Vita

Andrew W. Appel

Department of Computer Science, Princeton University, 35 Olden Street, Princeton NJ 08544-2087

appel@princeton.edu, +1-609-258-4627, fax: +1-609-258-1771,
http://www.cs.princeton.edu/~appel

Research Interests

Programming languages, computer security, compilers, type theory, semantics, software engineering, automated theorem proving.

Education

A.B. *summa cum laude* (physics) Princeton University, 1981
Ph.D. (computer science) Carnegie-Mellon University, 1985

Professional Appointments

Princeton University, Princeton, NJ. Asst. Prof. of Computer Science, 1986-92; Assoc. Prof., 1992-95; **Professor**, since 1995; Associate Chair, 1997-98, 2000-present.

Bell Laboratories, Murray Hill, NJ. Member of Technical Staff, Summer 1984. Consultant, 1983-2001.

Carnegie-Mellon University, Pittsburgh, PA. Research and teaching assistant, 1982-85.

College of Medicine, University of Illinois, Urbana, IL. Computer programmer, summers 1976-80.

Awards and Honors

Kusaka Memorial Prize in Physics, Princeton University, 1981.

National Science Foundation Graduate Student Fellowship, 1981-1984.

ACM Fellow (Association for Computing Machinery), 1998.

The Other Prize, Programming Contest of the ACM International Conference on Functional Programming, 1998.

Personal Information

Born June 30, 1960. U.S. citizen. Married, two children.

Professional Activities

1. Program Committee, *ACM SIGPLAN '89 Conf. on Prog. Lang. Design and Implementation*, 1989.
2. Program Committee, *Seventeenth ACM Symp. on Principles of Programming Languages*, 1990.
3. Associate Editor, *ACM Transactions on Programming Languages and Systems*, 1990-1992.
4. Associate Editor, *ACM Letters on Programming Languages and Systems*, 1991-1992.
5. Program Chair, *Nineteenth ACM Symp. on Principles of Programming Languages*, 1992.
6. Co-editor, *Journal of Functional Programming* special issue on ML, 1992.
7. Program Committee, *Sixth ACM Conf. on Functional Prog. Lang. and Computer Architecture*, 1993.
8. Editor in Chief, *ACM Transactions on Programming Languages and Systems*, 1993-97.
9. Program Committee, *International Conference on Functional Programming*, 1997.
10. General Chair, *26th ACM Symp. on Principles of Programming Languages*, 1999.
11. Program Committee, *IEEE Symposium on Security and Privacy*, 2002.

Research Grants

1. *Implementation of an efficient reducer for lambda expressions,* National Science Foundation DCR-8603453, $115,799, 1986-88.
2. Digital Equipment Corporation Faculty Incentive Grant, $180,000, 1986-89.
3. *Unifying compile-time and run-time evaluation,* National Science Foundation CCR-8806121, $123,510, 1988-90.
4. *Standard ML of New Jersey software capitalization,* National Science Foundation CCR-8914570, $119,545, 1990-91.
5. *Using immutable types for debugging and parallelism,* National Science Foundation CCR-9002786, $174,618, 1990-92.
6. *Optimization of space usage,* National Science Foundation CCR-9200790, $348,119, 1992-96.

7. *Framework, Algorithms, and Applications for Cross-module Inlining,* National Science Foundation CCR-9625413, $180,331, 1996-98.

8. *Development of a HIL/LIL Framework for a National Compiler Infrastructure*, Defense Advanced Research Projects Agency and National Science Foundation (as subcontractor to Univ. of Virginia), $1,397,293, 1996-99.

9. *Tools, Interfaces, and Access Control for Secure Programming,* National Science Foundation CCR-9870316, $322,000, 1998-2001 (co-PI).

10. *Scaling Proof-Carrying Code to Production Compilers and Security Policies*, Defense Advanced Research Projects Agency, $2,224,772, 1999-2002.

11. *Applying Compiler Techniques to Proof-Carrying Code*, National Science Foundation CCR-9974553, $220,000, 1999-2002.

12. IBM University Partnership Program, $40,000, 1999-2000.

Publications

Books, chapters in books

1. ``Garbage Collection,'' in *Topics in Advanced Language Implementation,* Peter Lee, ed. MIT Press, 1991.

2. *Compiling with Continuations,* Cambridge University Press, 1992.

3. *Modern Compiler Implementation in ML,*
Modern Compiler Implementation in Java,
Modern Compiler Implementation in C,
Cambridge University Press, 1998.

Journal papers

4. An Efficient Program for Many-Body Simulations. *SIAM Journal on Scientific and Statistical Computing* 6(1):85-103, 1985.

5. Generalizations of the Sethi-Ullman algorithm for register allocation. with Kenneth J. Supowit. *Software -- Practice & Experience* 17(6):417-421, 1987.

6. Garbage collection can be faster than stack allocation. *Information Processing Letters,* 25(4):275-279, 17 June 1987.

7. The World's Fastest Scrabble Program, with Guy J. Jacobson, *Comm. ACM* 31(5):572-578,585, May 1988.

8. Simulating digital circuits with one bit per wire, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 7(9):987-993, September 1988.

9. Simple Generational Garbage Collection and Fast Allocation. *Software -- Practice &*

Experience 19(2):171-183, February 1989.

10. Allocation without Locking. *Software -- Practice & Experience* 19(7):703-705, July 1989.

11. Runtime Tags Aren't Necessary. *Lisp and Symbolic Computation* 2, 153-162 (1989).

12. Vectorized Garbage Collection, with Aage Bendiksen. *The Journal of Supercomputing* 3, 151-160 (1989).

13. A Runtime System. *Lisp and Symbolic Computation* 3, 343-380, 1990.

14. Real-time concurrent garbage collection system and method, with John R. Ellis and Kai Li. U.S. Patent 5,088,036, 1992.

15. Callee-save registers in Continuation-Passing Style, with Zhong Shao. *Lisp and Symbolic Computation* 5, 189-219, 1992.

16. A Critique of Standard ML. *Journal of Functional Programming* 3 (4) 391-430, 1993.

17. A Debugger for Standard ML, with Andrew Tolmach. *Journal of Functional Programming* 5 (2) 155-200, 1995.

18. Axiomatic Bootstrapping: A Guide for Compiler Hackers. *ACM Trans. on Prog. Lang. and Systems* 16 (6) 1699-1718, November 1994.

19. Loop Headers in Lambda-calculus or CPS. *Lisp and Symbolic Computation* 7, 337-343, 1994.

20. Empirical and Analytic Study of Stack versus Heap Cost for Languages with Closures, with Zhong Shao. *Journal of Functional Programming* 6 (1) 47-74, 1996.

21. How to Edit a Journal by E-mail, *Journal of Scholarly Publishing* 27 (2) 82-99, January 1996.

22. Iterated Register Coalescing, with Lal George. *ACM Trans. on Prog. Lang. and Systems* 18(3) 300-324, May 1996.

23. Security and document compatibility for electronic refereeing, *CBE Views* 20(1) 9-10, 1997, published by the Council of Biology Editors.

24. Shrinking lambda-calculus in linear time, with Trevor Jim. *Journal of Functional Programming* 7 (5) 515-540, 1997.

25. Hierarchical Modularity, with Matthias Blume. *ACM Transactions on Programming Languages and Systems*, 21 (4) 812-846, July 1999.

26. Efficient and Safe-for-Space Closure Conversion, with Zhong Shao, *ACM Trans. on Prog. Lang. and Systems* 22(1) 129-161, January 2000.

27. Technological Access Control Interferes with Noninfringing Scholarship, with Edward

W. Felten. *Communications of the ACM,* 43 (9) 21-23, September 2000.

28. An Indexed Model of Recursive Types for Foundational Proof-Carrying Code, with David McAllester. *ACM Transactions on Programming Languages and Systems*, to appear.

29. SAFKASI: A Security Mechanism for Language-Based Systems with Dan Wallach and Edward W. Felten. ACM Transactions on Software Engineering and Methodology, 9 (4) 341-378, October 2000.

30. Polymorphic Lemmas and Definitions in Lambda Prolog and Twelf, with Amy P. Felty. Accepted for publication, *Theory and Practice of Logic Programming.*

Conference papers

31. A Microprocessor-Based CAI System with Graphic Capabilities, with F. J. Mabry and A. H. Levy. *Proc. 1978 conference, Assoc. for Development of Computer-based Instruction Systems.*

32. Rogomatic: A Belligerent Expert System, with M. L. Mauldin, G. J. Jacobson, and L. G. C. Hamey. *Proc. Fifth Nat. Conf. Canadian Soc. for Computational Studies of Intelligence,* May, 1984.

33. Semantics-Directed Code Generation, *Proc. Twelfth ACM Symposium on Principles of Programming Languages,* January 1985.

34. A Standard ML compiler, with D. B. MacQueen, *Proc. Third Int'l Conf. on Functional Programming & Computer Architecture (LNCS 274, Springer-Verlag),* Portland, Oregon, September 1987.

35. Real-time concurrent collection on stock multiprocessors, with John Ellis & Kai Li. *Proc. ACM SIGPLAN '88 Conf. on Prog. Lang. Design & Implementation,* pp. 11-20, June 1988.

36. Continuation-passing, closure-passing style, with Trevor Jim. *Proc. Sixteenth ACM Symposium on Principles of Programming Languages,* pp. 293-302, January 1989.

37. An advisor for flexible working sets, with Rafael Alonso. *1990 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems,* pp. 153-162, May 1990.

38. Debugging Standard ML without reverse engineering, with Andrew P. Tolmach. *Proc. 1990 ACM Conf. on Lisp and Functional Programming,* pp. 1-12, June 1990.

39. Virtual memory primitives for user programs, with Kai Li. *Proc. Fourth Int'l Conf. on Architectural Support for Prog. Languages and Operating Systems,* (SIGPLAN Notices 26(4)) pp. 96 107, April 1991.

40. Standard ML of New Jersey, with David B. MacQueen. *Third Int'l Symp. on Prog. Lang. Implementation and Logic Programming, Springer-Verlag LNCS 528,* pp. 1-13, August 1991.

41. Debuggable concurrency extensions for Standard ML, with Andrew P. Tolmach. *Proc. ACM/ONR Workshop on Parallel and Distributed Debugging,* May 1991 (SIGPLAN Notices, Dec. 1991), pp. 115-127.

42. Smartest Recompilation, with Zhong Shao. *Proc. Twenthieth ACM Symp. on Principles of Programming Languages,* January 1993.

43. Unrolling Lists, with Zhong Shao and John H. Reppy. *Proc. 1994 ACM Conf. on Lisp and Functional Programming,* pp. 185-195, June 1994.

44. Space-Efficient Closure Representations, with Zhong Shao. *Proc. 1994 ACM Conf. on Lisp and Functional Programming,* pp. 150-161, June 1994.

45. Separate Compilation for Standard ML, with David B. MacQueen. *Proc. 1994 ACM Conf. on Programming Language Design and Implementation* (SIGPLAN Notices v. 29 #6), pp. 13-23, June 1994.

46. A Type-Based Compiler for Standard ML, with Zhong Shao. *Proc. 1995 ACM Conf. on Programming Language Design and Implementation* (SIGPLAN Notices v. 30 #6), pp. 116-129, June 1995.

47. Cache Performance of Fast-Allocating Programs, with Marcelo J. R. Goncalves. *Proc. Seventh Int'l Conf. on Functional Programming and Computer Architecture,* pp. 293-305, ACM Press, June 1995.

48. Iterated Register Coalescing, with Lal George. *23rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* pp. 208-218, January 1996.

49. Lambda-Splitting: A Higher-Order Approach to Cross-Module Optimizations, with Matthias Blume. *Proc. ACM SIGPLAN International Conference on Functional Programming (ICFP '97)*, pp. 112-124, June 1997.

50. The Zephyr Abstract Syntax Description Language, with Daniel C. Wang, Jeff L. Korn, and Christopher S. Serra. *Conference on Domain-Specific Languages*, USENIX Association, October 1997.

51. Traversal-based Visualization of Data Structures, with Jeffrey L. Korn, *IEEE Symposium on Information Visualization (InfoVis '98)*, pp. 11-18, October 1998.

52. Lightweight Lemmas in Lambda Prolog, with Amy Felty, *16th International Conference on Logic Programming*, pp. 411-425, MIT Press, November 1999.

53. Proof-Carrying Authentication, with Edward Felten, *6th ACM Conference on Computer and Communications Security*, November 1999.

54. A Semantic Model of Types and Machine Instructions for Proof Carrying Code, with Amy P. Felty. *27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '00)*, pp. 243-253, January 2000.

55. Machine Instruction Syntax and Semantics in Higher Order Logic, with Neophytos G.

Michael. *17th International Conference on Automated Deduction (CADE-17)*, Springer-Verlag (Lecture Notes in Artificial Intelligence), pp. 7-24, June 2000.

56. Efficient Substitution in Hoare Logic Expressions, with Kedar Swadi and Roberto Virga. *4th International Workshop on Higher-Order Operational Techniques in Semantics (HOOTS 2000)*, September 2000.

57. Type-Preserving Garbage Collectors, with Daniel C. Wang. *POPL 2001: The 28th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 166-178, January 2001.

58. Fair use, public domain, or piracy ... should the digital exchange of copyrighted works be permitted or prevented? (panel symposium) with Y. Benkler, M. Carlinsky, et al., *Fordham Intellectual Property, Media & Entertainment Law Journal*, 11(2) 257-408, Winter 2001.

59. Optimal Spilling for CISC Machines with Few Registers, with Lal George. *ACM SIGPLAN 2001 Conference on Programming Language Design and Implementation*, pp. 243-253, June 2001.

60. Foundational Proof-Carrying Code. *16th Annual IEEE Symposium on Logic in Computer Science* (LICS '01), pp. 247-258, June 2001.

61. A Stratified Semantics of General References Embeddable in Higher-Order Logic, with Amal Ahmed and Roberto Virga. *17th Annual IEEE Symposium on LOGIC IN COMPUTER SCIENCE (LICS 2002),* to appear June 2002.

Review and Tutorial Articles

62. Book Review of *Garbage Collection: Algorithms for Automatic Dynamic Memory Management* by Richard Jones and Rafael Lins. *Journal of Functional Programming* 7(2), pp. 227-229, March 1997.

63. SSA is Functional Programming. *ACM SIGPLAN Notices* v. 33, no. 4, pp. 17-20, April 1998.

64. Protection against untrusted code. *IBM Developer Works*, September 1999.

Unrefereed papers

65. An Asymptotically Fast Algorithm for N-Body Simulations. Senior Thesis, Princeton University, 1981.

66. Compile-time Evaluation and Code Generation in Semantics-Directed Compilers. Ph.D. Thesis, Carnegie-Mellon University, July 1985.

67. Concise specifications of locally optimal code generators, Princeton Univ. Dept. of Computer Science CS-TR-080-87, 1987.

68. Re-opening closures, Princeton Univ. Dept. of Computer Science CS-TR-079-87,

February 1987.

69. Optimizing closure environment representations, with Trevor Jim. Princeton Univ. Dept. of Computer Science CS-TR-168-88, July 1988.

70. Unifying Exceptions with Constructors in Standard ML, with David MacQueen, Robin Milner, and Mads Tofte. Univ. of Edinburgh Dept. of Comp. Sci. CSR-266-88, May 1988.

71. Profiling in the presence of optimization and garbage collection, with Bruce Duba and David MacQueen. CS-TR-197-88, November 1988.

72. Emulating Write-Allocate on a No-Write-Allocate Cache, CS-TR-459-94, Princeton University, June 20, 1994.

73. Is POPL Mathematics or Science? *ACM SIGPLAN Notices* 27 (4), pp. 87-89, April 1992.

74. Intensional Equality ;=) for Continuations, *ACM SIGPLAN Notices* 31 (2), pp. 55-57, February 1996.

PhD Students

1. Andrew P. Tolmach, Ph.D. (1992) *Debugging Standard ML.* Associate Professor, Portland State University.

2. Zhong Shao, Ph.D. (1994) *Compiling Standard ML for Efficient Execution on Modern Machines.* Associate Professor, Yale University.

3. Marcelo J. R. Goncalves, Ph.D. (1995) *Cache Performance of Programs with Intensive Heap Allocation and Generational Garbage Collection*. Oracle.com, Portland, Oregon.

4. Matthias Blume, Ph.D. (1997) *Hierarchical Modularity and Intermodule Optimization*. Member of Technical Staff, Lucent Bell Laboratories.

5. Drew Dean, Ph.D. (1999) *Formal Aspects of Mobile Code Security*. Computer Scientist, SRI International.

6. Jeffrey L. Korn, Ph.D. (1999) *Abstraction and Visualization in Graphical Debuggers.* Member of Technical Staff, AT&T Research.

7. Daniel C. Wang, Ph.D. (2002) *Managing Memory with Types.* Member of Technical Staff, Agere Systems.