hudak. An Algebraic Model for Divide-and-Conquer and Its Parallelism. *Journal of Supercomputing*, 2:257–278, 1988.
408. A. Mycroft. *Abstract Interpretation and Optimising Transformations for Applicative Programs*. PhD thesis, Department of Computer Science, University of Edinburgh, 1981.
409. P. Nelson and L. Snyder. Programming Paradigms for Nonshared Memory Parallel Computers. In L. Jamieson, D. Gannon, and R. Douglass, editors, *Characteristics of Parallel Algorithms*, pages 3–20, 1987.
410. M. Nibhanupudi, C. Norton, and B. Szymanski. Plasma Simulation on Networks of Workstations Using the Bulk Synchronous Parallel Model. In *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, Athens, GA, November 1995.
411. F. Nielson, H. R. Nielson, and C. L. Hankin. *Principles of Program Analysis*. Springer, 1999.
412. F. Nielson, H. R. Nielson, R. R. Hansen, and J. G. Jensen. Validating Firewalls in Mobile Ambients. In *Proceedings of CONCUR'99*, Lecture Notes in Computer Science. Springer, 1999.
413. H. R. Nielson, T. Amtoft, and F. Nielson. Behaviour Analysis and Safety Conditions: A Case Study in CML. In *Proceedings of FASE'98*, number 1382 in Lecture Notes in Computer Science, pages 255–269. Springer, 1998.
414. H. R. Nielson and F. Nielson. Communication Analysis for Concurrent ML. In F. Nielson, editor, *ML with Concurrency: Design, Analysis, Implementation and Application*, Monographs in Computer Science, pages 185–235. Springer, 1997.
415. H. R. Nielson, F. Nielson, and T. Amtoft. Polymorphic Subtyping for Effect Analysis: The Static Semantics. In M. Dam, editor, *Analysis and Verification of Multiple-Agent Languages*, number 1192 in Lecture Notes in Computer Science, pages 141–171. Springer, 1997.
416. R. S. Nikhil. Id (Version 90.1) Reference Manual. Technical Report CSG Memo 284-2, Laboratory for Computer Science, MIT, July 1991.
417. R. S. Nikhil, Arvind, and J. Hicks. pH Language Proposal (Preliminary), 1st. September 1993. Electronic communication on comp.lang.functional.
418. R. S. Nikhil, G. M. Papadopoulos, and Arvind. *T: A Multithreaded Massively Parallel Architecture. In *19th. ACM Annual Symposium on Computer Architecture*, pages 156–167, 1992.
419. R. S. Nikhil, U. Ramachandran, J. M. Rehg, R. H. Halstead, Jr., C. F. Joerg, and L. Kontothanassis. Stampede — A Programming System for Emerging Scalable Interactive Multimedia Applications. Technical Report CRL 98/1, Cambridge Research Laboratory, May 1998.
420. E. G. J. M. H. Nöcker and M. J. Plasmeijer. Combinator Reduction on a Parallel G-Machine. Technical report, Department of Computer Science, Uni-

- versity of Nijmegen, 1986.
421. E. G. J. M. H. Nöcker, M. J. Plasmeijer, and J. E. W. Smetsers. The Parallel ABC Machine. In H. Glaser and P. Hartel, editors, *Proceedings of the Workshop on the Parallel Implementation of Functional Languages*, pages 351–382, Southampton, UK, 1991. Department of Electronics and Computer Science, University of Southampton.
 422. E. G. J. M. H. Nöcker, J. E. W. Smetsers, M. C. J. D. van Eekelen, and M. J. Plasmeijer. Concurrent Clean. In E. Aarts and J. van Leeuwen, editors, *Parallel Architectures and Languages Europe PARLE91*, volume 505 of *LNCS*, pages 202–219. Springer-Verlag, New York, NY, 1991.
 423. E. G. J. M. H. Nöcker, J. E. W. Smetsers, M. C. J. D. van Eekelen, and M. J. Plasmeijer. Concurrent Clean. In *Proceedings of Parallel Architectures and Languages Europe*, number 505/506 in *LNCS*, pages 202–219, Eindhoven, The Netherlands, June 1991. Spinger-Verlag.
 424. B. Nordström, K. Petersson, and J. M. Smith. *Programming in Martin-Löf's Type Theory — An Introduction*. Oxford University Press, 1990.
 425. J. O'Donnell. Functional Microprogramming for a Data Parallel Architecture. In *Glasgow Workshop on Functional Programming*, volume 89/R4, pages 124–145. Computing Science Department, University of Glasgow, 1988.
 426. J. O'Donnell. MPP Implementation of Abstract Data Parallel Architectures for Declarative Programming Languages. In *Frontiers '88: Proceedings of the Secons Symposium on the Frontiers of Massively Parallel Computation*, pages 629–636. IEEE Computer Society, 1988.
 427. J. O'Donnell. Data-Parallel Implementation of Extensible Sparse Functional Arrays. In *Parallel Architectures and Languages Europe*, volume 694 of *LNCS*, pages 68–79. Springer, June 1993.
 428. J. O'Donnell. The Comb Communication Algorithm for Fine Grain Data Parallel Systems. Technical report, University of Glasgow, 1998.
 429. J. O'Donnell and G. Ruenger. A Coordination Level Functional Implementation of the Hierarchical Radiosity Algorithm. In *Glasgow Workshop on Functional Programming*, Ullapool, Scotland, September 1997. Department of Computing Science, University of Glasgow.
 430. J. O'Donnell and G. Rünger. A Case Study in Parallel Program Derivation: the Heat Equation Algorithm. In *Functional Programming, Glasgow 1994*, Workshops in Computing, pages 167–183. Springer, 1994.
 431. J. O'Donnell and G. Rünger. Formal Derivation of a Parallel Binary Addition Circuit. Technical Report TR-1995-19, University of Glasgow, 1995.
 432. P. O'Hearn and R. Tennent, editors. *Algol-Like Languages, Vols I and II*. Progress in Theoretical Computer Science. Birkhauser, 1997.
 433. R. B. Osborne. Speculative Computation in Multilisp. In T. Ito and R. H. Halstead, editors, *Parallel Lisp: Languages and Systems*, volume 441 of *Lecture Notes in Computer Science*, pages 103 – 137. Springer, 1990.
 434. F. O. Osoba and F. A. Rabhi. A Parallel Multigrid Skeleton Using BSP. In *EuroPar'98*, volume 1470 of *LNCS*, pages 704–708. Springer-Verlag, September 1998.
 435. J. Palsberg and C. B. Jay. The Essence of the Visitor Pattern. Technical Report 05, University of Technology, Sydney, 1997. COMPSAC'98, to appear.
 436. P. Panangaden and J. Reppy. The Essence of Concurrent ML. In F. Nielson, editor, *ML with Concurrency: Design, Analysis, Implementation and Application*, Monographs in Computer Science, pages 5–29. Springer, 1997.
 437. G. Papadopoulos. Implementation of a General Purpose Dataflow Multi-processor. Technical Report 432, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1988.

438. D. Parkinson and J. Litt, editors. *Massively Parallel Computing with the DAP*. Research Monographs in Parallel and Distributed Computing. Pitman, 1990.
439. D. Parrott and C. Clack. A Common Graphical Form. In J. Darlington and R. Dietrich, editors, *Proceedings of the Phoenix Workshop on Declarative Programming*, number RN/91/27 in Workshops in Computing, pages 224–238. Springer-Verlag, 1992.
440. A. S. Partridge. *Speculative Evaluation in Parallel Implementations of Lazy Functional Languages*. PhD thesis, Department of Computer Science, University of Tasmania, 1991.
441. A. S. Partridge and A. Dekker. Speculative Parallelism in a Distributed Graph Reduction Machine. In *Proceedings of 22nd Annual Hawaii International Conference on System Sciences, Vol. 2*, pages 771 – 779, 1989.
442. L. C. Paulson. *Logic and Computation — Interactive proof with Cambridge LCF*. Cambridge University Press, 1987.
443. S. Pelagatti. Compiling and Supporting skeletons on MPP. In *Proceedings of MPPM97*. IEEE Computer Society Press, 1997. to appear.
444. S. Pelagatti. *Structured Development of Parallel Programs*. Taylor and Francis, 1998.
445. P. Pepper. Deductive Derivation of Parallel Programs. In *Parallel Algorithm Derivation and Program Transformation*, chapter 1, pages 1–53. Kluwer Academic Publishers, 1993. Also: Technical Report 92-23, Technische Universitat Berlin, July 1992.
446. P. Pepper and M. Südholt. Deriving Parallel Numerical Algorithms Using Data Distribution Algebras: Wang’s Algorithm. In *HICSS’97 — 30th Hawaii International Conference on System Sciences*, Hawaii, USA, January 7–10, 1997. IEEE.
447. F. Peters. Tree Machines and Divide and Conquer Algorithms. In *Proceedings CONPAR 81, LNCS 111*, pages 25–36. Springer-Verlag, 1981.
448. J. C. Peterson, J. Hammond, L. Augustsson, B. Boutel, F. W. Burton, J. Fasel, A. D. Gordon, R. J. M. Hughes, P. Hudak, T. Johnsson, M. P. Jones, E. Meijer, S. L. Peyton Jones, A. Reid, and P. L. Wadler. *Report on the Non-Strict Functional Language, Haskell, Version 1.4*, 1997.
449. F. Petrini. Total-Exchange on Wormhole K-ary N-Cubes with Adaptive Routing. In *12th International Parallel Processing Symposium & 9th Symposium on Parallel and Distributed Processing*, April 1998.
450. S. L. Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice-Hall, London, UK, 1987.
451. S. L. Peyton Jones. Parallel Implementations of Functional Programming Languages. *The Computer Journal*, 32(2):175–186, 1989.
452. S. L. Peyton Jones. Implementing Lazy Functional Languages on Stock Hardware: The Spineless Tagless G-machine. *Journal of Functional Programming*, 2(2), 1992.
453. S. L. Peyton Jones. Compiling Haskell by Program Transformation: A Report from the Trenches. In *ESOP’96 — European Symposium on Programming*, volume 1058 of *Lecture Notes in Computer Science*, pages 18–44, Linköping, Sweden, April 22–24, 1996. Springer-Verlag.
454. S. L. Peyton Jones, C. Clack, and N. Harris. GRIP: A Parallel Graph Reduction Machine. In *Proceedings Workshop on Implementation of Functional Languages*, pages 59–91, 1985. Technical Report 17.
455. S. L. Peyton Jones, C. Clack, and J. Salkild. High-Performance Parallel Graph Reduction. In K. Odijk, M. Rem, and J.-C. Syre, editors, *PARLE ’89: Parallel Languages and Architectures Europe, volume 1*, pages 193–206. Springer-Verlag, New York, NY, 1989.

456. S. L. Peyton Jones, C. Clack, J. Salkild, and M. Hardie. GRIP - A High-Performance Architecture for Parallel Graph Reduction. In G. Kahn, editor, *Functional Programming Languages and Computer Architecture*, pages 98–112. Springer-Verlag, Berlin, DE, 1987.
457. S. L. Peyton Jones, A. Gordon, and S. Finne. Concurrent Haskell. In *POPL'96 — Symposium on Principles of Programming Languages*, pages 295–308, St Petersburg, Florida, January 1996. ACM.
458. S. L. Peyton Jones and M. Joy. FLIC - a Functional Language Intermediate Code. Research Report 148, Department of Computer Science, University of Warwick, Coventry, UK, 1989.
459. S. L. Peyton Jones, E. Meijer, and D. Leijen. Scripting COM Components in Haskell. In *Fifth Intl Conf on Software Reuse*, Jun 1998.
460. S. L. Peyton Jones and J. Salkild. The Spineless Tagless G-Machine. In *Proceedings of the Conference on Functional Programming Languages and Computer Architecture '89, Imperial College, London*, pages 184–201, New York, NY, 1989. ACM.
461. M. Pil. First Class File I/O. In Kluge [289], pages 233–246.
462. M. Pil. Dynamic Types and Type Dependent Functions. In Hammond et al. [290], pages 171–188.
463. N. Pippenger. Pure Versus Impure Lisp. *ACM Transactions on Programming Languages and Systems*, 19(2), March 1997.
464. A. M. Pitts. A Co-Induction Principle for Recursively Defined Domains. *Journal of Logic and Computation*, 1992.
465. A. M. Pitts and I. D. B. Stark. Operational Reasoning for Functions with Local State. In A. D. Gordon and A. M. Pitts, editors, *Higher Order Operational Techniques in Semantics*, Publications of the Newton Institute. Cambridge University Press, 1998.
466. M. J. Plasmeijer and M. C. J. D. van Eekelen. *Functional Programming and Parallel Graph Rewriting*. Addison-Wesley Publishers Ltd., 1993.
467. E. Poll and S. Thompson. Review of [51]. *Journal of Functional Programming*, To appear.
468. F. Preparata and J. Vuillemin. The Cube-Connected Cycles: A Versatile Network for Parallel Computation. *CACM*, 24(5):300–309, 1981.
469. W. Press and others. *Numerical Recipes, the Art of Scientific Programming*. Cambridge University Press, 1986.
470. D. Pritchard and others. Practical Parallelism Using Transputer Arrays. In J. de Bakker, editor, *PARLE 87 (LNCS 258)*. Springer-Verlag, 1987.
471. M. J. Quinn. *Parallel Computing: Theory and Practice*. McGraw-Hill, 1994.
472. M. Raber, T. Remel, E. Hoffman, D. Maurer, F. Muller, H.-G. Oberhauser, and R. Wilhelm. Compiled Graph Reduction on a Processor Network. Technical report, Universitdt des Saarlandes, Saarbrucken, DE, 1987.
473. M. Raber, T. Remel, D. Maurer, F. Muller, H.-G. Oberhauser, and R. Wilhelm. A Concept for a Parallel G-machine. Technical Report SFB 124-C1, Universitdt des Saarlandes, Saarbrucken, DE, 1987.
474. F. A. Rabhi. A Parallel Programming Methodology Based on Paradigms. In P. Nixon, editor, *Transputer and Occam Developments*, pages 239–252. IOS Press, 1995.
475. T. Rauber and G. Runger. Parallel Numerical Algorithms with Data Distribution Types. Technical Report 95-04, University of Saarbrucken, Sonderforschungsbereich 124, 1995.
476. T. Rauber and G. Runger. Deriving Structured Parallel Implementations for Numerical Methods. *Microprocessing and Microprogramming*, 41:589–608, 1996.

477. T. Rauber and G. Rünger. The Compiler TwoL for the Design of Parallel Implementations. In *PACT'96 — International Conference on Parallel Architecture and Compilations Techniques*, pages 292–301, Boston, USA, October 1996. IEEE Computer Society Press.
478. C. Reade. *Elements of functional programming*. Addison-Wesley, 1989.
479. J. Reed, K. Parrott, and T. Lanfear. Portability, Predictability and Performance for Parallel Computing: BSP in Practice. *Concurrency: Practice and Experience*, 8(10):799–812, December 1996.
480. M. Reeve and S. Wright. The Experimental ALICE Machine. In T. Fountain and M. Shute, editors, *Multiprocessor Computer Architecture*, pages 39–20. North Holland, Amsterdam, 1990.
481. J. H. Reif and L. G. Valiant. A Logarithmic Time Sort for Linear Size Networks. *Journal of the ACM*, 34(1):60–76, 1987.
482. J. Reppy. CML: A Higher-Order Concurrent Language. In *PLDI'91 — Programming Languages Design and Implementation*, pages 293–305. ACM Press, June 1991.
483. J. Reppy. Concurrent ML: Design, Application and Semantics. In *Programming, Concurrency, Simulation and Automated Reasoning*, number 693 in Lecture Notes in Computer Science. Springer-Verlag, 1992.
484. J. Reppy. Higher-Order Concurrency. Tr 92–1852, Department of Computer Science, Cornell University, Ithaca, NY 14853, 1992.
485. J. Reppy. *Concurrent Programming in ML*. Cambridge University Press, 1999.
486. J. Reynolds. The Essence of ALGOL. In J. de Bakker and J. van Vliet, editors, *Algorithmic Languages*, pages 345–372. IFIP, North-Holland Publishing Company, 1981.
487. C. P. Riley, R. C. F. McLatchie, R. Janssen, A. Longo, T. Gutiérrez, S. Casado, J. Simkin, P. Brochet, G. Molinari, P. Alotto, J.-F. Lemoine, and G. Drago. Optimisation of Electromagnetic Design Using HPCN. In *HPCN'98: High-Performance Computing and Networking*, volume 1401 of *LNCS*. Springer-Verlag, April 1998.
488. H. Rischel and H. Sun. Design and Prototyping of Real-time Systems Using CSP and CML. In *Proceedings of 9th Euromicro Workshop on Real-Time Systems*, pages 121–127. IEEE Computer Society Press, 1998.
489. P. Roe. *Parallel Programming Using Functional Languages*. PhD thesis, Glasgow University, UK, 1991.
490. N. Rojemo. A Concurrent Generational Garbage Collector for a Parallel Graph Reducer. In Y. Bekkers and J. Cohen, editors, *Memory Management; International Workshop IWMM 92. Proceedings*, pages 440–453, Berlin, Germany, 1992. Springer-Verlag.
491. W. A. Roscoe. *Theory and Practice of Concurrency*. Prentice-Hall, 1998.
492. C. Ruggiero and J. Sargeant. Control of Parallelism in the Manchester Dataflow Machine. In G. Kahn, editor, *Functional Programming Languages and Computer Architecture*, volume 274 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, Berlin, DE, 1987.
493. C. Runciman and D. Wakeling. Profiling Parallel Functional Computations (Without Parallel Machines). In *Functional Programming, Glasgow '93*, pages 236–251, 1993.
494. C. Runciman and D. Wakeling. *A Quasi-Parallel Evaluator*, chapter 10, pages 161–176. UCL Press, 1995.
495. C. Runciman and D. Wakeling, editors. *Applications of Functional Programming*. UCL Press, 1995.
496. G. Sabot. *The Parallation Model : Architecture Independent SIMD Programming*. MIT Press, 1988.

497. A. H. Sameh. On Jacobi-Like Algorithms for a Parallel Computer. *Mathematical Comput.*, 25:579–590, 1971.
498. D. Sands. *Calculi for Time Analysis of Functional Programs*. PhD thesis, Imperial College, London, September 1990.
499. P. Sansom and S. Peyton Jones. Formally Based Profiling for Higher-Order Functional Languages. *ACM Transactions on programming languages and systems*, 19(1), 1997.
500. V. Sarkar and J. Hennessy. Partitioning Parallel Programs for Macro-Dataflow. In *ACM Symposium on Lisp and Functional Programming*, pages 202–211, 1986.
501. M. Scheevel. NORMA: A Graph Reduction Processor. In *ACM Conference on LISP and Functional Programming, Cambridge, MA*, pages 109–139, 1986.
502. C. Schmittgen, H. Bloedorn, and W. Kluge. π -RED* – A Graph Reducer for a Full-Fledged λ -Calculus. *New Generation Computing, OHMSHA Ltd. and Springer*, 10(2):173–195, 1992.
503. S.-B. Scholz. On Programming Scientific Applications in SAC – a Functional Programming Language Extended by a Subsystem for High-Level Array Operations. In *Proceedings of 8th International Workshop on Implementation of Functional Languages, Bonn*, pages 85 – 104. Lecture Notes in Computer Science, No. 1268, 1996.
504. S.-B. Scholz. *Single Assignment C – Entwurf und Implementierung einer funktionalen C-Variante mit spezieller Unterstützung shape-invarianter Array-Operationen (in German)*. PhD thesis, Institut für Informatik und praktische Mathematik, Universität Kiel, October 1996.
505. S.-B. Scholz. With-Loop-Folding in SAC — Condensing Consecutive Array Operations. In *IFL'97 — International Workshop on the Implementation of Functional Languages*, volume 1467 of *Lecture Notes in Computer Science*, pages 225–242, University of St. Andrews, Scotland, September 10–12, 1997. Springer-Verlag.
506. S.-B. Scholz. A Case Study: Effects of WITH-Loop-Folding on the NAS Benchmark MG in SAC. In *IFL'98 — International Workshop on the Implementation of Functional Languages*, volume 1595 of *Lecture Notes in Computer Science*, pages 220–231, September 9–11, University College London, UK, 1998. Springer-Verlag.
507. W. Schreiner. Application of a Para-Functional Language to Problems in Computer Algebra. In *HPFC'95 — Conference on High Performance Functional Computing*, pages 10–24, Denver, CO, April 10–12, 1995.
508. R. Schroeder. Distributed Breadth-first Searches as Speculative Evaluations. In *Proceedings of 9th International Workshop on Implementation of Functional Languages, University of St. Andrews*, pages 243 – 259, 1997.
509. J. Sérot. Embodying Parallel Functional Skeletons: An Experimental Implementation on Top of MPI. In C. Lengauer, M. Griebel, and S. Gorlatch, editors, *EuroPar'97 Parallel Processing*, number 1300 in LNCS, pages 629–633, Passau, Germany, August 1997. Springer.
510. J. Sérot, D. Ginhac, and J. Dértin. SKiPPER: A Skeleton-Based Parallel Programming Environment for Real-Time Image Processing Applications. In *PaCT'99 — International Parallel Computing Technologies Conference, St-Petersburg*, September 6–10, 1999.
511. P. R. Serrarens. Implementing the Conjugate Gradient Algorithm in a Functional Language. In Kluge [289], pages 125–140.
512. P. R. Serrarens. Distributed Arrays in the Functional Language Concurrent Clean. In *Proceedings of the 3rd international Euro-Par conference*, volume 1300 of LNCS, pages 1201–1208, Passau, Germany, August 1997.

513. P. R. Serrarens. Using Multicasting for Optimising Data-Parallelism. In C. Clack, K. Hammond, and T. Davie, editors, *Proceedings of 9th International Workshop on the Implementation of Functional Languages*, volume 1467 of *Lecture Notes in Computer Science*, pages 271–285. Springer Verlag, 1997.
514. P. R. Serrarens. Explicit message passing for Concurrent Clean. In *Implementation of Functional Languages, 10th International Workshop*, pages 289–308, September 1998.
515. P. R. Serrarens and M. J. Plasmeijer. Explicit Message Passing for Concurrent Clean. In Hammond et al. [290], pages 232–248.
516. E. Shapiro. The Family of Concurrent Logic Programming Languages. *ACM Computing Surveys*, 21(3), 1989.
517. A. Shaw. Impala – IMplicitly PARallel LAnguage Application Suite. Technical report, MIT Computation Structures Group, June 11th 1998.
518. A. Shaw, Arvind, K.-C. Cho, C. Hill, R. Johnson, and J. Marshall. A Comparison of Implicitly Parallel Multithreaded and Data Parallel Implementations of an Ocean Model. *Journal of Parallel and Distributed Computing*, 48(1):1–51, January 1998.
519. H. Simon. *The Sciences of the Artificial*. MIT Press, 1969.
520. A. Simpson, J. M. D. Hill, and S. R. Donaldson. BSP in CSP: Easy as ABC. In *Workshop on Formal Methods for Parallel Computing, held in conjunction with IPPS/SPDP (IPPS'99)*, San Juan, Puerto Rico, April 1999. Springer-Verlag.
521. S. K. Skedzielewski. Sisal. In Szymanski, editor, *Parallel Functional Languages and Compilers*. ACM Press, 1991.
522. D. B. Skillicorn. Models for Practical Parallel Computation. *International Journal of Parallel Programming*, 20(2):133–158, 1991.
523. D. B. Skillicorn. *Foundations of Parallel Programming*. Number 6 in Cambridge Series in Parallel Computation. Cambridge University Press, 1994.
524. D. B. Skillicorn. miniBSP: A BSP Language and Transformation System. <http://www.qucis.queensu.ca/home/skill/mini.ps>, November 1996.
525. D. B. Skillicorn. Parallel Implementation of Tree Skeletons. *Journal of Parallel and Distributed Computing*, 39:115–125, 1996.
526. D. B. Skillicorn. Towards a Framework for Cost-Based Transformation. *Journal of Systems and Architectures (the Euromicro Journal)*, 42:331–340, 1996.
527. D. B. Skillicorn and W. Cai. A Cost Calculus for Parallel Functional Programming. *Journal of Parallel and Distributed Computing*, 28(1):65–83, July 1995.
528. D. B. Skillicorn, M. Danelutton, S. Pelagatti, and A. Zavanella. Optimising Data-Parallel Programs Using the BSP Cost Model. In *EuroPar'98*, volume 1470 of *LNCS*, pages 698–708. Springer-Verlag, September 1998.
529. D. B. Skillicorn, J. M. D. Hill, and W. F. McColl. Questions and Answers about BSP. *Scientific Programming*, 6(3):249–274, Fall 1997.
530. D. B. Skillicorn and D. Talia. Models and Languages for Parallel Computation. *ACM Computing Surveys*, 30(2):123–169, 1998.
531. G. S. Smith. Principal Type Schemes for Functional Programs with Overloading and Subtyping. *Science of Computer Programming*, 23:197–226, 1994.
532. S. Smith. The LMI Lambda Technical Summary. Technical report, LMI Inc., Los Angeles, CA, 1984.
533. M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra. *MPI: The Complete Reference*. MIT Press, Cambridge, MA, 1995.
534. A. Sodan and H. Bock. Extracting Characteristics from Functional Programs for Mapping to Massively Parallel Machines. In *HPFC'95 — High Performance Functional Computing*, pages 134–148, Denver, CO, April 10–12, 1995.

535. J. Sparud and C. Runciman. Complete and Partial Redex Trails of Functional Computations. In *Proceedings of 9th International workshop on the Implementation of Functional Languages (IFL '97)*, pages 160–177. Springer-Verlag LNCS 1467, 1998.
536. M. Spivey. *The Z Notation: A Reference Manual*. Prentice-Hall, 1989.
537. G. L. Steele Jr. and W. D. Hillis. Connection Machine Lisp : Fine-Grained Parallel Symbolic Processing. In *ACM Conference on Lisp and functional programming*, pages 279–297, 1986.
538. W. Stoye. The SKIM II Microprogrammer's Guide. Technical note, Cambridge University Computer Laboratory, 1983.
539. M. Südholt. *The Transformational Derivation of Parallel Programs Using Data-Distribution Algebras and Skeletons*. PhD thesis, Fachbereich 13 Informatik, Technische Universität Berlin, August 1997.
540. M. Südholt, C. Piepenbrock, K. Obermayer, and P. Pepper. Solving Large Systems of Differential Equations Using Covers and Skeletons. In *50th IFIP WG 2.1 Working Conference on Algorithmic Languages and Calculi*, Strasbourg, France, Feb. 1997. Chapman & Hall.
541. K. R. Sujithan. *Bulk Synchronous Parallelism in Object-Relational Database Systems*. D.Phil. thesis, Oxford University Computing Laboratory, Trinity Term 1997.
542. K. R. Sujithan and J. M. D. Hill. Collection Types for Database Programming in the BSP Model. In *5th EuroMicro Workshop on Parallel and Distributed Processing (PDP'97)*. IEEE Computer Society Press, Jan. 1997.
543. S. Sur and A. P. W. Böhm. Analysis of Non-Strict Functional Implementations of the Dongarra-Sorensen Eigensolver. In *Proceedings of ICS94*, pages 412–418, Manchester, UK, 1994.
544. S. Sur and A. P. W. Böhm. Functional, I-Structure, and M-Structure Implementations of NAS Benchmark FT. In *PACT'94 — International Conference on Parallel Architecture and Compilation Techniques*, pages 47–56, Montreal, Canada, Aug. 1994.
545. J.-P. Talpin. The Calumet Experiment in Facile — A Model for Group Communication and Interaction Control in Cooperative Applications. Technical report ECRC-94-26, ECRC Munich, 1994.
546. J.-P. Talpin and P. Jouvelot. Compiling FX on the CM-2. In *Workshop on static analysis*, number 724 in LNCS, 1993.
547. J.-P. Talpin and P. Jouvelot. The Type and Effect Discipline. *Information and Computation*, 111(2):245–296, 1994.
548. F. Taylor. *Parallel Functional Programming by Partitioning*. PhD thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London, January 1997.
549. A. Telford and D. Turner. Ensuring Streams Flow. In M. Johnson, editor, *Algebraic Methodology and Software Technology 1997*. Springer-Verlag, 1997.
550. S. Thompson. *Type Theory and Functional Programming*. Addison Wesley, 1991.
551. S. Thompson. A Logic for Miranda, Revisited. *Formal Aspects of Computing*, 7, 1995.
552. S. Thompson. *Haskell: The Craft of Functional Programming*. Addison-Wesley, 1996.
553. B. Thomsen, L. Leth, and T.-M. Kuo. FACILE—from Toy to Tool. In F. Nielson, editor, *ML with Concurrency: Design, Analysis, Implementation and Application*, Monographs in Computer Science, pages 97–144. Springer, 1997.
554. B. Thomsen, L. Leth, S. Prasad, T.-M. Kuo, A. Kramer, F. Knabe, and A. Giacalone. Facile Antigua Release Programming Guide. Technical Re-

- port ECRC-93-20, ECRC Munich, 1993.
555. A. Tiskin. The Bulk-Synchronous Parallel Random Access Machine. *Theoretical Computer Science*, 196(1):109–130, 1998.
556. H. To. Optimising the Parallel Behaviour of Combinations of Program Components. PhD Thesis, Department of Computer Science, Imperial College, 1995.
557. K. Traub, D. Culler, and K. Schauser. Global Analysis for Partitioning Non-Strict Programs into Sequential Threads. In *LFP'92 — Conference on LISP and Functional Programming*, San Francisco, CA, June 1992.
558. P. Trinder, E. Barry Jr., M. Davis, K. Hammond, S. Junaidu, U. Klusik, H.-W. Loidl, and S. L. Peyton Jones. GPH: An Architecture-Independent Functional Language. *IEEE Transactions on Software Engineering*, 1999. Submitted for publication.
559. P. Trinder, K. Hammond, H.-W. Loidl, and S. L. Peyton Jones. Algorithm + Strategy = Parallelism. *Journal of Functional Programming*, 8(1):23–60, January 1998.
560. P. Trinder, K. Hammond, H.-W. Loidl, S. L. Peyton Jones, and J. Wu. A Case Study of Data-intensive Programs in Parallel Haskell. In *Glasgow Workshop on Functional Programming*, Workshops in Computing, Ullapool, Scotland, July 8–10, 1996. Springer-Verlag.
561. P. Trinder, K. Hammond, S. L. Peyton Jones, and A. Partridge. GUM: A Portable Parallel Implementation of Haskell. In *Proceedings SIGPLAN Symposium on Programming Language Design and Implementation (PLDI'96)*, 1996.
562. D. A. Turner. A New Implementation Technique for Applicative Languages. *Software - Practice and Experience*, 9:31–49, 1979.
563. D. A. Turner. Miranda: A Non-Strict Functional Language with Polymorphic Types. In J. Jouannaud, editor, *Functional Programming Languages and Computer Architecture*, volume 201 of *LNCS*, pages 1–16, Nancy, France, 1985. Springer-Verlag.
564. D. A. Turner. Elementary Strong Functional Programming. In P. Hartel and R. Plasmeijer, editors, *Functional Programming Languages in Education (FPLE)*, *LNCS 1022*. Springer-Verlag, Heidelberg, 1995.
565. University of Washington. ZPL. <http://www.cs.washington.edu/research/zpl/index.html>, 1997.
566. L. G. Valiant. A Scheme for Fast Parallel Communication. *SIAM Journal on Computing*, 11(2):350–361, May 1982.
567. L. G. Valiant. A Bridging Model for Parallel Computation. *Communications of the ACM*, 33(8):103–111, August 1990.
568. L. G. Valiant. General Purpose Parallel Architectures. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. A*. Elsevier Science Publishers and MIT Press, 1990.
569. L. G. Valiant. Bulk-synchronous Parallel Computer. U.S. Patent No. 5083265, 1992.
570. M. van Eekelen and M. Plasmeijer. Specification of Reduction Strategies in Term Rewriting Systems. In J. Fasel and R. Keller, editors, *Graph Reduction: Proceedings of a Workshop at Santa F, New Mexico*, pages 215–239, New York, NY, 1987. Springer-Verlag.
571. J. van Groningen. Implementing the ABC-Machine on M680x0 Based Architectures. Master's thesis, University of Nijmegen, 1990.
572. S. R. Vegdahl. A Survey of Proposed Architectures for the Execution of Functional Languages. *IEEE-Transactions on Computers*, C-33(12):1050 – 1071, 1984.

573. J. Vrancken. Reflections on Parallel Functional Languages. In *Proceedings 2nd International Workshop on Implementation of Functional Languages on Parallel Architectures*, 1990.
574. W. Vree. Experiments with Coarse-Grain Parallel Graph Reduction. *Future Generation Computer Systems*, 4:299–306, 1988.
575. P. Wadler. Comprehending Monads. *Mathematical Structures in Computer Science*, 1993. Special issue of selected papers from 6'th Conference on Lisp and Functional Programming, 1992.
576. P. Wadler. Why No One Uses Functional Languages. *ACM SIGPLAN*, 33(8), August 1998.
577. P. Wadler and S. Blott. Type Class Overloading. In *Proceedings of POPL '89 – ACM Symposium on Principles of Programming Languages*, pages ??–??, 1989.
578. C. P. Wadsworth. *Semantics and Pragmatics of the Lambda-Calculus*. PhD thesis, University of Oxford, 1971.
579. C. P. Wadsworth. The Relation Between Computational and Denotational Properties for Scott's D_∞ Models of the Lambda-Calculus. *SIAM Journal of Computing*, 5, 1976.
580. I. Watson, J. Sargeant, P. Watson, and V. Woods. Flagship Computational Models and Machine Architecture. *ICL Technical Journal*, pages 555–574, 1987.
581. P. Watson and I. Watson. Evaluating Functional Programs on the FLAGSHIP Machine. In G. Kahn, editor, *Functional Programming Languages and Computer Architecture*, pages 80–97. Springer-Verlag, Berlin, DE, 1987.
582. C. Wedler and C. Lengauer. Notes on the Classification of Parallel Implementations of Linearly Recursive Programs. In *Proceedings of GI/ITG FG PARS'95*, PARS Mitteilungen Nr.14, pages 140–147. Gesellschaft für Informatik e.V., 1995.
583. P. Wegner. *Programming Languages, Information Structures and Machine Organization*. McGraw-Hill, 1971.
584. D. J. Whalen, D. E. Hollowell, and J. S. Hendricks. MCNP: Photon Benchmark Problems. Technical Report Report LA-12196, Los Alamos National Laboratory, 1991.
585. R. P. Whittington. *Database Systems Engineering*. Clarendon Press, Oxford, 1998.
586. G. Winskel. *The Formal Semantics of Programming Languages*. MIT Press, 1993.
587. Q. Wu, A. J. Field, and P. H. J. Kelly. M-Tree: A Parallel Abstract Data Type for Block-Irregular Adaptive Applications. In C. Lengauer, M. Griehl, and S. Gorlatch, editors, *EuroPar'97 – European Conference on Parallel Processing*, volume 1300 of *Lecture Notes in Computer Science*, pages 638–49, Passau, Germany, August 1997. Springer-Verlag.
588. J. Yang. *Co-ordination Based Structured Parallel Programming*. PhD thesis, Department of Computing, Imperial College of Science, Technology and Medicine, University of London, September 1998.
589. T. Yuba, T. Shimada, K. Hiraki, and H. Kashiwagi. SIGMA-1: A Dataflow Computer for Scientific Computations. *Computer Physics Communications*, pages 141–148, 1985.
590. Z.G. Mou, and M. Hudak. An Algebraic Model for Divide-and-Conquer Algorithms and its Parallelism. *Journal of Supercomputing*, 2(3), 1988.
591. X. Zhang, M. Webster, J. Sharp, and P. Grant. Parallel Computational Fluid Dynamics. In C. Runciman and D. Wakeling, editors, *Applications of functional programming*. UCL Press, 1995.

592. R. Zimmer. *Zur Pragmatik eines Operationalisierten λ -Kalkuels als Basis fuer Interaktive Reduktionssysteme*, volume 192 of *GMD-Bericht*. Oldenbourg, 1991.