DAVID GRIES

March 2019

Professor Emeritus
Computer Science
Cornell University
Ithaca, NY 14853, USA

Date of Birth: 26 April 1939
Citizenship: United States
Email: gries@cs.cornell.edu
Phone: (607) 255-0393

AWARDS

1. Doctor of Science (Honorary Degree), Miami University, Oxford, Ohio, May 1999.
2. Doctor of Laws (Honorary Degree), Daniel Webster College, Nashua, NH, May 1996.
3. Ranked 8th in the *Top 10 Papers of the first 50 years of SIGCSE* for the 1974 paper, *What should we teach in an introductory programming course?*
4. 2016 Amity Booker Prize, Amity University, New Delhi, for the 2004 text *Multimedia Introduction to Programming using Java*, (with Paul Gries).
5. 1995 ACM Karl V. Karlstrom Outstanding Educator Award, Feb. 1996.
6. 1994 Taylor L. Booth Award Education Award, IEEE Computer Society, Dec. 1995.
7. Fellow of the ACM (charter member), March 1994.
8. Advisor of T.V. Raman, winner of the ACM Best Ph.D. Dissertation Award, 1995.
9. ACM-SIGCSE Award for Outstanding Contributions to Computer Science Education, 1991.
10. Computing Research Association Award for Service to the Computing Community, 1991.
11. Fellow of the AAAS, 1990.
12. Education Award, American Federation of Information Processing Societies, 1986.
13. Guggenheim Fellowship, 1983-84.
14. ACM Programming Systems and Languages Paper Award, 1977 (with S. Owicki).
15. Superior Accomplishment Award, U.S. Naval Weapons Lab, Dahlgren, VA, 1961.

AWARDS —Cornell University

1. 1995. Weiss Presidential Fellow, for contributions to undergraduate education. Cornell University. (seventh person out of over 1,000 Cornell faculty chosen for this award.)
2. 2017. Class of 2019 Award, “In recognition of extraordinary achievement and service, making the class of 2019 and Cornell a better experience for all.”
3. 2017. Chosen by Cornell Merrill Presidential Scholar Yezy Lin as the faculty member who had the most positive influence on her education at Cornell, 2017.
4. 2017. ACSU Faculty of the Year, 2016-2017 (presented by the Cornell Association for Computer Science Undergraduates).
5. 1999. ACSU Faculty of the Year, 1998-99 (presented by the Cornell Association for Computer Science Undergraduates).
6. 1996.ACSU Faculty of the Year, 1995-96 (presented by the Cornell Association for Computer Science Undergraduates).
7. 1990. Chosen by Cornell Merrill Presidential Scholar Thomas Yan as the faculty member who had the most positive influence on his education at Cornell
8. 1987. Clarke Award for Excellence in Undergraduate Teaching; College of Arts & Science.

EDUCATION

1960 Queens College, B.S. in Mathematics

1963 University of Illinois, M.S. in Mathematics
1966 Munich Institute of Technology, Dr. rer. nat. in Mathematics

EXPERIENCE

2012-	Professor Emeritus
2003-11	Associate Dean for Undergraduate Programs, Engineering College, Cornell University
2003-11	Professor of Computer Science, Cornell University
2001-01	Visiting Professor, Computer Science, Cornell University
1999-03	Franklin Professor of Computer Science, University of Georgia
1992-99	William L. Lewis Professor of Engineering, Computer Science, Cornell University
1997-97	Sabbatic, Courtesy Professor, Computer Science, Oregon State University
1990-91	Sabbatic, Regents Chair in Computer Science, Univ. Texas, Austin
1983-84	Sabbatic, Oxford University
1982-87	Chair, Computer Science Department, Cornell
1977-99	Professor, Computer Science, Cornell University
1975-76	Sabbatic, Technical University, Munich
1969-77	Associate Professor, Computer Science, Cornell University
1966-69	Assistant Professor, Computer Science, Stanford University
1963-66	Assistant, Munich Inst. of Technology
1960-62	Mathematician-programmer, U.S. Naval Weapons Lab.

PH.D. STUDENTS ADVISED

1971 Susan L. Graham
1972 Jean-Pierre Levy
1972 John S. Brown
1973 Donald B. Johnson
1975 Narain Gehani
1975 Susan S. Owicki
1977 Juan M. Lafuente
1977 Kuo-Chung Tai
1980 Gary M. Levin
1980 John Holm
1981 Robert Melville
1984 Peter Vanderbilt
1985 Dean Jacobs
1985 Van Long Nguyen
1987 Jennifer Widom
1987 Jan Prins
1991 Geoffrey Smith
1994 T.V. Raman
1994 Sofoklis Efremidis
2000 Eric Aaron

COMMUNITY PARTICIPATION AND LEADERSHIP (SELECTED)

1. Member, Computing Research Board, 1983-1990.
2. Chair, Computing Research Board, 1987-89.
3. Member, Steering Group, AAAS Section on Information, Computing, and Communication (T), 1999-2003.
4. Series Editor, Springer Verlag Texts and Monographs in Computer Science, 1973-.
5. Main Editor, *Acta Informatica*, 1982-.
6. Managing Editor, *Information Processing Letters*, 1988-2003.
7. Editor, Programming Languages section of *CACM*, 1969-72.
8. Editorial Board: Acta Informatica, 1970-; Information Processing Letters, 1973-; Springer Verlag Lecture Notes in Computer Science, 1973-00; Distributed Computing 1985-88; Software –Concepts and Tools, 1988-95; Formal Aspects of Computer Science, 1988-95.
9. Coeditor, Special issue of IPL, April 2001. (with V. Almstrum).
10. Advisory Board, Conf. on Foundations of Software Technology and Theoretical CS (India), 1981-90.
11. Chair, Program Committee, 2nd IEEE Symp. Logics in Computer Science, 1987.
12. Co-chair, Program Committee, IFIP Working Conference on Programming Concepts and Methods

(PROCOMET '98).

13. Member of Program Committees (selected): 5th Int. Symp. Programming. 1981; 6th Int. Symp. Programming. 1982; POPL 1978; POPL 79; 6th Int. Conf. Software Engineering, 1982; IFIP Working Conference on Programming Concepts and Methods, 1990; SCCC 97; Mathematics of program construction 98.
14. National Research Council to Survey International Developments in Computer Science. 1980-82.
15. Computer Science Screening Committee, Council for International Exchange of Scholars, 1984-86.
16. Chair, Panel on research directions in Software. NSF Advisory Committee Meeting, Dec. 1988.
17. Member, Committee to Assess the Scope and Direction of Computer Science and Technology (Computer Science and Telecommunications Board of the National Research Council), 1990-92.
18. Member, Selection committee for NAS Graduate fellowships, 1971-72.
19. Scientific Director, Institute for Encapsulation, Modularization and Reuseability, Year of Programming, University of Texas at Austin, 1987.
20. Member, Adhoc AP committee to recommend the next programming language, 1999-2001.
21. ACM Task Force on Computer Science Core Curriculum, 1987-89.
22. ACM Strategic Planning Committee, 1988-90.
23. Chair, ACM Award Committee, Best Paper in Programming Languages and Systems 1972-75.
24. National ACM Lecturer, 1970-71.
25. Member, IFIP Working Group 2.3, 1974-.
26. Reviewer for the PROCOS research project, the ESPRIT program, 1990.
27. Review Committee, several computer science departments.

PATENTS

1. United States Patent 5,572,626, 5 November 1996. Method for generating audio renderings of digitized works having highly technical content (with T.V. Raman).

PUBLICATIONS

Books

1. *The ABC Book of Computer Science at Cornell.* Computer Science, Cornell University, Ithaca, NY, 2005.
2. *Leadership in Research and Education: 40 years* (Editor). Computer Science, Cornell University, Ithaca, NY, 2005.
3. *Multimedia Introduction to Programming Using Java*, Springer Verlag, NY. 2004, 536 pages (with P. Gries).
4. *The ProgramLive Companion.* John Wiley & Sons, NY, 2001. (with P. Gries and P. Petra).
5. *ProgramLive.* Data Description, Ithaca, NY, 2003 (revised). [This intro to programming using Java is a livetext: a CD with over 250 recorded lectures with synched animation.]
6. *Programming Concepts and Methods, PROCOMET '98* (ed.). Chapman and Hall, London, 1998. (with W.-P. De Roever).
7. *A Logical Approach to Discrete Math.* Springer Verlag, NY, 1993, 500 pages (with F.B. Schneider).
8. *Instructor's Manual for "A Logical Approach to Discrete Math".* D. Gries and F.B. Schneider, Ithaca, 1993. 311 pages (with F.B. Schneider).
9. *Beauty is our Business.* Editor. Springer Verlag, New York, 1990, 453 pages. (with W.H.J. Feijen, A.J.M. van Gasteren, and J. Misra).
10. *The Science of Programming.* Springer Verlag, New York, 1981, 350 pages. (Translated into Russian and Japanese.)

11. *Programming Methodology: a Collection of Articles by Members of IFIP WG2.3.* Editor. Springer Verlag, New York, 1979, 437 pages.
12. *An Introduction to Programming —a structured approach.* Winthrop, Cambridge, Edition 3, 1978, 509 pages. (with R. Conway).
13. The following books are extensive revisions to suit different needs:
    1. *A Primer on Structured Programming, Using PL/I, PL/C and PL/C7.* 1976, 397 pages. (with R. Conway).
    2. *Primer on Structured Programming using Pascal.* 1976, 433 pages. (with R. Conway and E.C. Zimmerman).
    3. *An Introduction to Structured Programming using SP/K.* 1977. (with R. Conway and D. Wortman).
    4. *Introduction to Microprocessor Programming using PL/Z.* 1979, 593 pages. (with R. Conway, M. Fay, and C. Bass).
14. *Compiler Construction for Digital Computers.* John Wiley and Sons, New York, 1971, 491 pages. (Translated into Spanish, Japanese, Chinese, Italian and Russian.)

Invited Lectures and Panels since 2000

1. A Principled Approach to Teaching Java. Workshop, Arizona State University, 20 Jan 2012.
2. Teaching Logic. Banquet speech, 17th Intl Symp Formal Methods, Limerick, Ireland, 24 June 2011.
3. Calculational Logic: teaching student to think about proofs. Keynote address, Third International Congress on Tools for Teaching Logic (TICTTL), Salamanca, Spain, 3 June 2011.
4. Teaching a useful logic. Computer Science Department, RIT, Rochester, NY, 17 March 2010.
5. Teaching programming, not programs. Keynote, 4th International Conference on Informatics in Secondary Schools, ETH, Zurich, Switzerland, 13 Jan 2010.
6. Compiler construction. Lectures and practicum, Sri Sathya Sai University, Puttaparthy, India, 6-10 July 2009.
7. Panel: Discrete Mathematics/Structures: How Do We Deal With the Late Appreciation Problem? CCSNE 2009, SUNY Plattsburgh, 24 April 2009.
8. A Principled Approach to Teaching Java. Workshop, Arizona State University, 9 Feb 2009.
9. Teaching OO to beginners. Keynote, Educators' Symposium, OOPSLA 2008, Nashville, TN, 20 Oct 2008.
10. A principled approach to teaching OO first. SIGCSE 2008, Portland, Oregon, 20 March 2008.
11. Education in Programming Methodology. Invited Presentation, 40 Jahre Informatik an der TUM, Garching, Germany, 25 Oct 2007.
12. Teaching OO Programming [using Java]. Keynote, PPPJ07, Lisbon, Portugal, 6 Sept 2007.
13. Developing Correct Software. Keynote. International Forum on Computer Science and Advanced Software Technology, Jiangxi Normal University, Nanchang, Jiangxi, China, 9 June 2007.
14. Compiler construction: 24 June – 10 July 2006. 3-week course given to MTech students, Sri Sathya Sai Institute of Higher Learning. Prasanthi Nilayam, India.
15. Panel moderator. Next Generation Info. Tech. for Societal Advancement and Integration, Sri Sathya Sai Institute of Higher Learning, Prasanthi Nilayam, India, 17 July 2006.
16. Programming and software engineering education. Keynote address, 30th Annual Software Engineering Workshop, Columbia, MD, 26 April 2006.
17. Teaching program development. Computer Science Department, IIT Kanpur, India, 16 Nov. 2005.
18. CS Education. Invited presentation, NSF Workshop on Integrative Computing Education and Research, Boston, MA, 3-4 Nov. 2005.

19. An ABC Book. Banquet speech at the CS Cornell 40th anniversary symposium, 1 October 2005.
20. Compiler writing. 2-week practicum, Computer Science & Math Department, Sri Sathya Sai Institute of Higher Learning, Puttaparthy, India, 13-26 July 2005.
21. The science of programming. InfoSys, Bangalore, India, 11 July 2005.
22. Teaching Program Development. Keynote lecture. ACM Mid-South College Computing Conference (MSCCC), University of Mississippi, Oxford, MS (April 1-2, 2005).
23. Calculus and calculational logic. Invited talk. Festveranstaltung Friedrich L. Bauer 80 Jahre (June 2004). Technishe Universitaet Muenchen, Garching bei Munich, Germany.
24. Teaching the development of algorithms. Keynote lecture. ACM MidSoutheast Conference, Knoxville, Tennessee (November 11-12, 2004).
25. Imparting Skill and Values, not Just Knowledge. Invited lecture. Symposium on Teaching Formal Methods. Ghent, Belgium (November 18-19, 2004).
26. Problems in computer science education. Banquet speech. ITiCSE 2002, Aarhus, Denmark, 25 June 2002.
27. On the occasion of the retirement of Juris and Elly Hartmanis. Banquet speech. Cornell University, Ithaca, NY, 6 May 2001.

Published Articles and Lectures

1. What have we not learned about teaching programming? *Computer, 39*, 10 (Oct 2006), 81-82.
2. Supporting workflow in a course management system. *SIGCSE 2005* (Feb 23-27, 2005), St. Louis, MO. (with 19 others)
3. Substitution of equals for equals. (with F.B. Schneider). In *Millennial Perspectives in Computer Science*, (eds. J. Davies, B. Roscoe, and J. Woodcock). Palgrave Houndmills, U.K. 2000. p, 119–132.
4. Monotonicity in calculational proofs. In *Correct System Design* (eds. E.-R. Olderog and B. Steffen), LNCS 1710, Springer Verlag, 1999, p. 79–85.
5. Logic as a tool. 10 lectures at WOFRACS 98, Workshop on Formal and Applied Computer Science, Winter School on Programming Methodology, University of CapeTown, South Africa, 6-11 July 1998.
6. The programming language Java. 10 lectures, Math Department, Sathya Sai Baba Institute of Higher Learning, Puttaparthi, India. 13–18 July 1998.
7. Eliminating the chaff —again. Banquet speech at Marktoberdorf 1996. In M. Broy and B. Schieder (eds.) *Mathematical Methods in Program Development.* NATO ASI Series, Series F: Computer and Systems Sciences 158, Springer Verlag 1997, 1–7.
8. Foundations for calculational logic. Ibid, 83–126.
9. The once and future programmer: our educational dilemma (Banquet speech). Third Annual Northeastern Conference, the Consortium for Computing in Small Colleges. Fairfield, CT, 24 April 1998.
10. Adding the everywhere operator to propositional logic. *Journal of Logic and Computation 8*, 1 (February 1998). (With F.B. Schneider).
11. KMP string matching revisited. *IPL 64* (1997), 217–223. (with E.M. Reingold and K.J. Urban).
12. Formal justification of underspecification for S5. *IPL 64*, 3 (Nov. 1997), 115–121. (with EȦaron).
13. *What's education for?* Address at the Opening Convocation, Daniel Webster College, 6 September 1996.
14. Data refinement and the transform. *Deductive Program Design* (M. Broy, ed.), AASI Vol. F152, Springer Verlag, pp. 205-232, 1996.
15. The need for education in useful formal logic. *IEEE Computer* (April 1966), 29-30.
16. Avoiding the undefined by underspecification. In J. van Leeuwen (Ed.). *Computer Science Today.* Springer Verlag Lecture Notes in Computer Science 1000, October 1995, 366–373.

17. Teaching math more effectively through calculational proofs. *The Mathematical Monthly* (October 1995), 691–697. (with F.B. Schneider).
18. A new approach to teaching discrete mathematics. *PRIMUS V*, 2 (June 1995), 113–138. (with F.B. Schneider).
19. Equational propositional logic. *IPL 53*, 3 (February 1995), 145–152. (with Fred B. Schneider).
20. Documents mean more than just paper! *Mathl. Comput. Modeling 26*, 1 (1997), 45-53. (With T.V. Raman).
21. Documents mean more than just paper! *Proc. Second International Workshop on the Principles of Document Processing*, April 1994 (with T.V. Raman).
22. Data refinement and the transform. In Broy (ed.), *Program Design Calculi.* NATO ASI Series F, 118, Springer-Verlag, Heidelberg, 1993. pp. 92-119.
23. Computer Science, Academic. Encyclopedia of Computer Science, Third Edition (A. Ralston and E.D. Reilly, eds.), Van Nostrand Reinhold, New York, 1993, p. 322-323 (with D. Marsh).
24. Loop invariant. Encyclopedia of Computer Science, Third Edition (A. Ralston and E.D. Reilly, eds.), Van Nostrand Reinhold, New York, 1993, p. 796-797.
25. Trace-based network proof systems: expressiveness and completeness. *TOPLAS 14*, 3 (July 1992), 396-416. (with J. Widom and F.B. Schneider).
26. A new programming language. *Proc. Software Technology Conference 1992*, Los Angeles, California, 27-29 April 1992, pp. 97-101.
27. A constructive proof of Vizing's theorem. *IPL 41*, 3 (March 1992), 131-133. (with J. Misra).
28. On progress in programming. In Broy, M. (ed.). *Informatik und Mathematik*, Springer Verlag, Heidelberg, 1991, pp. 297-302.
29. Teaching calculation and discrimination: a more effective curriculum. *Comm. ACM 34*, 3 (March 1991), 44-55.
30. Binary to decimal, one more time. In *Beauty is our Business* (W.H.J. Feijen et al, eds.), Springer Verlag, New York, 1990, pp. 141-148.
31. The transform —a new language construct. *Structured Programming 11* (1990), 1-10. (with D. Volpano).
32. An optimal parallel algorithm for generating combinations. IPL 33 (November 1989), 135-139. (with S.E. Akl and I. Stojmenovic).
33. A hands-in-the-pocket presentation of a k-majority vote algorithm. In *Formal development of programs and proofs* (E.W. Dijkstra, ed.), Addison-Wesley, Menlo Park, 1990, pp. 43-45.
34. Influences (or lack thereof) of formalism in teaching programming and software engineering. (Banquet speech.) *Ibid*, pp. 229-236.
35. Inorder traversal of a binary tree and its inversion. *Ibid*, pp. 37-42. (with J.L.A. van de Snepscheut).
36. The maximum-segment-sum problem. *Ibid*, pp. 33-36.
37. An algorithm for transitive reduction of an acyclic graph. *Science of Computer Programming 12*, 2 (July 1989), 151-155. (with A.J. Martin, J.L.A. van de Snepscheut, and J.T. Udding).
38. Developing a linear algorithm for cubing a cyclic permutation. *Science of Computer Programming 11*, 2 (December 1988), 161-166. (with J. Xue).
39. Generating a random cyclic permutation. *BIT 28*, 3 (1988), 569-572. (with J. Xue).
40. A note on Graham's convex hull algorithm. *IPL 25*, 5 (July 1987), 323-328. (with I. Stojmenovic).
41. Horner's rule and the computation of linear recurrences. *IPL 25*, 4 (June 1987), 237-240. (with A. Pascoletti and L. Sbriz).
42. Programming pearls, guest oyster. *Comm. ACM 30*, 4 (April 1987), 284-290.
43. Mclaren's masterpiece. *Science of Computer Programming 8*, 2 (April 1987), 139-146. (with J.F. Prins).
44. In-situ inversion of a cyclic permutation. *IPL 24*, 1 (January 1987), 11-14. (with W.H.J. Feijen and

A.J.M. van Gasteren).

45. Completeness and incompleteness of trace-based network proof systems. *Proc. Fourteenth Ann. ACM SIGACT-SIGPLAN Symp. Princ. Programming Languages*, January 1987, Munich. (with J. Widom and F.B. Schneider).
46. Programming methodology: making a science out of an art. *Engineering: Cornell Quarterly 20*, 2 (Fall 1985), 23-27. (with F.B. Schneider).
47. A model and temporal proof system for networks of processes. *Distributed Computing 1*, 1 (1986), 7-25. (with V. Nguyen, A. Demers, and S. Owicki).
48. A new notion of encapsulation. *Proc. SIGPLAN 85 Symp. Language Issues in Programming Environments*, Seattle, June 1985, 131-139. (with J. Prins).
49. Behavior: a temporal approach to process modeling. Conference on Logics of Programs, Brooklyn, June 1985. *LNCS 193* (1985), 237-254. (with V. Nguyen, A. Demers, and S. Owicki).
50. A model and temporal proof system for networks of processes. *Twelfth Ann. ACM Symp. Princ. Progr. Lang.*, New Orleans, January 1985, 121-131. (with V. Nguyen and S. Owicki).
51. General correctness: a unification of partial and total correctness. *Acta Informatica 22* (1985), 67-84. (with D. Jacobs).
52. The emerging science of programming (Keynote Address). *6th Australian Computer Science Conference*, Sydney, Australia, February 12, 1983.
53. Towards a practical science of programming (Keynote Address). *2nd Conf. Foundations of Software Technology and Theoretical Computer Science*, Bangalore, India, December 13, 1982.
54. Fast fault-tolerant broadcasts. *Science of Computer Programming 4* (April 1984), 1-16. (with F.B. Schneider and R.D. Schlichting).
55. A note on a standard strategy for developing loop invariants and loops. *Science of Computer Programming 2* (1984), 207-214.
56. Finding repeated elements. *Science of Computer Programming 2* (November 1982), 143-152. (with J. Misra).
57. A proof technique for communicating sequential processes. *Acta Informatica 15* (1980), 281-302. (with G. Levin).
58. Educating the programmer: notation, proofs and the development of programs. Invited lecture. *Information Processing 80, Proc. IFIP Congress 80,* S. Lavington, (Ed.), North-Holland, New York, 1980, 935-944.
59. Computing a Fibonacci number in log time. *IPL 2* (October 1980), 68-69. (with G. Levin).
60. Assignment and procedure call proof rules. *TOPLAS 2* (October 1980), 564-579. (with G. Levin).
61. Controlled-density sorting. *IPL 10* (July 1980), 169-172. (with R. Melville).
62. Is sometime ever better than alway? *TOPLAS 1* (October 1979), 258-265.
63. Eliminating the chaff. In *Program Construction*, F.L. Bauer and M. Broy (Eds.), *LNCS 69* (1979), 70-74.
64. Current ideas in programming methodology. In *Research Directions in Software Technology*, P. Wegner (Ed.), MIT Press, 1979, 254-275.
65. The Schorr-Waite graph marking algorithm. *Acta Informatica 11* (1979), 223-232.
66. A linear sieve algorithm for finding prime numbers. *Comm. ACM 21* (December 1978), 999-1003. (with J. Misra).
67. Parallel programming (invited lecture). GI-8 Jahrestagung, *Springer Verlag Informatik Berichte 16* (October 1976) Berlin, Germany, 214-233.
68. The multiple assignment statement. *IEEE Trans. Softw. Eng. 4* (March 1978), 89-93.
69. Language facilities for programming user-computer dialogues. *IBM J. Research and Development 22* (March 1978), 148-158. Reprinted in E. Edmunds (ed.). *The Separable User Interface.* Academic Press, San Diego, 1992, pp. 97-128. (with J. LaFuente).

70. An exercise in proving parallel programs correct. *Comm. ACM 20* (December 1977), 921-930.
71. Some ideas on data types in high-level languages. *Comm. ACM 20* (June 1977), 414-420. (with N. Gehani).
72. An illustration of current ideas on the derivation of correctness proofs and correct programs. *IEEE Trans. Softw. Eng. 2* (December 1976), 238-244.
73. Compiler. *Encyclopedia of Computer Science and Technology 5*, Belzer, Holzman and Kent (Eds.), Markel Dekker Inc., New York, 206-243.
74. Error recovery and correction —an introduction to the literature. *Compiler Construction —an Advanced Course*, F.L. Bauer and J. Eickel (Eds.), *LNCS 21* (edition 2), Springer Verlag, New York, 1976, 627-638.
75. Some comments on programming language design (invited lecture). *Programmiersprachen: Fachtagung 1976*, H.J. Schneider and M. Nagl (Eds.), Informatik-Fachberichte I, Springer Verlag, Heidelberg, 1976, 235-252.
76. An axiomatic proof technique for parallel programs. *Acta Informatica 6* (1976), 319-340. (with S. Owicki).
77. Verifying properties of parallel programs: an axiomatic approach. *Comm. ACM 19* (May 1976), 279-285. (with S. Owicki).
78. On structured programming —a reply to Smoliar. *Comm. ACM 17* (November 1974), 655-657; also *Comm. ACM 18* (October 1975), 600-601 and *Comm. ACM 19* (January 1976), 47-48.
79. What should we teach in an introductory programming course? SIGCSE Bulletin 6, February 1974, 81-89. (*Proc. 4th Symp. Computer Science Education.*)
80. Describing an algorithm by Hopcroft. *Acta Informatica 2* (1973), 97-109.
81. Programming by induction. *IPL 2* (1972), 100-107.
82. Program schemes with pushdown stores. *SIAM J. Computing 1* (1972), 242-268. (with R. Szymanski and S. Brown).
83. On classes of program schemata. *SIAM J. Computing 1*, 1 (1972), 66-118. (with R. Constable).
84. Automatic compiler writing techniques. In *Systemprogrammierung*, Ganzhorn, Schieferdecker, Endres (Ed.), Oldenbourg Verlag, Munich, 1972.
85. Translator writing systems. *Comm. ACM 11* (February 1968), 77-113. (with J. Feldman).
86. The use of transition matrices in compiling. *Comm. ACM 11* (January 1968), 26-34.
87. The ALCOR-ILLINOIS 7090/7040 post mortem dump. *Comm. ACM 10* (December 1967), 804-808. (with R. Bayer, M. Paul and H.R. Wiehle).
88. Characterizations of certain classes of norms. *Numerische Mathematik 10* (1967), 30-41.
89. Some results on fields of value of a matrix. *SIAM J. Numerical Analysis 4* (1967), 283-299. (with J. Stoer).
90. Some techniques used in the ALCOR-ILLINOIS 7090. *Comm. ACM 8* (August 1965), 496-500. (with M. Paul and H.R. Wiehle).

Published Reports

1. Computer science. *Cornell Engineering Quarterly 25*, 4 (Summer 1991), 21-23.
2. Die 150 wissenschaftlichen Nachtkommen von Prof.Dr.Dr.h.c.mult. F.L. Bauer. In Broy, M. (ed.). *Informatik und Mathematik*, Springer Verlag, Heidelberg, 1991, pp. 441-450.
3. The 1989-90 Taulbee survey. *Computing Research News* (January 1991) and *Comm. ACM 35*, 1 (January 1992), 132-143. (with Dorothy Marsh).
4. The 1988-89 Taulbee survey. *Computing Research News 2*, 1 (January 1990) and *Comm. ACM 33*, 9 (September 1990), 160-169. (with Dorothy Marsh).
5. Computer languages. In J.E. Hopcroft and K.W. Kennedy, ed. *Computer Science: achievements and opportunities.* Report of the NSF Advisory Committee for Computer Research. SIAM, Philadel-

phia, 1989, pp. 15-28. (with C.A.R. Hoare, K.W. Kennedy, F.C.N. Pereira, and D.S. Scott).

6. The Computer Worm. (Report from the Commission of Preliminary Enquiry. Cornell University), Cornell University, 6 February 1989. (with T. Eisenberg, J. Hartmanis, D. Holcomb, M.S. Lynn, and T. Santoro).
7. The 1988 Snowbird Report: a discipline matures. *Computer 22*, 2 (1989), 72-75. (with T. Walker and P.R. Young).
8. Computing as a Discipline. Final report of ACM Task Force on the Core of Computer Science. *Computer 22*, 2 (1989), 63-70. (with P.J. Denning,. D.E. Comer, M.C. Mulder, A. Tucker, A.J. Turner, and P.R. Young).
9. The 1987-88 Taulbee survey. *Comm. ACM 32*, 10 (October 1989), 1217-1224; and *Computer 22*, 11 (November 1989), 49-56. (with D. Marsh).
10. The 1986-87 Taulbee survey. *Comm. ACM 31* (August 1988), 984-991; and *Computer 21* (August 1988), 53-61. (with Dorothy Marsh).
11. The 1985-86 Taulbee survey. *Comm. ACM 30* (August 1987), 688-694.
12. Meeting the basic research needs of computer science. Report of the study group of the Advisory Committee for Computer Research, National science Foundation, December 1986. (with S.R. Kosaraju and M.J. Fischer).
13. The 1984-85 Taulbee survey. *Comm. ACM 29* (Oct 1986), 972-977.
14. Twenty years of computer science at Cornell. *Engineering: Cornell Quarterly 20*, 2 (Fall 1985), 2-12.
15. Imbalance between growth and funding in academic computer science: two trends colliding. *Computer 19* (September 1986), 70-76, and *Comm. ACM 29* (September 1986), 870-878. (with R. Miller, R. Ritchie, and P. Young).

Other reports and reviews

1. Banquet speech, Conference at the retirement of Professor Juris Hartmanis. Cornell University, Ithaca, NY. 5 May 2001.
2. Foreword for T.V. Raman's book *Auditory User Interfaces (AUI) —toward the speaking computer.*
3. On proving monotonicity and on EA=>AE. Tech. Rpt. TR95-1512, Computer Science Department, Cornell University, May 1995.
4. Teaching math more effectively through the design of calculational proofs. Tech. Rpt. 94-1415, March 1994. Accepted for publication in the *Mathematical Monthly*, January 1995.
5. A constructive proof of Vizing's theorem. Tech. Rpt., Computer Science Department, Cornell University, August 1991. (with J. Misra).
6. Forward, *Essays in Computing Science* by C.A.R. Hoare and C.B. Jones. Prentice Hall International, London, 1989.
7. The Hopcroft-Tarjan planarity algorithm, presentation and improvements. Tech. Rpt. 88-906, April 1988. (with J. Xue).
8. Presenting an algorithm to find the minimum edit distance. Tech. Rpt. 88-903, March 1988. (with B. Burkhart).
9. Inorder traversal of a binary tree and its inversion. Tech. Rpt. 87-876, November 1987. (with Jan L.A. van de Snepscheut).
10. Complete, trace-based, network proof systems: an advisor's perspective. Tech. Rpt. 87-865, Computer Science Department, Cornell University, September 1987.
11. Developing two of Arsac's funny algorithms. Tech. Rep. 85-711, Computer Science Department, Cornell University, November 1985.
12. The Gibbs Programming Environment. IEEE International Conference on Computer Design: VLSI in Computers. (ICCD '85), Oct 7 1985, Port Chester, NY.

13. The seven-eleven problem. Tech. Rep. 83-574, Computer Science, Cornell University, September 1983. (with P. Pritchard)
14. Review of R.G. Dromy's book *How to solve it by computer. Science of Computer Programming 3* (1983).
15. A reply to Susan Gerhart's review of *Craft* and *Science*. May 1982.
16. The science of programming answer book. Tech. Rep. 82-498, Computer Science, Cornell University, Ithaca, New York, June 1982.
17. The 711 problem. Tech. Rep. 82-493, Computer Science, Cornell University, Ithaca, New York, May 1982. (with J. Misra).
18. Finding the longest prefix. August 1981.
19. Notes on Mathematical Induction and Well-Founded Sets. November 1981.
20. **Cand** and **Cor** before **and then** and **or else** in ADA;. Tech. Rep. 79-402, Computer Science, Cornell University, Ithaca, New York, May 1979.
21. Yet another exercise: using shared variables in two processes to provide starvation-free mutual exclusion. Tech. Rep. 79-372, Computer Science, Cornell University, Ithaca, New York, 1979.
22. Tutorial on the development of programs. COMCON 77, September 1977. (with D. Parnas).
23. A note on iteration. Tech. Rep. 77-323, Computer Science, Cornell University, Ithaca, New York, September 1977.
24. A proof of correctness of Rem's semaphore implementation of the *with-when* statement. Tech. Rep. 77-314, Computer Science, Cornell University, Ithaca, New York, May 1977.
25. The use of comments on programming. 1976. (with R. Conway).
26. A student-written compiler. Tutorial Software Document 1, DISE Task Force on Software Engineering, May 1975.
27. Recursion as a programming tool. Tech. Rep. 75-242, Computer Science, Cornell University, Ithaca, New York, May 1975.
28. A note on program development. Tech. Rep. 74-202, Computer Science, Cornell University, Ithaca, New York, March 1974.
29. Using recipes to teach programming. 1970.
30. CIL: Compiler Implementation Language. SLAC Report 102, March 1969.
31. MPL: Mathematical Programming Language. Computer Science Report, Computer Science Department, Stanford University, 1968. (with 7 others).
32. *Ueber einige Klassen von Normen.* Ph.D. Thesis, Technische Hochschule Muenchen, 1966.
33. ALCOR-ILLINOIS 7090 —an ALGOL compiler for the IBM 7090. Report 6415, Rech. der Tech. Hochschule Munich, 1964. (with M. Paul and H.R. Wiehle).
34. The object program produced by the ALCOR-ILLINOIS 7090 compiler. Report 6412, Rech. der Tech. Hochschule Munich, 1964.
35. User's manual for the ALCOR-ILLINOIS 7090 ALGOL translator. University of Illinois, September 1964, 120 pages. (with R. Bayer and E. Murphree Jr).